Management for Professionals

Mario Vanhoucke

The Illusion of Control

Project Data, Computer Algorithms and Human Intuition for Project Management and Control

Springer

Management for Professionals

The Springer series Management for Professionals comprises high-level business and management books for executives, MBA students, and practice-oriented business researchers. The topics span all themes of relevance for businesses and the business ecosystem. The authors are experienced business professionals and renowned professors who combine scientific backgrounds, best practices, and entrepreneurial vision to provide powerful insights into how to achieve business excellence.

Mario Vanhoucke

The Illusion of Control

Project Data, Computer Algorithms and Human Intuition for Project Management and Control

Springer

Mario Vanhoucke
Faculty of Economics and Business
Administration
Ghent University
Gent, Belgium

ISSN 2192-8096 ISSN 2192-810X (electronic)
Management for Professionals
ISBN 978-3-031-31784-2 ISBN 978-3-031-31785-9 (eBook)
https://doi.org/10.1007/978-3-031-31785-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my wife, my children, and their partners
who make my life better.
To my amazing team of PhD students who
make my work inspiring and fun.
To the Portuguese sun that makes me forget
the weather in Belgium for a while.

Preface

We live in a world where the words *big data* have become buzzwords. They refer not only to the growing availability of data but also to the increasingly powerful methods for analysing this huge amount of data. The presence of data has always been important for decision-making in most management disciplines, and thus also for improving the decision-making process of managing projects.

This book tells a story about the increasing importance of such data for project scheduling, risk analysis, and project control. It is a story about the importance of project data for researchers and professionals, and why collecting, processing, and using such data are not as easy as we often think. The book also aims to show the differences and similarities in project data needs between researchers and professionals. Because they both need data, albeit for slightly different purposes, the book is also about connecting two worlds (academic and professional) by sharing their project data, the algorithms, the statistical methodologies, and the results of their analyses.

The theme of this book is known in the literature as *dynamic scheduling* or *integrated project management and control* and is now called *data-driven project management*. The first part of this book provides a brief overview of the basic components of data-driven project management, emphasising the type of data required for each component and providing an overview of the current state-of-the-art methodologies available in the literature.

The second part of this book emphasises the importance and relevance of *artificial project data* for academics and describes the specificities and requirements of such data for research purposes. It describes how academics deal with their need for project data to test new ideas and new methodologies, and how they generate and adapt these project data to their research purposes. This section mainly introduces the readers to the wonderful world of academic research in project control and shows that through a rich set of artificial project data, the research has taken a long and exciting journey.

The third part of this book focuses on the availability of diverse project data from a professional point of view, describing how professionals have been able to collect a diverse set of *empirical project data*. Their data are often very case-specific and tailored to specific needs, often sector or company dependent, not always complete, and often, if not always, confidential and therefore usually, unfortunately, not freely available to researchers. Despite these drawbacks, this part of the book emphasises the relevance and necessity of real project data for scientific research. It aims to convince researchers not only to rely on their own generated project data but also to look to the outside world and spend some time collecting data for real projects. This section introduces my readers to the exciting journey of research relevant to real project managers.

The fourth part of this book takes a deep dive into the two forms of project data, (*artificial* and *empirical*), and presents an approach for collecting, generating, and analysing projects to create a database that is relevant to academic researchers *and* professional project managers. It presents a two-way approach, transferring project data from business to academia and back to business, to gain a better understanding of project scheduling and control methods by both researchers and professionals.

Finally, in the fifth part, I close this book with a personal view of four important qualities that a good researcher must possess. Unlike the previous parts, it is not based on science at all, but simply a reflection of how I look at my wonderful job as a researcher.

While I fully realise that this book is not the first, and probably not the last, written work on the relevance of data for decision-making in project management, it is primarily intended to explore these two different worlds (theoretical academia and professional business) and bring them closer together so that projects under risk can be better managed. This book therefore aims to shed some light on the often confusing relationship between academics and professionals when it comes to generating project data, implementing optimisation algorithms, and using statistical analysis to improve the decision-making process in project scheduling, risk analysis, and project control.

Welcome to a data-driven journey through the world of project management.

Lisbon, Portugal
March 2023

Mario Vanhoucke

Acknowledgements

This book tells the research story of my team from the *Operations Research and Scheduling* group. Each member has contributed to this story in their own way. Some did this by developing research ideas and turning them into accepted papers. Others helped me write case studies for education. Some even put me in touch with professionals in the field from whom I got a lot of inspiration. Of course there were also people who read the book in detail and noticed the many mistakes I had made. Their contribution, however varied, was crucial in the making of this book, and so I prefer not to include their names in this acknowledgement. Instead, I have mentioned them in the various chapters of this book, where their contribution will be discussed in detail.

Contents

Part I
Data-Driven Project Management

Without data,
you are just another project manager with an opinion.

Chapter 1
About This Book

This introductory chapter is meant to explain why I have written a book about the gap between theory and practice in data-driven project management. It explains why I have written more than 300 pages on the use of algorithms and human intuition for decision making in project management and on the importance of project data for scientific research (*the academic world*) and in the business (*the professional world*). In this introduction, I want to convince my readers that there is a strong need for a book on project data and that the book's theme is interesting for both the academic *and* the professional world. I will give a concise summary of the topics that will be discussed in the upcoming chapters of this book and explain why the chapters are divided into four different parts. It is important to know that I wrote every chapter from my own personal perspective, without having any ambition to give a full overview of the existing work published in the literature. Instead, I will take the readers on my personal journey in the research conducted over the last two decades, and while showing them most of the key findings of my research, I will also have them meet my key team members. Let me start at the beginning of my research career in project management and tell how my work has resulted in the decision to write a book about the theory/practice gap when using project data.

1.1 Theory and Practice

My research interest in data-driven project management started in 1996, when I just graduated from the *University of Leuven* (Belgium) as a *Master in Business Engineering* and decided to start an academic career as a PhD student in Operations Management. When I look back, I do not know whether this choice was a deliberate move into academia. I think it was nothing more than an obvious continuation of my interest in algorithms and programming that I developed during my Master Thesis, and it felt right to stay in this challenging domain for a while. In search

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_1

for a relevant PhD topic to further develop my interest in algorithms, I (randomly) chose the topic of *project scheduling*, which gave me the chance to become part of an excellent team of researchers with an unbelievable experience and amazing track record in this fascinating field. And so I became, without real ambition or specific goal, an academic researcher in data-driven project management, and up to today, I am glad and passionate about this research topic. As said, entering academia was not a carefully taken choice, and I even had no particular interest (no knowledge nor experience) in Project Management. It was the interest in *computer algorithms* and the passion for *C++ programming* that accidentally brought me to the exciting job of academic research in project management. I guess life is very different from a project, as it just comes as it goes, without having everything carefully planned. As far as I can predict my own future, I think (and hope) that I will stay in academia for the rest of my professional life, as I simply love this job too much to consider any significant change. Of course, many parts of the job have changed along the years, and also my research focus gradually shifted from a purely *computer science* focal point (using algorithms and data analysis) to a more *management-oriented* approach (searching for relevance and impact in reality). Today, I consider myself more of a *project management researcher* who only relies on computer algorithms and statistical data tools as supportive methodologies to improve decisions for projects, instead of a computer scientist with an interest in algorithms and software tools. Despite these gradual changes, I have never left my first research theme, and my passion for coding *algorithms* for *project scheduling* problems still resonates in many of my research endeavours, most notably in the work of many of my PhD students for whom algorithms and scheduling problems are the only things that matter. You never forget your first love.

After my amazing time as a doctoral student,[1] I wanted to stay in academia and was looking for a job as a young professor. A few months after my graduation, I had the chance to join *Vlerick Business School* (Belgium) as –at that time– the youngest professor of the school, and I chose to combine this appointment with a part-time assignment at *Ghent University* (Belgium). Almost ten years later, I joined the engineering department of *University College London* (UK), and five years later, I switched to the *UCL School of Management*, and I have worked at all those places up to today. The combination between being a university professor and a professor at two business schools has always been attractive to me since both institutes have a somewhat different way of teaching, doing their research and approaching their (theoretical and practical) problems. Thanks to this mixed appointment so early in my career, I constantly had (and still have) to balance between the university goals and the business schools' ambitions, and although they are fundamentally not very different, I quickly learned that there is a sort of a gap, maybe a tension, between

[1] When I recruit young PhD students, I honestly tell them that my PhD period was the best period (1996–2001) of my professional life. During interviews, I sometimes consider taking my electric guitar to sing the song "Summer of 96" (not 69!) and convince them to join my team, but so far, I never did.

the ambitions of the two institutes. While the academic research at the university could consist of any possible theme, I felt the pressure of the business schools to provide practical relevance and impact on society and business. I was not only forced to critically look at the relevance of my theoretical research for practice, but I also realised that practical relevance can only be reached when a profound underlying theory and sound methodology are used. The continuous interaction between the theoretical approach of universities and the practical orientation of business schools has been a life-changing experience which has had a significant impact on my research agenda in the past decades. Today, I look at academic research as a careful walk on the bridge between theory and practice. After all those years, it still feels like walking on eggshells at times, trying to satisfy the ambitions of both stakeholders (*"researchers want new theories and publications, and do not care about the real world"* while *"professionals have no particular interest in academia and want practical results"*) and aiming at bringing these two separate worlds closer together. This dialogue between these two worlds is the story of this book in which *research meets practice*.

Thanks to the experiences and lessons learned as a young professor at the dawn of my career, I knew I had to expand my research horizon beyond my favourite *project scheduling* theme. Around the year 2004, I started an intense but fruitful collaboration with some prominent project management professionals in Belgium that pulled me out of my comfort zone of computer algorithms and C++ coding. The goal was to extend my restricted focus of project scheduling to a broader theme, including *risk management* and *project control*. In my first research project with a professional project manager (Stephan Vandevoorde), the goal was to bring clarity in the often confusing state-of-the-art knowledge on *Earned Value Management* (EVM) and to compare and validate the existing methodologies using sound and proven principles from the literature. It resulted in the comparison article published in 2006 entitled *"A comparison of different project duration forecasting methods using earned value metrics"* (Vandevoorde and Vanhoucke, 2006).[2] This work received an unexpectedly positive attention, not only from academic researchers but also mainly from professional project managers, and it was the first time that I realised that my research could be interesting to people outside academia. It was an amazing sensation I had never experienced before! Given the interest from the professional world, I decided to continue with this new research topic, which has eventually resulted in my first book on Earned Value Management entitled *"Measuring Time"* (Vanhoucke, 2010). As if my hunger for attention was not satisfied enough, the book was awarded by the renowned *International Project Management Association* (IPMA) in 2008 in Rome (Italy), and suddenly, I was no longer an academic with theoretical papers but someone with an interest in *research with relevance for business*. In the years after the award ceremony, everything changed for the better. The collaborations between academia and practice became

[2] My co-author Stephan Vandevoorde has been a partner and friend in this search to practical relevance, and I will come back on our collaboration in Chaps. 3 and 4 of this book.

greater and more intense (resulting in periods that I was abroad for the better part of the year), I founded my own company OR-AS with my friend and business partner, Tom Van Acker, and we worked our way through many interesting projects at various companies in Europe. In 2007, we launched our commercial project management software tool *ProTrack* which was an extended version of our project management business game *PSG* that we introduced two years earlier (Vanhoucke et al., 2005). With little to no time left for academic research, I even doubted for a moment to leave the academic world, but I am glad I never did. The icing on the cake of this intense and sometimes crazy journey to practical relevance was the founding of the *EVM Europe organisation* (2009–2013), in which we brought together researchers and practitioners with five conferences in five consecutive years. After these conferences, I was somewhat stuck between academia and practice with no clear direction, and I thought it was time to explore other horizons. It eventually brought me back to the academic world, and I decided to start a research group of young PhD students and reduce the number of company projects to the absolute minimum to free enough time to guide my new academic team. Nevertheless, I still look back at this crazy period as a time of hard work, many travels, and new friendships, some of them discussed in this book.

Building a new team of young and enthusiastic researchers is not as easy as it may sound. First and foremost, I had to rethink my research strategy, since I was no longer willing to focus only on project scheduling algorithms. Instead, I aimed at working further on the EVM study that attracted so much attention from the real world, but this time, I wanted to expand this well-known project control method to a new and better methodology.[3] With this ambition in mind, I submitted a challenging research proposal, known as a *Concerted Research Action*, with the title *"Searching for static and dynamic project drivers to predict and control the impact of management/contingency reserves on a project's success"*. With a total requested budget of €1.3 million and the promise of writing 8 doctoral theses in the next 6 years, it was the most ambitious project so far in my life. €1.3 million might sound like a reasonable amount of money to a professional, but for an academic researcher, it is similar to winning the jackpot for life. The research proposal passed the first phase and so I was admitted for an oral defence to a board of national and internationally recognised researchers, but – to my disappointment – the proposal was rejected. The reviewers' comments were generally positive, but the mixed review reports indicated that some jury members considered the proposal as *too academic* (not enough practical relevance), while other said it was *too practical* (no sound academic methodology) and so I realised that I really was stuck in the middle of these two worlds, with no clear identity or direction anymore. After a deeper dive into the detailed comments of the reviewers, I noted that the criticised lack of a strong academic foundation that resulted in the rejection of the proposal was mainly due to the way I proposed to use *project data* in my research experiments.

[3] This academic mission of translating existing methods into totally new and improved methodologies is the topic of Part II of this book.

Table 1.1 Stereotypes: academics versus professionals

Academics	Professionals
Theory	Practice
Research	Application
University	Business
Artificial	Empirical
Ivory tower	Real world
Nerdy	Cool
Smart	Handy
...	...

As a matter of fact, I defended the use of a mix of *artificial* and *empirical* project data for all my computational experiments, but some reviewers felt I had to choose. The use of empirical data, some reviewers argued, would have no merit since any empirical dataset would be too small and, therefore, could never be used to generalise any result of computational experiments. Using empirical case study data, they continued, could be good for learning case-specific features of projects that hold only for the small set of data but would never provide the general insights necessary in academic studies. Other jury members, however, stated exactly the opposite and argued that the use of artificial data would cause major problems. They argued that artificial data is often generated in a random way and would therefore never fully capture the real characteristics that typify projects in real life. Hence, they argued that their use would make the results too theoretical, i.e., too far away from any practical relevance. Apparently, even the review team for an academic research proposal consisted of people of both worlds (academia and practice), each with their own desires and ideas of carrying out sound academic research. The gap between academia and practice was now a choice between artificial data generation and the collection of empirical projects. The reviews stated that I had to choose.

However, I chose not to choose. Instead, I decided to fight and accept the struggle between academia and practice by considering the notorious gap between these two worlds as a challenge and strength of my research proposal, rather than an unavoidable downside of my job. A year after the failure, I completely redesigned the research strategy of my proposal, aiming to build bridges between academia and professionals by fully exploiting the gap between academia and practice (which often are referred to as stereotypes, as shown in Table 1.1). I submitted the updated proposal –still suggesting a well-balanced mix between artificial and empirical project data–[4] and after some favourable review comments (for the second year in a row) and a positive oral defence to the board, the jury decided to grant the funding. I am grateful, up to today, to many colleagues at Ghent University who supported my proposal in different ways and helped me to get this amazingly high amount of money to carry out research between 2012 and 2020. This research project has been

[4] The use of both artificial and empirical project data for academic research is the topic of Part IV of this book.

the start of the growth of my team, and without this funding, I would never have been able to do what I have done so far in my career. Today, I have the privilege to work with a fantastic team of people at my *Operations Research and Scheduling* (OR&S) group, including many young and ambitious PhD students from Belgium, China, Iran, and Turkey, two post-doctoral students helping me with the guidance of these students (Annelies Martens and Tom Servranckx), and a colleague professor and friend from Portugal (José Coelho).[5] While writing this book, Annelies decided to explore other places after years of research, so I now continue with great enthusiasm with Tom as my right hand and partner in research crime and José as my research colleague whom I regularly visit when I need sun, light, and pastéis de nata. The results of this new funding were impressive. Twelve years after the first IPMA award, I was granted a second *IPMA Research Award: Outstanding Contribution* award for the research *"Data-driven project management: Research by and for academics, students, and practitioners"* by the *International Project Management Association* in Berlin (Germany). This award was very special, not only because it was given in the middle of the COVID-19 crisis (some good news in-between the continuous stream of negative news), but especially because it was the result of team work with Gaëtane Beernaert, Jordy Batselier, Annelies Martens, Tom Servranckx, and José Coelho (all names will appear in later chapters of this book). This book gives a summary of this awarded research and tells the story of my amazing team.

1.2 Data and People

Working at an academic institute is more than carrying out research, and teaching is one of the tasks I enjoy the most. I have never been a person who could teach a course module based on a student handbook written by someone else. I have to experience everything myself before I can talk about it, and carrying out research and working in companies as a consultant helped me a lot in defining my teaching agenda. Thanks to the mixed appointments, I had the privilege to teach *project management* and *decision making* course modules at universities (master students), business schools (MBA students), and commercial companies (professionals). Depending on the audience, I had (and still have) to balance between focusing on *data-driven skills* (using algorithms and statistical tools) and *people skills* (how to work in a project team). In the beginning of my (teaching) career, most courses focused on the people side of project management. It was a time in which most professionals considered project management primarily as a job of managing people, and I had a hard time explaining why scheduling algorithms and statistical risk analyses might be equally important for managing projects. Today, things have changed a lot, and most students no longer doubt about the relevance of a data-driven view on managing and controlling projects.

[5] If you want to meet these fantastic people, go to Appendix A.

It is true, of course, that you cannot become an excellent project manager without the appropriate people skills, but recent research has shown that the decision making by people does not exclude the necessity of algorithms, statistics, and a sound data analysis. People have experience, have their own way of working, are often very talented and creative, and solve problems in ways no model can do. But people are strange too, often have their own agenda, their biases, the averseness for risk, and so many other (undesirable) traits that might have a (negative) impact on the decision making process for projects. People do not perform well under stress, often envy each other, and mostly avoid any possible change. So why would project management course modules focus only on people skills when it is known that they suffer from so many weaknesses?

Data and algorithms obviously suffer less from these inherent people traits, and statistical analyses are less prone to subjectivity and misunderstanding. While personal opinions might be biased or even bluntly wrong, numbers never lie and might therefore be a useful secondary source to project managers for taking better decisions for their projects. Thanks to the increasing power of computer algorithms and statistical methodologies, and with the help of ever-faster software systems, many of us have now easy access to more and better data analysis tools than ever before. With the increasing availability of data-driven tools and methodologies, the growing importance of a thorough analysis of such project data has redefined many traditional project management courses into *data-driven project management* lectures. In my course modules, I focus on how to rely on project key data to make better decisions for projects in progress using both *computer algorithms and statistical analyses* and the *experience and creativity of people*. Project management no longer is a discipline defined solely by people skills but now requires the integration of data and experience in an integrated decision making methodology. I always start any project management course module with the words of the American engineer and professor William Edwards Deming who expressed the importance of data for business:

> Without data you are just another person with an opinion.

The focus of this book will lie on the data-driven skills of project management and more specifically on the correct and sound analysis of data using algorithms and statistics to improve the decision making process for projects. Although this is *not* a book about the people skills necessary for managing projects, I will often refer to simple *rules of thumb* used by people to manage projects. The underlying assumption of this book is that most of, if not all, the methodologies and algorithms discussed in this book should not be used as a stand-alone *decision making* method, but rather as a *decision support* system to objectify opinions by people (project managers) before a final decision can and must be made.

I will show at various places in this book that the importance of project data, algorithms, and statistics is often not well understood. Much of this discussion dates back to the previously discussed gap between the use of artificial data (by academic researchers) and empirical data (by professionals) as was noted in the review reports of my big research proposal and perfectly summarised in Fig. 1.1. It is known that

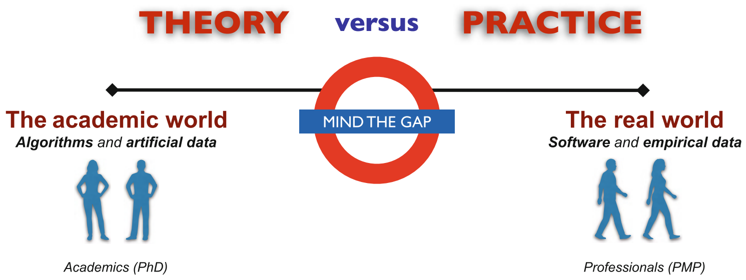


Fig. 1.1 The gap between theory and practice (managing projects)

academics primarily develop new (project scheduling) algorithms as a way to get publications in academic journals. They aim at extending the current state-of-the-art knowledge with more and richer features, but due to the pressure of getting the work published, these extensions mostly contain new elements that seldom result in major improvements to solve real project problems. Many project management professionals (PMPs) are not able, nor willing, to digest the overwhelming overload of new findings in the academic literature and criticise them (often rightly so) for being too complex, too restrictive, and too difficult to be used in a practical setting. Much to the irritation of the professionals, the academic world counters this criticism by stating that business people are not open for new advanced algorithms and criticise the current software tools for being too generic and not project-specific. The academics often argue (also rightly so) that every project is unique and, therefore, requires a tailor-made approach consisting of new algorithms or improved statistical analyses instead of a generic approach. Professionals then reply that these project-specific academic algorithms are only tested on artificial data, with no proof that they could also perform well on their (empirical) project data. This never-ending discussion between academia and business results in misunderstandings between researchers and practitioners in the requirements of proper project data and the necessary algorithms to analyse these project data. It eventually creates a tension between the two worlds, resulting into a growing gap and suboptimal knowledge sharing between academia and practice. I believe it is time to end discussions and talk to each other, and I hope this book will be a helpful guidance to join forces.

1.3 Book Outline

Writing a book is a very personal and intense process in which a thousand decisions about *what to include and what not* have to be made. There is so much to tell that I continuously change my mind about the book content, I keep thinking about the best book structure and specific writing style, and after each draft, I doubt whether I took the best decision. As a matter of fact, I have a *love and hate* relationship with the process of writing,[6] and I always swear that it is going to be my last book ever. I swore that this book too would be my last one, but I do not doubt that I will change my mind, one day, and start all over again. For this book, I have chosen not to go too deep into the technical details, nor to give the readers a full overview of the existing project management methods in the literature. Instead, this book must be considered as a personal reflection of the work I have done with my research team in the past decades. You will see many references to my own research work and only a few references to other people's work. This is a deliberate choice, not to express that other research is not relevant or important, but rather to tell my own personal story of research in data-driven project management. While I will mainly focus on results of research studies, I will also add some personal statements, use quotes from books or websites, and refer to people I have met throughout the years. This book not only tells the data-driven project management story of the OR&S group through my own eyes but also serves as a way of thanking everybody who helped me in this amazing journey to better understand project management. It is organised into four different parts, and a concise overview is given along the following paragraphs.

In Part I of this book, a general introduction and book outline are given. It consists of the introductory Chap. 1 you are reading right now, followed by two other chapters. In Chap. 2, I will give a concise overview of other books I have written in the past about the same project management topic, albeit from a totally different perspective. Furthermore, Chap. 3 describes the three components of *data-driven project management*, which will be referred to as *project scheduling*, *risk analysis*, and *project control* and constitute the central theme of all further chapters in this book.

Part II provides a summary of my recent research studies in data-driven project management. I called this part *"what academics do"* since it describes the results of research studies from an academic point of view, sometimes inspired by practice, but often carried out in complete isolation from the real world. As a matter of fact, the research of this part started with a collaboration with some professional project managers that were in need of better understanding the existing project control methodologies. The positive results of this study quickly became a trigger for wanting more and resulted in many follow-up studies that primarily focused on getting results published in academic journals. During these studies, I was not always interested in developing new practical tools for business, but rather in understanding the tools and extending them to make them better… significantly better. This book part contains three chapters to explain the three important missions of academic research. Chapter 4 explains how academic research can bring structure into the often confusing literature. More specifically, it shows how the existing methods for project control can be compared and benchmarked with computational experiments, offering a better understanding of why these methods work well or fail

[6] One of the most pleasant side effects of writing a book is that it forces you to bring the different pieces of research, published in separate papers, together in one manuscript such that you suddenly see the bigger picture.

miserably. Chapter 5 describes how such benchmarking study allows researchers to extend and modify the existing methods to new and better methodologies. It will be shown that proposing such improvements belongs to the core of academic research, regardless of whether or not these extensions have practical value. Finally, Chap. 6 describes how and why academic researchers sometimes go completely crazy in their search for improvements. Having the time and freedom to explore, they sometimes come up with totally new ideas that are far beyond the current needs of practice. It shows that academic research does not always have to lead to practical relevance and illustrates that topics that are not important now can become a crucial tool in tomorrow's business world. Despite the academic nature of the research of these studies, they have become the foundation of Part III of this book in which I look at the research studies from a practical point of view.

Part III elaborates on the research from a practical perspective. I called this part *"what professionals want"* since much of the research discussed in this part is inspired by numerous close contacts I had with professional project managers. Sometimes without realising, these professionals have profoundly defined and even completely changed my research agenda. Without these business contacts, Part III would have been much shorter, totally different, or even completely void. Now, it consists of four chapters. Chapter 7 explains how the theoretical results of the previous studies in Part II can be used to make project managers more *efficient*. The chapter compares two alternative ways of controlling projects (*bottom-up* and *top-down* control) and shows how they can be used by three different types of project managers. The chapter shows some empirical results and proposes an *integrated project management and control* approach as a best practice. Chapter 8 goes deeper into the details of project control and compares three fundamentally different project control models, each analysing project data in a totally different way. The chapter ranks these three models from easy to hard and argues that the *analytical project control* method reaches almost the same results as the advanced *statistical project control* models but requires much less effort to implement. Finally, Chap. 9 introduces the readers to project forecasting and discusses the well-known *reference class forecasting* model. Based on three research studies, the chapter shows that data from historical projects is key to make better decisions for future projects in progress.

Since the chapters of the previous Parts II and III will clearly show that the availability of project data is key for academic research *and* practical relevance, Part IV contains six chapters about project data. In this part *"about project data"*, it will be shown that data analysis should be treated with care, as these project data are not always clearly defined, readily available or easy to use. Chapter 10 gives a brief summary of the current state in using project data by academics and professionals and introduces the readers to two types of project data. A first type, discussed in Chap. 11, consists of *artificial* project data. It will be shown that the generation and analysis of artificial projects is not as easy as it seems and requires a good understanding and knowledge of the best practices used in academic research. It will be shown that my team has proposed various artificial project databases, each serving a different (academic) research goal. Chapter 12 makes use of the artificial projects of the previous chapter and explains how additional data can be generated to imitate project progress. More specifically, the chapter presents three models to generate time and cost deviations from the project baseline schedule to imitate reality. A second type of project data is discussed in Chap. 13, which gives an overview of how my team collected and analysed a set of *empirical* projects in a time span of more than ten years. Chapter 14 presents a so-called *data calibration method* to define realistic probability distributions for activity durations and costs using empirical project data in order to use the progress models of Chap. 12 in a realistic setting. Finally, Chap. 15 serves as a summary chapter to show the different project datasets made available by my research team for research on project scheduling, risk analysis, and project control.

In Chap. 16 of Part V, I conclude this book by giving my (humble) opinion on the necessary traits an academic needs to have to become a good researcher. Finally, technical details and background information can be found in the Appendix at the end of this book.

1.4 Keep Reading

Before I introduce you to the fascinating world of data-driven project management in Chap. 3, I want to invite you to the personal stories told in my previous books. More precisely, I will give an overview of my previous books on *project management* and *decision making* in Chap. 2. I will provide a brief summary of each book's content (hoping that you are triggered for wanting to read more) and concisely tell you why I wrote each book. You can easily skip this chapter and immediately jump to Chap. 3 that outlines the main theme of this book used in all other chapters.

References

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*, 289–302.

Vanhoucke, M. (2010). *Measuring time – improving project performance using earned value management* (Vol. 136). International Series in Operations Research and Management Science. Springer.

Vanhoucke, M., Vereecke, A., & Gemmel, P. (2005). The project scheduling game (PSG): Simulating time/cost trade-offs in projects. *Project Management Journal, 36*, 51–59.

Chapter 2
Each Book Tells a Story

I was once asked by a colleague why I spend so much of my precious time on writing books on a topic that only interests a tiny little fraction of the people. *"You will never write a best-seller"*, he argued, not understanding that the joy of academic research lies in the process of experimenting, trying and failing, and writing your findings as if they will change the world, even if they do not. I guess what he really meant was that I should realise that my enthusiasm in my research topic (*data-driven project management*) is not in line with the enthusiasm of most people, and therefore, sacrificing weekends and evenings on such a big endeavour with only a little amount of recognition is –according to my colleague– a total waste of time.

I disagree. Of course, I know that there is not a huge Harry Potter-sized audience out there waiting for my next book, queuing in long lines the day before its release at their favourite book store. Writing books about *data-driven project management* is indeed not for the masses, but I can pretend it is. As a matter of fact, each book that I write tells a story for a specific audience, and even if only a few people can benefit from the new story, I feel the urge to tell it. Some books target my students and are used as supportive material for my lectures at the universities, business schools, and companies. Other books aim at summarising my research in an understandable language and target researchers and young doctoral students. The *data-driven project management* story can be told in various ways, and each book is written from the belief that its specific content could not be found in any other book available in the literature. And so the answer to my colleague's question is that I write books because the American author, and Nobel Prize winner (in Literature) professor Toni Morrison, gave me the following reason to do so:

> If there's a book that you want to read, but it hasn't been written yet, then you must write it.

As I already said in the previous chapter, the truth is that I have a love-and-hate relationship with the act of writing. At times, I enjoy it a lot and consider it as the best part of the academic job, but at other times, I swear to never write a single chapter again. But most importantly, I truly think that I do not have much of a

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_2

choice, as writing is in my nature since I was a kid, and I am glad that I have found a job with an audience to which I have something to tell.

Publishing books with your name on them has one other positive side effect: my books have brought me to places I would otherwise never have visited, and I have met people who first were work partners and later became (some of them) close friends. I have seen various places in China due to one of my most technical books, I travelled around in Brazil promoting a book for managers, and I have a very close relation with people from Lisbon thanks to research summaries in one of my other books. As a kid, I dreamt of becoming a rock star[1] to travel around with a couple of friends and make music to perform on stage, but that dream (that I still have) never came true. Luckily, my books are the second best alternative to go on tour and perform (in the classroom, not on a stage). All of my books gave me the chance of wandering around, and I feel very much like the American novelist Roman Payne who wrote in his book *"The Wanderer"*:

> Just as a painter paints,
> and a ponderer ponders,
> a writer writes,
> *and a wanderer wanders.*

I have no choice. I want to wander and so I have to write books.

2.1 Bookstore

Most of my Project Management books have been published by Springer. I had the privilege to work in close contact with Christian Rauscher. and then senior editor of Business, Operations Research & Information Systems at Springer Heidelberg, Germany. We had numerous interesting and joyful conversations during workshops in Europe and the USA about travelling, and we endlessly talked about our shared love for Portugal and – of course – about books. In 2021, Christian brought me in contact with Jialin Yan who took over Christian's work for this current book. I am grateful to both Christian and Jialin for their support and suggestions for improvements. In this chapter, I will briefly discuss the underlying story of each book, why it was written, and the places it reminds me of.

Measuring Time (2010) The book "*Measuring Time: Improving Project Performance Using Earned Value Management*" is my first book ever, and it is the result of a very intense and inspiring research stay in London (UK). I still remember most of the places where I wrote the different chapters, and I keep good memories of the times I was away from my country (Belgium) trying to summarise my ideas into a

[1] All of my friends wanted to become famous football players, but I have never been as interested in sports as they were and so travelling in a bus and performing on stage bringing music was my ultimate dream.

single manuscript. At that time, I was working with a colleague from *London School of Economics* on a research paper in the field of *Management Accounting*, spending hours in small, dark, and cheap hotel rooms in London, programming a simulation algorithm that was eventually published in two flagship journals (*Management Science* and *The Accounting Review*). Despite the nice collaboration with my co-author and the challenging nature of management accounting research, I knew my (research) heart was somewhere else. As from the beginning of my career, I had worked on the *project scheduling* research theme, and I realised during these lonely rainy London evenings that this would always be the main subject of my interest.

So I decided to split my London time in half, spending 50% of my time to the simulation experiments for the management accounting research and the other 50% to applying similar simulation models for project management (which was, without even realising it, the start of writing my first book). I was lucky to meet Stephan Vandevoorde during that period, who was (and still is) a project management professional with an engineering background and an interest in applying academic research into practical applications. He was particularly interested in the well-known methodology for project control called *Earned Value Management* (EVM), which consists of a set of metrics to measure the performance of projects in progress. While the methodology was, and still is, widely used by professional project managers (like Stephan), little to nothing was done in the academic world to investigate *why it works* for some projects and *why if fails* for others. He was interested to find out and suggested to work together on this fascinating topic. Suddenly, the simulation runs on artificial project data in cheap London hotel rooms changed into a collaboration with a professional project manager with access to real project data and so I accepted his offer with pleasure.

I stayed in London for some extra months (still cheap hotels), trying to integrate *risk analysis* (understanding the risk of projects in progress) and *project control* (managing the project risk and take actions when problems occur) into my favourite *project scheduling* research theme. The insights that we got were that the use of algorithms (to construct a project schedule) and Monte Carlo simulations (to analyse the project risk) could be combined with project control methodologies (such as Earned Value Management), and this *baseline schedule – risk analysis – project control* focus became the foundation of all my research endeavours that followed afterwards. Accepting Stephan's offer was the best decision in my academic career.

I will mention the work with Stephan in several chapters of this current book. As a matter of fact, I already mentioned this in the previous chapter, as in 2006, Stephan and I published our first paper in the *International Journal of Project Management* (Vandevoorde and Vanhoucke, 2006), which has become our most referenced article. Our second paper consists of the simulation study carried out in London and compares three EVM models for project duration forecasting which will be outlined later in Chap. 4. This study was published in the *Journal of Operational Research Society* in 2007 (Vanhoucke and Vandevoorde, 2007). One year later, I decided to submit this and much more research work to the *International Project Management Association* for a research award, and they granted me the *IPMA Research Award* in Rome (Italy). When I had to present my research in Rome to a room full of professional project managers, I suddenly realised that this research had practical relevance, and so, while I was talking to that specific audience, I decided to write my very first book with a summary of all the work done. The book *"Measuring Time"* was published two years later by Springer (2010), and up to today, I believe much of the chapters are still highly relevant, as they are used as the foundation of the chapters of the current book.

I tell this story to show that my initial search for practical relevance that initially brought me to *management accounting* research and then pushed me back to *project management* has been a search full of coincidences and random choices. I had no intention whatsoever to become a writer, but it just happened as a side effect of the intense London period. I remember the writing period as regular travels between Brussels and London at late hours (cheaper tickets) and dark and uncomfortable London hotel rooms, having discussions with people from everywhere in the world, and I somehow knew that it would define the future of my career. Above anything else, I remember it as a period where Stephan and I were looking enthusiastically to some topics to investigate, in which we turned from colleagues to friends for life. No wonder I will cherish this book forever.

For those who are interested in reading the book, I have to warn them that it is not written for dummies. Instead, the readers are assumed to have a (relatively) strong background in project scheduling and control, with an affinity for algorithms, statistical data analysis, and Monte Carlo simulations. It is a book written for researchers with an interest in technical details, and it contains so many detailed descriptions of project forecasting formulas, project data generation methods, and simulation algorithms, which I do not advise to read it on a Sunday morning. Despite that, I still believe that – after all these years – the book is still worth reading if you like a challenge, and I am still proud of it.

Dynamic Scheduling (2013, 2nd Edition) Unlike my other books, there is no particular time and place I can recall for writing my second book. For the book titled "*Project Management with Dynamic Scheduling: Baseline Scheduling, Risk Analysis and Project Control*", I had a totally different audience in mind, and it took more than 10 years to finish it. After my PhD graduation, I had the privilege to immediately start teaching a Project Management course module at Ghent University (Belgium) for Business Engineering and Civil Engineering students. Initially, I had not much material of my own, and I had to fall back on short articles and easy-to-read academic papers as learning material for my students. As years passed by, I adapted and improved some of these articles and then wrote my own short articles and shared them with my students, until everything became a mess without much structure. I knew I needed a student handbook.

I searched on the Internet for a good book, and despite the plenty of excellent books on Project Management, no book covered the topics the way I discussed them in my lectures. I decided to write my own student handbook focusing on the integration of *scheduling*, *risk*, and *control*.[2] I referred to this integration as *dynamic scheduling* which I borrowed from a book with the same title by Uyttewaal (2005). Up to today, this book is still used as optional background material for my project management lectures, and I have the (nice) feeling that my students are happy with it.

Ten years to write a book is ridiculously long, but it has been a gradual process with no apparent ambition but with a lot of implicit feedback from my students. The book started out as a collection of small articles for my students that gradually grew into longer chapters based on their comments and feedback. I made references to my course slides, augmented with exercises and countless examples, until I realised I was working on a book. The day I finally decided to put everything I have collected over the years in one book was when I saw a group of students in Lisbon (Portugal) reading some of these articles during one of my course modules. When I asked them where they had found this material, they told me that some Belgian students had sent it to them. Not realising that students from two countries interacted for my course modules, it was a great feeling to know that the course material was relevant to people with different backgrounds and aspirations. The first version of the book was published in 2012, and the updated second version was published in 2013, mainly correcting some errors and concerns of my students. I owe them a lot.

Theory Meets Practice (2014) With my two project management books – one for researchers and another one for my master students (at the university) – I thought that was it. I had nothing more to say, but my MBA students (at the business school) thought differently. Any management course module leader has experienced the difference between university lecturing and MBA teaching. The (young) university students have no experience in management and therefore care more about the technical details (which are tested on the exam) and focus less on the practical relevance and implementation issues. MBA students are often not very interested in the theoretical foundation and underlying assumptions of the project management methodologies but are eager to learn how they can implement them for managing their own projects.

I started teaching Project Management course modules at different business schools in 2010 (10 years after my first PM lectures at the university), and I quickly realised that my previous student handbook (*Dynamic Scheduling*) was good for university students, but not very relevant for people with practical experience and an interest in implementing PM methods in practice. Too many technical details and too little advice on how to implement them, that is what I thought. But I was wrong. As a matter of fact, many of my fantastic MBA students contacted me after my lectures with detailed questions about the PM methodologies. They were working on self-made spreadsheets to monitor and follow up their projects, and as they wanted to use the course topics for their own projects, they needed more details. They

[2] You might recall that this integration comes from the research I have done with Stephan in my first book "Measuring Time".

told me that implementing the PM methodologies discussed in class could only be correctly implemented in their spreadsheets when they had a detailed knowledge about the specific formulas and underlying statistical assumptions. They told me that I convinced them during the course module that many of the PM methodologies were useful in practice by *not* focusing on the technical details, which is why they now needed these details for their spreadsheets.

It came to a surprise to me, but although my previous student handbook contained the right level of technical details, they wanted more of a stepwise detailed description about how to implement these different formulas and methods into a spreadsheet to be used for managing real projects. *"If we better understand the theory"*, they argued *"we can implement it much easier in our own spreadsheets"*. And therefore, upon specific request of my MBA students, I decided to write a third book, focusing solely on the formulas and technical details, leaving every other topic untouched.

The book does not tell much about the construction of a project baseline schedule and relies on the easy *critical path method* to schedule the project activities. Instead, the focus lies on the use of this baseline schedule for performing a *schedule risk analysis* and monitoring the project progress using *Earned Value Management*. This integration of *baseline scheduling*, *risk analysis*, and *project control* is of course identical to the theme discussed in my previous books, but this time, I referred to it as *integrated project management and control* to highlight the importance of integrating these three components. I used my students' positive criticism as the subtitle of my book, i.e., *first the theory* and *then the practice*.

It took me quite some time to write down all the detailed calculations without any error. I made use of three artificial projects with a totally different network structure to illustrate how all the calculations should be performed. To avoid errors (I had many in the early drafts of the book), I asked help from some of my university students at the Engineering Department of University College of London (UK). Our Friday afternoon discussions at the UCL Bloomsbury Campus became a weekly habit for some months, and I guess that 2010 was the end of a period where pen and paper discussions were considered more interesting than Facebook and Instagram distractions. I still look back with joy to our coffee breaks (only two pounds for a huge cup of delicious coffee) and discussions about every little step of calculating activity sensitivity metrics and earned value performance metrics by hand.

Technical Sourcebook (2016) Input from my students has always been an important inspiration for my books and so it was for this fourth book, albeit in a somewhat different way. This time, the student input was not coming from my students in the classroom, but rather from anonymous people (*students? professionals? lecturers?*) everywhere in the world. It all started with the increasing success of the online learning platform PM Knowledge Center (www.pmknowledgecenter.com) that I developed for my Project Management course modules. The website contains a series of small articles on project scheduling, risk analysis, and project control, which were massively visited by my students, sometimes reaching over 10,000 visits per month. I wondered where all these visitors came from (*only from my students?*)

and so I decided to put my email address on the website, which suddenly resulted in an explosion of questions, comments, and remarks from everywhere. Among the many interesting comments, the one that kept coming back was the request to add sample questions to test the student's knowledge and understanding for each article. That was easier said than done, but it was the trigger for another book.

Each chapter of my fourth book consists of a set of short stand-alone articles, containing a summary of maximum three pages about one little relevant topic. Each article is linked with other articles of the book, such that it becomes an integrated piece of work on project scheduling, risk analysis, and project control. All articles are highly technical, contain lots of details to improve understanding, and are accompanied by a set of questions (open questions or multiple choice) and answers (do not look at them, try to solve the questions first). I have no idea whether this book is used a lot. All I know is that I see my own students using these questions to test their knowledge about my PM course. On a certain day, I even saw a group of students playing a Q&A quiz about project management, and the quiz master used the questions of the book to interrogate the student audience. How nice!

This fourth book called "*Integrated Project Management Sourcebook: A Technical Guide to Project Scheduling, Risk and Control*" was the most difficult book to write. It took me a lot of effort to structure the book in a good way and to come up with questions for each article that can be solved as stand-alone questions without the need of having to read the other articles. It is not a book that you will read from start till end, but rather a reference book on project scheduling, risk, and control that can be used as a reference guide and supportive tool in the classroom. With the more than seventy articles and a huge number of short and to-the-point questions and exercises, I believe the book can be an interesting source for students (and maybe professionals) when studying the technical aspects of project management. Despite the sometimes frustrating process of finding good questions for each book article, I enjoyed the writing process because it was done at my all-time favourite place on the Earth. I wrote this book when I lived in Lisbon (Portugal) in 2015, and since I had been in love with Portugal's city of light for decades, this is what I wrote in the book's preface:

> This book has been written in the sunlight of Lisbon during my four months stay at the city of light. While artists say that light is all important to creating a masterpiece, I just think back on it as a period where I enjoyed writing in my apartment at Beco da Boavista, on the terraces of Jardim da Praça Dom Luis I (my favourite one, I called it the red terrace), Praça do Comércio and Portas do Sol but also on the Miradouro de Santa Catarina, the city beach of Cais do Sodré and of course at Universidade Aberta de Lisboa. In fact, it is my stay at the city that has become the masterpiece, while the book is simply the result of hard work in complete isolation from all Belgian distractions.

Business Novel (2018) After four technical books about data-driven project management, I thought it was time for something else. A year after my summer in Lisbon in 2015, I decided to go back in 2016 and stay there for another summer of work. This time, I wrote a technical business novel titled "*The data-driven project manager: A statistical battle against project obstacles*". Portugal's city of light never disappoints. I was completely alone for a month (my wife had to go back for the exams of the kids) and so I woke up every morning before sunrise to start writing on the amazing sunny terrace of my apartment, with a cup of strong coffee (black, no sugar), some good music in the background, and my laptop in front of me. It was an amazing feeling not only because I was in the city of my dreams but also it felt like a dream coming true, almost being a *real* writer, inventing a story instead of summarising research results from the academic literature.

I call the book a *business novel* since it is written as a narrative telling the story of a company that wants to install a data-driven methodology for managing its projects. It is a story not only about people who struggle with the new concepts but also about other people who cannot wait to implement the new system as soon as possible. It combines the explanation of the statistical methodologies (i.e., sound academic research) with the complexities and difficulties to use them in a real business environment (i.e., the practical relevance) in a single story of a fictitious company. More specifically, the company called *GlobalConstruct* is responsible for a tennis stadium construction project in Australia. The book tells the story of Emily Reed and her colleagues who are in charge of managing this tennis stadium project. The CEO of the company, Jacob Mitchell, dreams of planning to install a new data-driven project management methodology as a decision support tool for all upcoming projects. He challenges Emily and her team to start a journey in exploring project data to fight against unexpected project obstacles.

Storytelling may be trendy these days, and I believe that I added my own contribution to this emerging trend with this technical novel. While the book is not a real novel, it nevertheless tells the story of why some PM methodologies work for some projects and why they sometimes miserably fail for other projects. I think this storytelling works perfectly in the classroom, since most chapters of this book are used in my PM course modules at Vlerick Business School (Belgium) and UCL School of Management (UK) in a very intense five-day course, in which the students get one case study per day (coming from the book chapters) to solve a project management problem. Students have to work in teams on each day's problem, and student evaluations tell me they love it, which illustrates that telling stories helps in teaching (complex) topics in a joyful way. Apart from the nice Lisbon experience while writing this book, it is also dedicated to Thierry, my wife's brother, and Koen, my best friend, who are unfortunately no longer among us, which is another reason why these memories are so special to me. If you have a passion for project management, an appetite for decision making, and an affinity with numbers, then I invite you to read this book.

2.2 Only a Click Away

I feel very fortunate to have found a job I really like. My passion for research and love for teaching have brought me at places I have never seen before, and I have met students all over the world from which I learned as much as they learned (hopefully) from me. I could not choose, if I had to, between my passion for research and my love for teaching, and I consider my books as the best alternative to work on both (since the content is the result of my research written down for supporting my lectures for my students). However, at sometimes, I feel the urge to tell something more than just the content I always discuss. Instead, I believe that taking a look behind my research and teaching activities and writing about the endless passion I feel for my work might also be interesting. That is why I have written some books that try to do this, and they are completely free to download from www.or-as.be/books. You can share my passion for Project Management by posting a reference on LinkedIn so that other people who follow you can get excited too. These books tell the same PM story as my other books, but from a completely different point of view, with many more stories and anecdotes than just focusing on the content.

Work and Passion (2021, 6th Edition) Everybody knows it: life begins at 40! On 14 March 2013, I introduced the first edition of "*The Art of Project Management: A Story about Work and Passion*" on the OR-AS website.[3] On that day, it was my 40th birthday, and after a keynote lecture at a PM workshop, someone asked why I always talk about the PM profession with great passion, without telling *why* I am so passionate about it. *"You are so amazingly enthusiastic when you talk about your job"*, she told me in front of the audience, *"that I want to know what keeps driving you!"* It was undoubtedly not only the strangest but also nicest remark I ever got after a presentation and so I decided, then and there, to write a book about it.

The book contains stories about friendship, crazy ideas, hard work, and research results in the field of Project Management. It gives a look into the endeavours done in the past and the ideas that will be done in the future. It tells about the products and ideas of OR-AS and provides a brief overview of the most important people who inspired me in my research, consultancy, and teaching. It tells a story about the work, and the passion, that has brought me where I am. It is not a scientific book. It is not a managerial book either. It is just a story . . . about work and passion. I regularly update the book when there is something more to tell. The 6^{th} edition of the book is now available for a free download, and I hope I will have to update the book for many more years to come.

[3] OR-AS is an abbreviation for Operations Research—Application and Solutions and is the name of the company that I founded with my best friend Tom Van Acker in 2007. After 10 years of hard work and fun, we decided to shut down our activities to explore other, more interesting territories, but we stayed best friends until today.

Decision Making (2017, 2nd Edition) The book "*Taking Sound Business Decisions: From Rich Data to Better Solutions*" is written for anyone with an interest in modelling and decision making. It is (so far) my only book outside the field of Project Management, although the topic of *decision making* is applicable in *project management* too. Since management is all about making sound decisions using a mix of data, models, experience, and intuition, this also holds for managing projects. I wrote this book especially for my MBA students at Vlerick Business School (Belgium) and Peking University (China) where I teach the course *Decision Making for Business*, but I have been using it also in other *Operations Research* course modules at Ghent University (Belgium). The book consists of three parts. The first part contains a technical summary of data-driven modelling techniques including *linear and integer programming*, *decision tree analysis*, and *Monte Carlo and discrete event simulation*. The second part shows some examples of real applications from my own consultancy experience to highlight the relevance and usefulness of the different modelling techniques. Finally, in a last part, references to non-technical and popular science books that inspired me a lot are added, including some of my all-time favourite authors such as Nicholas Taleb, Daniel Kahneman, Richard Dawkins, Jared Diamond, and many more. Since I believe that reading books outside my expertise makes me richer (not financially of course), I thought that adding them could be inspiring for my students. As a matter of fact, I seldom trust anyone who does not carry a book, and my book is a call to the younger generation to put the smartphone aside at some times and consider reading books again.

Software (2010) The book "*Dynamic Scheduling on your Desktop: Using ProTrack 2.0 developed by OR-AS*" written with co-author Tom Van Acker is not a book with research results on project management, but rather a software tutorial for our PM Tool *ProTrack* that we developed for OR-AS. The original idea of developing our own PM software tool dates back to a vacation in France in 2004 where Tom and I decided to start our own consultancy company aiming at convincing companies that data-driven project management is the way to go. Six years and thousands of lines of code later, we introduced *ProTrack* to the PM world. It has been the start of collaborations with different companies in Belgium, the Netherlands, and the UK, and despite the joy we had during the most intense period of my career, we also realised much later that maintaining a software tool (with requests from customers for updates and extensions) is not the path we both strived for. Today, *ProTrack* is no longer available as a commercial software package, but I believe the book is still relevant if you want to learn more about data-driven project management. You can judge yourself. Just download it.

2.3 Keep Writing

Each time I am writing a new book, I go through a process of enthusiasm and apathy. During the periods of enthusiasm, I surf on an endless stream of positive thoughts and write dozens of pages a day. During the periods of apathy, I throw away most of the writings realising that it does not make much sense, and then I lose interest, start doubting about the relevance of another book, and decide quitting the whole project to spend time on other, more interesting things. These alternating periods of creativity and lack of interest keep coming back during the whole writing process, and I consider myself lucky to not have a very good memory, since I mostly forget the bad periods, and only remember the good periods after the publication of each book. As a matter of fact, at the final stage of each book, I promise myself it will definitely be the last one, but I keep this promise only until I enter a new enthusiastic phase (I have many of these in my job) with too many ideas for another new book. And then I decide to write a new book, and the process starts all over again.

It maybe has something to do with a lack of trust whether the new book will be different from my previous books. Indeed, all of my books are about the *data-driven project management* topic (a research field I will formally introduce in the next chapter) and so they are – in a certain way – somewhat similar (cf. summary Fig. 2.1). Despite the same general theme, I still believe that each of my books has a specific reason for existing and tackles the topic from a different angle as I clarified in this chapter. I have been lucky to get numerous positive comments from my readers around the world who told me how some of my books changed their (academic) careers. A few people however could not resist to tell me that one book or another was not what they expected (and therefore, as they conclude, is not a good one). I should be happy with the positive comments and ignore the negative ones, but it is not in my nature, or in my capacity, to do so. Whenever I hesitate to continue writing, I think of Salman Rushdie's famous quote and that is what drives me to do it anyway. The quote goes as follows:

> A book is a version of the world.
> If you do not like it, ignore it, or offer your own version in return.

I have read myself many project management books, and while most of them were very well written and highly interesting, I could not find a book that discusses the project management topic the way it does in this current book. And that is why, my dear readers, I offer my version in return.

Fig. 2.1 My data-driven project management bookstore

References

Uyttewaal, E. (2005). *Dynamic Scheduling With Microsoft Office Project 2003: The book by and for professionals*. Co-published with International Institute for Learning, Inc.

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*, 289–302.

Vanhoucke, M., & Vandevoorde, S. (2007). A simulation and evaluation of earned value metrics to forecast the project duration. *Journal of the Operational Research Society, 58*, 1361–1374.

Chapter 3
The Data-Driven Project Manager

If you have carefully read the previous chapters, then you should know by now that the main theme of this book is referred to as *data-driven project management* (DDPM). It is a general framework that consists of a collection of methodologies using project data, computer algorithms, and human intuition to manage and control projects under uncertainty. The framework can be used as an *integrated decision making methodology* to monitor the performance of projects in progress and to take actions to maximise the chances that they finish on time and within budget to successfully deliver them to happy clients. It will be shown that the framework consists of three major components, which will be referred to as *baseline scheduling*, *schedule risk analysis*, and *project control*. This chapter reviews the specific details for each of these components and discusses their place in the DDPM framework. This integrated framework is also known in the literature as *dynamic scheduling* or *integrated project management and control*.

3.1 Three Components

Students, professors, and managers are all sensitive to hypes and trends. Hypes come in various shapes and colours and often have profound effects, but mostly do not stick around very long. They often sound tempting, and if they work elsewhere, it is believed that they should work with us too. Some of them come with no good reason and go without leaving any trace. Others stay longer, eventually find their way in companies and universities all over the world, and fundamentally change the way courses are taught and business processes are managed. Some hypes indeed become trends and lead to lasting changes. It happens everywhere, and it happened in project management too: the *data* hype became a long-lasting trend.

In the beginning of my career, I taught a project management course module at Ghent University (Belgium) for master students with an engineering background,

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_3

and I focused very much on a well-designed mix of people skills (the *soft* skills,[1] 20% of the total time spent) and statistics and algorithms (the *hard* skills, 80% of the time spent). While most of my (engineering) students were very enthusiastic about this course module and really loved the lectures on statistics and algorithms for managing *real* projects, some of them were somewhat disappointed that I did not get deeper into the people skills of project management. I told them that I do not know much about leadership, communication, and teamwork topics, and they should follow a *human resource management* course module instead.

In order to warn my students in advance that the importance of people skills does not constitute the main theme of my project management course module, I renamed it into *"Project Management using Dynamic Scheduling"* to highlight my focus on data-driven methodologies for management projects. I borrowed the term *Dynamic Scheduling* from a book written by Eric Uyttewaal (Uyttewaal, 2005) and used it in the title of my student handbook that I discussed in the previous chapter. This course is still running twice a year at two different universities and now attracts students from engineering, economics, pharmacy, biology, law, sociology, and communication sciences. It is surprising to see that most students – even the ones with no strong mathematical background – become very enthusiastic about using advanced statistical methodologies in project management, and it makes me happy that I can share my knowledge and passion with them, even though I realise that most topics are not easy to grasp. An Erasmus student of mine at my Project Management course posted a LinkedIn message that I must share, with pride and joy, in my book. Beatriz Seabra Pereira attended the course as part of the exchange Business Engineering programme at Ghent University (she comes originally from the Faculdade de Engenharia da Universidade do Porto (Portugal)) and she wrote:

> The Project Management world has fascinated me for some time and, luckily, this semester I had the unique opportunity to attend the Project Management course taught by Professor Mario Vanhoucke at Ghent University. It exceeded my expectations! I have learned how to apply many useful tools and techniques and, it also gave me a deep understanding of their limitations, which allowed me to gain sensitivity and critical spirit to make the best choices when managing projects. My interest in this world grew and I must say Professor Mario was highly responsible for that. It is rare to find someone so passionate about his work as the professor is about his. I am not able to express in words the emotion and the enthusiasm that the professor reveals when teaching this course. He engages every student in the audience and it's impossible not to be dazzled by the world of project management! Thank you very much, professor! I wish some day I find my passion too!

I immediately replied *"This is why I teach. You've made my day!"*.

[1] On a conference on *"skills for managing projects"*, some human resource management researchers told me that I should never refer to *people skills* as *soft skills*. Apparently, for some (to me) unknown reason, people skills sound much better than soft skills. I feel a bit guilty, as I wrote a paper with co-author Tom Servranckx on people skills in project management in which we made a clear distinction between the *hard* and *soft* skills (Servranckx and Vanhoucke, 2021). I did not know any better.

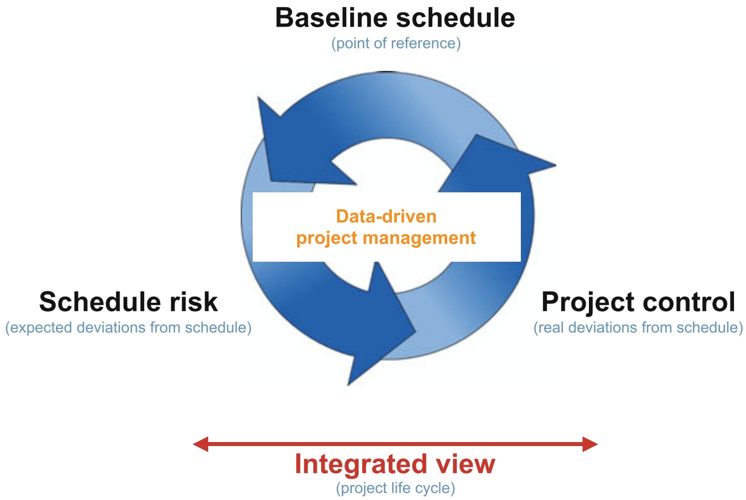


Fig. 3.1 The three components of data-driven project management

Thanks to the increasing attention on data science, the growing number of students, and their positive reviews, I was asked a few years later whether I could organise this course as a commercial training at business schools to students with at least three years of business experience. I was told not to call the course module *dynamic scheduling*, since *no one knows what it really means*, even though the programme director told me not to change the content (since, she argued, this course module focuses on quantitative skills that have become increasingly important in the past few years[2]). I teach this course for some years now, and it had different names (following the hypes of the times), ranging from *"statistics in project management"* (20 paying students, 1 edition per year) to *"project risk management"* (30 students, 2 editions per year), and eventually now called *"data-driven project management"*, resulting in 40 students (maximum allowed) and three editions per year. Hypes and trends, they have an impact on how many people I see in my classroom.

As said earlier, the *data-driven project management* concept is used to refer to the integration of three components necessary to manage projects. Such decision making framework involves taking timely decisions when projects in progress are in trouble in order to deliver them successfully, within the agreed time and cost, to the project client. The three components are known as *baseline scheduling*, *schedule risk analysis*, and *project control* as shown in Fig. 3.1. These three components will be used as the foundation of the whole book and will therefore be briefly explained along the following paragraphs.

[2] It is strange how times change. 20 years ago, program directors warned me to spend a little more time on people skills, and now, 20 years later, I have to warn my students that algorithms and data have no value without the right people skills. *The times they are a changin'*, Bob Dylan would sing.

Baseline scheduling is the act of constructing a timetable that provides start and finish times for each activity of the project. The timetable must take the precedence relations between activities, the limited availability of resources as well as other project specific characteristics (such as activity constraints, due dates, etc.) into account and should aim at reaching a certain scheduling objective, such as the minimisation of the total project duration, the optimisation of the project cost, or any other possible objective.

Risk analysis refers to the analysis of the strengths and weaknesses of the project baseline schedule in order to obtain information about the sensitivity of the project for schedule disruptions. It is often referred to as *schedule risk analysis* to recognise that the project baseline schedule is nothing more than a predictive model of the project progress assuming that *nothing will go wrong*. Risk analysis then uses this baseline schedule and incorporates possible unexpected events (*assuming something will go wrong*) into the project schedule and analyses the impact of these events on the project schedule objectives.

Project control is the process of monitoring the project progress and measuring the (time and cost) performance of the project at different stages during its progress. These performance metrics act as signals that should warn the project manager whether the project progress is acceptable or not and serve as a trigger for making well-informed decisions. More specifically, when the project performance tends to go out of control, these warning signals act as triggers for taking corrective actions to bring the project in danger back on the right track.

In the next section, it will be shown that the *baseline schedule* has a central place in the *dynamic scheduling* framework of Fig. 3.1, as it will act as a *point of reference* for the *risk analysis* and *project control* components.

3.2 A Reference Point

When I talk to professional project managers about new findings for my favourite research theme (*"algorithms for project scheduling"*), I seldom get the same enthusiasm as I got from Beatriz (not even close). Simply ask any professional project manager how much time they spend on the construction of a project baseline schedule, and you will hear words like *"I waste too much time on it"* or complaints like *"the software is crap"*. I have had many discussions with project managers who said they no longer want to lose time on the construction of a project plan, and almost no one is happy with the commercial project scheduling software tools they use (mostly MS Project). Professionals often consider the scheduling process as a huge waste of time and consider it a necessary evil and/or unrealistic exercise. In their mind, project scheduling is nothing but a mandatory, often useless part of the project life, and when they argue that *"the real project progress will be totally different than the plan anyway"*, I often fear that the algorithms I develop are totally useless in the real world. At least, that was the story before *data science* became the new trend in business.

Ask these same professionals about the importance of *risk analysis* and *project control*, and you will hear a totally different story, this time full of enthusiasm and an eagerness to learn more. As a professional project manager, you understand that unexpected events (i.e., changes compared to the baseline schedule) pop up everyday, and some might bring the project in real danger. As they know that risk constitutes the biggest threat for the project's success, they understand that the essence of project management is to cope with the continuous stream of changes in a timely way such that problems can be solved before they go completely out of control. Hence, understanding the main sources of potential risks and monitoring and controlling the deviations between the plan and reality define the real job of a project manager.

It is true, of course, that the construction of a *baseline schedule* can never be a goal on itself, but that does not make it useless. The *data-driven project management* framework recognises the limited role of the baseline schedule and states that the construction of a baseline schedule does not necessarily have to be a perfect prediction of the possible project outcome. Instead, the baseline schedule can and should only be used as a well-defined *point of reference* for understanding the impact of sources of risk and monitoring the differences between the schedule and the real progress of the project. Consequently, the *baseline schedule* acts as a reference point for the two other components, *risk analysis* and *project control*, of Fig. 3.1. The need for a *reference point* is not new and is relevant in any area outside project management. In their amazing book *"The art of possibility"*, Benjamin Zander (conductor of the Boston Philharmonic) and co-author Rosamund Stone Zander state that the main reason atonal music (music with no home key) never developed into a universal art lies in the lack of a sense of destination, as they write the following:

> How can you know where you are unless you have a point of reference?

A project is a *journey* with the schedule as your roadmap to find your desired destination. While the schedule only serves as a prediction for the *expected* project outcome (i.e., the *point of reference*), the *schedule risk analysis* and *project control* phases show you how to reach the *best possible* outcome (i.e., the *destination*). Without the schedule, you will never know where you are (current progress) or where you are going to (destination). In a similar often-cited but not-quite-accurate quote from Lewis Carroll's classic children's tale *Alice in Wonderland*, the following is stated:

> If you don't know where you are going, any road will take you there.

The *data-driven project management* framework does not only tell you where you are, or where you are going to, but also defines the best possible path to your destination. This path of the project journey can be best illustrated by the so-called *project life cycle* of Fig. 3.2, which displays the different phases of any project from start (the conceptual phase) to end (the termination phase, i.e., delivery to the client). The figure shows the three components of the proposed framework, classified into

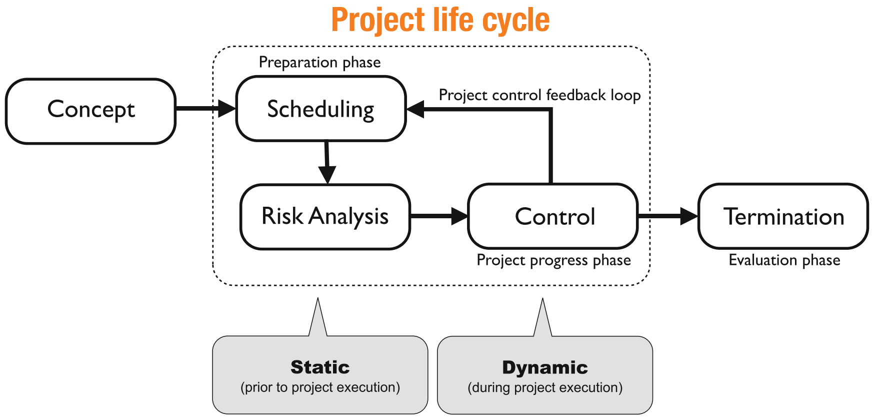


Fig. 3.2 The project life cycle (static and dynamic phases)

two different categories (static or dynamic) which will be used in all other chapters of this book and briefly explained along the following lines:

- **The static phase** is used to refer to all the preparatory work that should be done prior to the start of the project. The construction of the *baseline schedule* is considered as static since it must be done before anything else can start. The *schedule risk analysis* is also said to be static, as it consists of a quantitative analysis of all possible unexpected events that possibly change the project schedule before they actually happen. However, such risk analysis can and should be repeated along the project progress and can therefore also be classified as *dynamic*. Nevertheless, since such analysis is always done *before* the real risk happens, it is better to classify it as completely *static*.
- **The dynamic phase** refers to all the work that must be done during the project's progress. Since project control consists of periodic and repetitive measurements of the project progress at various stages of completion, it undoubtedly belongs to the dynamic phase. This phase is characterised by the repetitive nature of the work to be done. Indeed, unlike the construction of the baseline schedule, the project control phase is repeated at different stages of the project, each time updating the current performance with new information from the real project progress. This repetitive nature, as you will find out soon, has a huge impact on the analysis of the project data.

The classification between *static* and *dynamic* might look like an artificial and purely semantic categorisation without much value, but the contrary is true. Recall that the static work consists of the preparatory work, which consists of all the work that must be done before the project starts. Experimental analysis has shown that this takes sometimes up to 40% of all the project work, and it must therefore be done in a very careful and peculiar way, paying attention to all the possible details without being afraid of spending some extra time on collecting, updating, and analysing additional data before going to the final baseline schedule approval. Since the basic schedule only acts as a reference point for the other two components, it must be created with care, trying to make it as realistic and accurate as possible. After all, once the project has started, no further changes are (ideally) made to this preparatory work. The dynamic phase, however, is different and consists of the collection, analysis, and interpretation of project data at regular time intervals. Thanks to its repetitive nature, scrutinising all the details is less important or simply too time-consuming. Besides, pin-pointing every little detail would make project control practically impossible since it would distract the project manager from the real work to be done (which is drawing conclusions from the project control data and take timely actions to solve the problems in the project). While this distinction between the desired level of detail in the project data analysis might look obvious, I will discuss in other chapters of this book that it has a huge impact on the way the project data will be collected, analysed, processed, interpret, and used for making decisions. A short sneak preview on the *beauty* and *danger* of too many/little details is given in the next section.

3.3 The Beauty of Details

In academic research, details matter the most. Academic research is working on the frontier of knowledge, trying to dive into the tiniest little details of a well-defined topic and exploring possible areas for improvements, however small they are. When the aim is to gradually increase the current state-of-the-art knowledge in a stepwise fashion, the accomplishments are hidden in the details. With the pressure of publishing in high-ranked journals, academic research has become so highly competitive that it requires a certain drive and dedication that you also find in professional athletes. The analogy between academic research and sports has been made in different articles, and I think the analogy is true because of the importance of searching for marginal improvements by paying the utmost attention to all the details. The famous American basketball player and coach John Wooden clearly expressed that improvements can only be made in small steps, one by one, and that they can only be made by focusing on the details:

> It's the little details that are vital.
> Little things make big things happen.

For a professional (project) manager, details are often less important since they consume too much time or do not add enough value. Not seeing the bigger picture might be harmful and eventually might lead to poor decisions, missed opportunities, or even catastrophes. Professional project managers are prone to have a clear view on the bigger picture and often consider micro-management to have a negative connotation, showing too much attention to details, leading to a lack of freedom and trust in the workplace. The best managers – so it is often said – are the ones who make quick decisions by zooming out from the current situation and take a helicopter view on the project. With such a view on the project, these decisions are no longer obstructed by small unimportant issues and day-to-day tasks, which generally leads to a better problem solving approach. The benefits of setting the details apart to focus on the bigger parts is clearly expressed by the American biologist Leroy Hood who stated:

> If you just focus on the smallest details, you never get the big picture right.

Setting the right level of detail clearly depends on the goals one wishes to reach, and this is also the case for managing projects with the three components of dynamic scheduling. More particularly, the difference between the *static phase* (project preparation) and *dynamic phase* (project in progress) of the project life cycle is defined by the required level of detail during the project analysis. Indeed, both phases make use of a different set of project data and require decisions to be made for different time horizons and under a different time pressure, and all these differences have an impact on the appropriate level of detail. Recall that the static phase is used to refer to the work that should be done prior to the start of the project and primarily consists of constructing a baseline schedule and analysing its risk. Since this preparation is done prior to the project start, it should be done in a careful and peculiar way, paying attention to all the possible details. One should not care too much about spending some extra time on collecting and updating data since the baseline schedule is made as the point of reference for the rest of the project phases. There is no real-time pressure, and you better think twice before you get your schedule approved. The dynamic phase, on the contrary, works under a much shorter time horizon, as it refers to the periodic and repetitive measurement done along the project progress. Since this phase consists of the collection, analysis, and interpretation of project data at regular intervals, less details are obviously desirable compared to the one-time preparatory static phase data analysis. Pin-pointing all details as is done in the static phase might be desirable from a theoretical point of view (staring at details probably provides more accurate data), but this is practically impossible due to the repetitive nature of project control. Often times, the project manager must make decisions under time pressure, and focusing on every little detail would lead the manager too far from the real task (i.e., making good and timely decisions when the project is in trouble).

The distinction between a high level of detail at the static phase and a much lower level at the dynamic phase might seem obvious, but it is quite controversial in the professional project management literature. No one argues about the necessity of details for the baseline schedule construction, but there is no general agreement on the appropriate level of detail for the dynamic project control phase. Some believe that project control should focus on some basic project key performance metrics (no details), while others think a control system should consist of a detailed schedule control system. The *Earned Value Management* (EVM) methodology is a typical project control system that consists of a set of key performance indicators to provide project managers with easy-to-understand warning signals to tell whether the project is in danger or not. Ideally, this system only gives warning signals when project problems arise, which should then trigger the project manager to perform a detailed search to identify (and preferably fix) the cause of the problems. Consequently, EVM is a system that should be used as a top-down methodology, not spending too much time on detailed control to report problems, but going deeper into the activity network and resource efficiency details when problems must be solved. This top-down approach has been beautifully expressed in a paper published in the *International Journal of Project Management* (Lipke et al., 2009). The authors have drawn the attention to the crucial difference between a detailed schedule control and a general helicopter view on project management and control in their concluding remarks, and they expressed their preference for a less detailed control approach as follows:

> Some practitioners of Earned Value Management (i.e., project control) hold a belief that project duration forecasting can be made only through the analysis of the network schedule. They maintain the understanding and analysis of task precedence and float within the schedule cannot be accounted for by an indicator. Detailed schedule analysis is a burdensome activity and if performed often can have disrupting effects on the project team.

Some authors disagree and criticise this general approach as overlooking the most crucial details. They argue that the EVM performance measures are true indicators for project performance as long as they are used on the activity level (i.e., at the detailed schedule level) and not on the higher levels of the so-called Work Breakdown Structure (as suggested in the previous quote). For example, Jacob and Kane (2004) illustrate this statement using a simple example with two activities, leading to wrong and misleading results. They show that a delay in a non-critical activity might give a warning signal that the project is in danger, while there is no problem at all since the activity only consumes part of its slack. If the performance measures are calculated on the project level (instead of on the level of this activity), this will lead to a false warning signal and hence possibly wrong corrective actions.

This example shows a true danger of ignoring details, but I nevertheless tend to agree with the former approach that a detailed schedule control (i.e., monitoring every single activity of the project) might be a too burdensome task for a project manager. I must confess that I might have sowed confusion in my first four books (cf. Chap. 2), as all the example calculations on the EVM metrics were done on the detailed activity level. However, these books are written for researchers, and I chose for details only to illustrate how the formulas and calculations should be applied in an artificial setting. In my fifth book "The data-driven project manager: A statistical battle against project obstacles", I wanted to reach a wider audience (researchers *and* professionals), which is why I referred to the so-called *work package control* as a less-detailed alternative of controlling projects by measuring the performance of a set of activities, rather than for each activity individually. In short, I believe that details matter the most for the static scheduling and risk analysis phases but are far less important or desirable for the dynamic control phase.

The discussion and controversy on the right level of detail has become a central theme throughout this book, without often explicitly mentioning it. Since this book discusses the use of data-driven project management from both a research and a practical perspective, choosing the right level of detail will differ along the different chapters in the book. Part II discusses the three important missions of academic research and relates these missions to my own research on data-driven project management. It shows that academics should satisfy three specific needs (each chapter discusses one mission) to serve different audiences, ranging from colleague researchers who are really interested in the details of academic research to professionals who care less about details but want to translate these research findings into easy guidelines and best-practices. Hence, the level of detail will vary along the different chapters of Part II of this book. Part III shows that professional project managers have different ambitions than academics, with different and sometimes conflicting needs on how to use the data for managing projects. It will be shown that both academics and professionals share the common goal to improve the decision making process for managing projects, but they do it under a different level of detail. While academics care about every little detail in carrying out their research, professionals only want to use these academic methods when they can be adapted into easy guidelines and strong lessons-to-learn for professional use. Hence, a significant part of academic research will never reach the business world and is intended to increase the knowledge as a goal on itself. However, another part of the academic methodologies can – under the right circumstances – be applied in business on the condition that it keeps its best performing elements and gets rid of the unnecessary details. Part III discusses some simplifications of existing research and introduces the so-called *control efficiency* concept as a professional way to select the right level of detail in project management and control. Finally, a book about data-driven project management would not be complete without a detailed (*no pun intended*) discussion of *project data*. In Part IV of this book, it will be shown that choosing the right level of detail in generating or collecting project data is key for academic research with practical relevance. It will be shown that both artificial data (*researchers*) and empirical data (*professionals*) can be used to improve the understanding in managing projects, and a so-called *calibration procedure* is presented to transform one into the other, aiming at narrowing the bridge between academia and the professional business. Part IV consists of six chapters which I consider as the biggest contribution of this book, not only because the OR&S team has spent most of its precious research time on this important research area but also primarily because I truly believe that using the project data in a correct way is key for improving the decision making process in project management. Remember that *"without data you are just another person with an opinion"*.

The next section briefly reviews the academic literature of the three components of data-driven project management as the foundation of the upcoming chapters of this book. If you have already read some of my previous books or if you are a researcher familiar with the general theme of this book, I suggest to jump immediately to the next part of this book. If you, however, are interested in some specific details of the *scheduling*, *risk*, and *control* components, keep reading this chapter.

3.4 Literature (*in a Nutshell*)

I know that every (good) book written by an academic should start with a summary of the state-of-the-art literature on the topic under study, but this time, I will keep it very short. It is not that the current work in the literature is not good enough, but it would simply lead me too far from the book's central theme. An impressive amount of work has been published in the last decades on data-driven project management, especially in the field of project baseline scheduling, and much of the work I present in this book finds its foundation in the work of others. Therefore, this section will only provide a short overview of the three components of data-driven project management with a basic summary on how the current academic knowledge has progressed from the early endeavours to its current impressive status. Without having the intention to give a full literature overview, some interesting references to research studies will be given that can be used as a base for a further search to other sources. The three components of the data-driven project management framework are shown in Fig. 3.3 and will be explained along the following paragraphs. It displays the necessary steps that must be followed for the construction of a baseline schedule, the analysis of its risk, and for project control. The figure clearly illustrates the importance of the *baseline schedule* as a point of reference for the *risk analysis* and *project control* components, which is the central idea behind the *dynamic scheduling* philosophy discussed earlier in this chapter. A good understanding of the four phases of each of the three components is crucial in the upcoming chapters of this book and will therefore be briefly discussed along the following lines.

Schedule
The research on project scheduling as a subfield of project management finds its roots in the field of *Operations Research* and mathematically determines the start

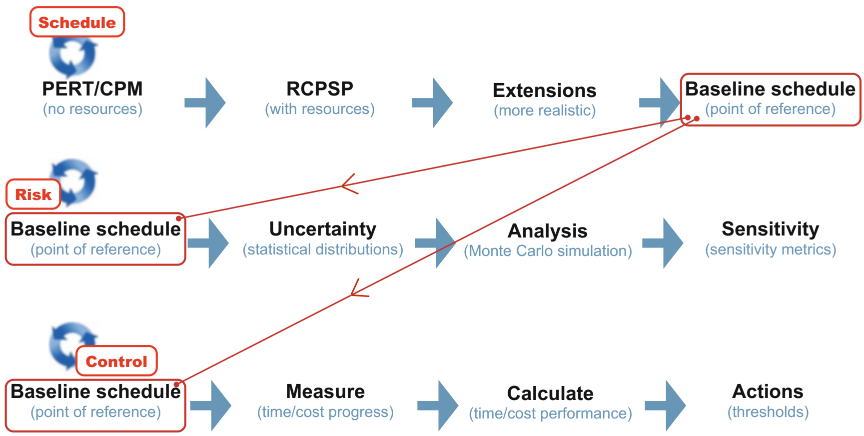


Fig. 3.3 The phases of the three components of data-driven project management

and finish times of project activities subject to precedence and resource constraints while optimising a certain project scheduling objective. In a research handbook written by the advisor (Erik Demeulemeester) and co-advisor (Willy Herroelen) of my PhD (Demeulemeester and Herroelen, 2002), the construction of a baseline schedule of a project is defined as follows:

> Project scheduling involves the construction of a project base plan which specifies for each activity the precedence and resource feasible start and completion dates, the amounts of the various resource types that will be needed during each period, and as a result the budget.

The literature on project scheduling is rich and diverse, and an overwhelming amount of research papers have been written on this challenging topic which makes it impossible to give a full overview without missing a few important ones. The initial research done in the late 50s mainly focused on network-based techniques, such as the *Critical Path Method* (CPM) and the *Program Evaluation and Review Technique* (PERT), which are still widely recognised as important project scheduling methods. Thanks to the development of the personal computer, project scheduling algorithms started to shift from solely *activity scheduling* to *resource allocation models* and an increasing number of software vendors have incorporated some of these resource allocation/scheduling models in their systems. The well-known *resource-constrained project scheduling problem* (RCPSP) has become the standard project scheduling problem in the academic literature which has resulted, due to its challenging nature, in numerous research studies and journal papers. These studies present a variety of methods to construct a resource-feasible schedule, including simple and fast priority rules, fast and efficient meta-heuristic solution approaches, and challenging exact algorithms. The list of publications in this challenging project scheduling field has become so overwhelmingly long that I have chosen to provide a short list of summary papers as well as some recent work done by my OR&S group in the next paragraphs, hoping that it can be used as a guide to explore new research directions.

- **Priority rules**: The research on *priority rules* to solve the RCPSP has started several decades ago, and excellent summaries are provided by Kolisch (1996a,b) and later updated and extended by Hartmann and Kolisch (2000) and Kolisch and Hartmann (2006). Despite their simplicity and low quality, priority rules are still used in recent studies to further improve the construction of a baseline schedule for big projects. As a matter of fact, two of my team members started to work on the selection and/or design of priority rules for solving the RCPSP for huge projects (up to thousands of activities). More precisely, Guo et al. (2021) have designed a system to automatically detect the best performing existing priority rule in the literature, while Luo et al. (2022) proposed a genetic programming approach to design new and better priority rules for this challenging project scheduling problem with resources. While priority rules will never be able to compete with the more advanced methodologies, they will remain interesting for big projects due to their ability to generate a resource-feasible schedule in no time.

- **Exact algorithms**: With the increasing power of computers and availability of efficient software tools, researchers began to develop exact algorithms to solve the RCPSP, which mainly consist of *mixed integer programming* (MIP) formulations and *branch-and-bound* (BnB) procedures. These algorithms contain advanced features based on a mathematical formulation of the problem and often are too hard to implement in a commercial software system. However, a real researcher prefers the difficult algorithms over the easy priority rules, not only for the challenge and inherent complexity but also mainly because they are able to provide *optimal* solutions. Such solutions guarantee to have the best possible objective (e.g., the lowest duration), but they can only be obtained for relatively small projects after a long search, making these algorithms not easily usable in practice. An excellent overview of most of the MIP formulations for the RCPSP can be found in Artigues et al. (2015), while a summary of the most-widely used BnB procedures for this problem is given by Coelho and Vanhoucke (2018). In Chaps. 6 and 11, various branch-and-bound procedures from the literature will be used for *machine learning* and *data generation*, respectively.
- **Meta-heuristics**: I might be wrong, but I have the impression that working on exact algorithms is out of fashion in academia. When I was working on branch-and-bound procedures for solving the RCPSP with the *net present value* (*npv*) objective, one of the major challenges for coding these advanced algorithms consisted of making the best use of the limited resources of a personal computer (slow CPU time, no hard disk, limited memory). Almost 30 years later, my PhD students no longer care about limited computer memory and do not know what it means to use hash tables and bit programming to spare computer memory. Thanks to this increased availability of fast computers, an impressive number of new solution procedures have been developed that can be classified under the umbrella of meta-heuristics. These algorithms aim at constructing solutions for the RCPSP to near-optimality by searching the search space in a very intense way and providing a whole range of different solutions hoping that the best one (or close to the best) is found. Most of these algorithms found their origin in nature-based processes (imitating cell division in the human body, the gravity of planets or the behaviour of ants, to name a few), and Pellerin et al. (2020) have given an impressive summary of the latest state-of-the-art results of most meta-heuristic solution approaches developed for the challenging scheduling problem.
- **Extended formulations**: The basic RCPSP is formulated under strict assumptions for the activity network and resource use and aims at minimising the total project duration. However, thanks to its practical relevance, many extensions have been formulated in the literature, including other objectives (such as the previously mentioned *npv*), extensions of activity and resource assumptions, and much more. The paper by Hartmann and Briskorn (2021) reviews several interesting extensions of the basic formulation of the RCPSP and illustrates that the research on the RCPSP is certainly not at a dead end. Their paper is an updated version of an older paper by the same authors written a decade earlier (Hartmann and Briskorn, 2010) and presents an overview of variants and practical extensions for resource-constrained project scheduling. Someone once

asked me if the research into this problem has not gradually solved everything. If you read these overview papers, then you know that this is certainly not the case. Researchers can certainly work on this problem for a few more decades, and I hope to see many new algorithms in the coming years that tackle this ancient problem from a completely new perspective.

The focus of this book does not lie on the use of advanced algorithms to construct the baseline schedule of a project under limited resources. As a matter of fact, in most chapters, it will be assumed that the availability of the resources is unrestricted, in which case the *critical path schedule* will be used as the baseline schedule for the project.[3] Consequently, in this book, it is more important to understand that the project baseline schedule must be used as the point of reference for the two other components of the data-driven project management methodology (risk and control, cf. Fig. 3.3), and the algorithm used to construct such schedule plays a secondary role. A brief summary of the literature for these two other components is given next.

Risk

While a lot of articles have been written on the importance of risk management for managing projects, much less attention has been spent on the so-called *schedule risk analysis* (SRA) methodology that focuses on the analysis of variability in the activities of the baseline schedule. I heard about the *schedule risk analysis* concept for the very first time in the non-academic paper of Hulett (1996), and it immediately got my attention due to its link with the project schedule. A deeper dive into the literature brought me to the paper of Williams (1995) who presented a classified bibliography of research related to project risk management and a follow-up study in Williams (1999) in which it is argued that the use of simulations in project management should be treated with care in order not to lose realism and reduce the concept to a theoretical exercise. Since an SRA study makes use of Monte Carlo simulation runs on the project baseline schedule, these papers have become the foundation of my own research in the analysis of the project schedule risk. Despite the fact that the SRA technique is not very complex, it took a while before I fully understood the relevance of this method. I started using the technique in 2006 during the aforementioned collaboration with Stephan, but only had a first publication in 2010 where I really understood why the technique was so important to the project plan. Maybe I am just a little slow.

The idea is simple though. A baseline schedule is nothing but a deterministic prediction of possible start and finish times for the activities without taking any possible variability into account. Since it only acts as a point of reference rather than a realistic prediction of the future project progress, it is known that the uncertainty during project progress will cause schedule disruptions which will lead to inevitable changes in the original project plan. A schedule risk analysis tries to predict the

[3] Only occasionally, the critical-path schedule (no resources) will be replaced by the resource-limited schedule of the RCPSP. In Chap. 11, some of the advanced BnB algorithms will be used to generate artificial project data.

impact of these disruptions in advance, prior to the project start and before the actual problems occur. Consequently, a schedule risk analysis puts the purpose of the baseline planning into perspective by adding uncertainty to the project activities. This uncertainty is expressed by probability distributions to model the variability in duration (or cost) and the model then analyses the potential impact of this variability on the final project objectives. This impact is measured by Monte Carlo simulations to imitate literally thousands of (artificial) dynamic project executions, and the output of such a simulation consists of so-called *sensitivity metrics* that analyse the relation between the variability in the activities of the project and the overall project objective. The four steps of an SRA (the second row of Fig. 3.3) are briefly summarised along the following lines (more details can be found in Vanhoucke (2015)):

- **Step 1. Schedule**: The construction of the project baseline schedule has been discussed in the previous paragraphs and consists of a table with start and finishing times for each activity of the project. It provides deterministic data based on single-point estimates about the duration and cost of each activity and completely ignores variability in the activities. This schedule plays a central role and acts as a point of reference for all calculations and simulations of the SRA method.
- **Step 2. Uncertainty**: While the time and cost estimates for the baseline schedule assume deterministic values, real project progress, however, is flavoured with uncertainty, leading to unexpected changes and problematic time and cost overruns. This behaviour must be mimicked in a Monte Carlo simulation by defining the uncertainty using probability distributions for the time/cost estimates used in Step 1.
- **Step 3. Analysis**: During the Monte Carlo simulation runs, values are generated from the distributions of the previous step to model the variability in the original time and cost estimates of the baseline schedule. In each simulation run, all activities of the project get a new value for their duration (and cost), which changes the total project duration and the critical path. After thousands of simulation runs, the simulation engine has generated a huge number of different project runs, and all the generated data is stored to calculate the activity sensitivity metrics in the next step.
- **Step 4. Sensitivity**: The data captured during the simulation runs are now ready to be processed, and sensitivity metrics for the time and cost behaviour of individual activities can be calculated. These metrics show the relative importance of an activity in the project network and measure the possible impact of activity variability on the project outcome.

The sensitivity metrics of the fourth step measure how sensitive the activities are to deviations from the schedule's original value, and these can be measured as deviations in time and cost estimates, as well as in the use of the project resources. In the next paragraphs, these three types of sensitivity metrics will be briefly discussed (for time, cost, and resources), and some of them will play a central role in the upcoming chapters of this book.

- The *time sensitivity metrics* are used to measure the sensitivity of changes in the activity durations on the total project duration. They refine the black-and-white view of the critical path (which stipulates that an activity is either critical or not) to a *degree* of criticality expressed as a percentage between 0% and 100%. The four most well-known time sensitivity metrics for measuring the time sensitivity of project activities are as follows:

  – Criticality Index (CI): Measures the *probability* that an activity lies on the critical path.
  – Significance Index (SI): Measures the relative importance of an activity by predicting the *impact* of the changes in the activity durations on the total project duration.
  – Schedule Sensitivity Index (SSI): Measures the relative importance of an activity taking the CI into account. This sensitivity metric measures both the *probability of criticality* and the *impact of activity duration variability* and is therefore a more complete metric than the two previous metrics.
  – Cruciality Index (CRI): Measures the correlation between the activity duration and the total project duration. These correlations can be measured in three different ways, using *Pearson's product-moment correlation coefficient* (CRI(r)), *Spearman's rank correlation coefficient* (CRI(ρ)), or *Kendall's tau rank correlation coefficient* (CRI(τ)).

  These time sensitivity metrics are obtained by applying the Monte Carlo simulation runs on *critical path* based schedules without taking the limited availability of resources into account. However, recently, Song et al. (2022) have extended these sensitivity metrics for project schedules with limited resources and have redefined the three metrics to RCI, RSI, and RSSI, with R used to refer to the presence of scarce resources in the project baseline schedule (which replaces the critical path based schedule by a feasible resource-constrained project schedule). These metrics have the same interpretation as the critical-path based metrics but will not be used in the current book.
- In many practical settings, the uncertainty in activity durations also has an influence on the variable cost of the activity, and therefore, the *cost sensitivity metrics* are used to measure the sensitivity of changes in the activity costs on the total project cost. While the time sensitivity metrics make use of the critical path schedule, the total cost of a project does not depend on the way the activities are scheduled but simply is equal to the sum of the costs of the activities. Therefore, three of the time sensitivity metrics (CI, SI, and SSI) that make use of the critical path data cannot be used for cost sensitivity. The Cruciality Index, however, does not make use of the critical path in its calculations and can easily be adapted to cost sensitivity by measuring the correlation between the cost variability in the activities and the variability in the total project cost. Consequently, the three time-based versions of the Cruciality Index, CRI(r), CRI(ρ), and CRI(τ) are also used for cost sensitivity using the generated cost data obtained from the various runs in the simulation.

- The *resource sensitivity metrics* are an extension of the cost sensitivity metrics but are used to measure the sensitivity of cost variability in the resources used by the activities (rather than treating the activity cost as a general cost). Activities require resources during the project progress and the total activity cost can consist of various components, including the fixed or variable costs for the renewable resources connected to these activities. Rather than measuring the general cost sensitivity of an activity, it is often interesting to see how sensitive each resource is with respect to the total project cost. Similar to the general activity cost sensitivity metrics, the resource cost sensitivity can be measured by the three versions of the Cruciality Index, CRI(r), CRI(ρ), and CRI(τ), but they will now be calculated for each type of renewable resource rather than for each project activity.

It should be noted that the use and importance of the schedule risk analysis methodology perfectly fits within the previous discussion of defining an appropriate *level of detail* when managing projects (Sect. 3.3). Since the sensitivity metrics are defined within the range between 0% (low sensitivity) and 100% (high sensitivity), the purpose of SRA is to determine which parts of the projects are important and which are not. Since the activities with low sensitivity values are far less important than the ones with high values for the metrics, SRA is a method for trying to get rid of less-important details and for paying attention to what really matters for the project. When unexpected disruptions occur during the project's progress, the high-sensitive activities require more attention since their variability is more likely to have a bigger impact on the project objective. From this perspective, SRA is a methodology to *improve the project manager's focus* on the most essential parts of the project, without losing too much time on the activities that do not matter much. It took me a while, but I finally understood that the real purpose of a static *schedule risk analysis* is to improve the dynamic project control phase of the project, as I wrote in my first SRA study (Vanhoucke, 2010) the following words:

> The interest in activity sensitivity from both the academics and the practitioners lies in the need to focus a project manager's attention on those activities that influence the performance of the project. When management has a certain feeling of the relative sensitivity of the various parts (activities) on the project objective, a better management's focus and a more accurate response during project tracking should positively contribute to the overall performance of the project.

In this book, three *time sensitivity metrics* (CI, SI, and SSI) will be used in various chapters as a way to control a project with a better focus, and it will be referred to as a *bottom-up control method* to express that the project manager wishes to monitor the performance of a project in progress with the least possible effort (*no details, just focus!*) and the maximal possible impact (*taking actions when necessary!*). This *project control* phase constitutes the third and last component of data-driven project management and will be discussed next.

Control

Monitoring and controlling the progress of a project is key to the success of the project, and constitutes the third and most important component of data-driven project management. *Project control* is the periodic and repetitive act of collecting key data about the project progress that must be transformed into easy and understandable performance metrics and forecasts to enable the project manager to understand the current status of the project, and to invoke corrective actions when the project tends to run into the danger zone. The most well-known methodology used to monitor the dynamic progress of a project is known as *Earned Value Management* (EVM). The EVM methodology has been used since the 1960s, when the USA Department of Defense proposed a standard method to measure a project's performance. It is a generally accepted methodology used to measure and communicate the real physical progress of a project and to integrate the three critical elements of project management (scope, time, and cost management). It takes into account the work completed, the time taken, and the costs incurred to complete the project and it helps to evaluate and control project risk by measuring project progress in monetary terms. The four steps of Fig. 3.3 (third row) will be briefly outlined along the following lines, and further details about the EVM metrics will be given in Chap. 4.

- **Step 1. Schedule**: The project baseline schedule is (again) a necessary point of reference for measuring the progress of the project. The baseline schedule is no longer visualised as a well-known Gantt chart but expressed as the cumulative increase in the costs of the activities along their planned start times, and this cost curve is known as the *planned value* curve. This curve is the first of three key metrics that serve as periodic inputs for measuring the performance of the project.
- **Step 2. Measure**: When the project is in progress, the project manager has to periodically measure its (time and cost) progress using two additional key metrics, known as *actual costs* and *earned value*. These two additional values must be calculated at each review period and will be used – together with the *planned value* of that period – to calculate the time and cost performance of the project at that time.
- **Step 3. Calculate**: The EVM system automatically calculates *time and cost performance indicators* that express how much the project is delayed or ahead of schedule relative to the baseline schedule (or over or under budget relative to the planned project budget). These performance indicators give a general view on the health of the project at the current moment in time and act as warning signals for actions in case the progress is no longer satisfactory.
- **Step 4. Actions**: The actions of the project manager taken after the warning signals reported a problem should enable the manager to solve the project problems and should (ideally) bring the project progress back on track.

When I came into contact with the EVM methodology as a general and practical project management system at the beginning of my academic career, I was surprised how few scientists were interested in this widely used system. As a matter of fact, although the system was widely accepted by many professional project managers as a good control system, not many research papers were written to validate, critique, improve, or even replace this method with other, possibly better control methods. Despite this lack of interest from the academic world, many good books and white papers (mostly not peer reviewed) were written, and I have to mention the books by Quentin W. Fleming and Joel M. Koppelman who took the mystery out of the EVM methodology and made it easy to understand for everyone. In their book (Fleming and Koppelman, 2010), they reduce an EVM system to the essence as follows:

> Earned Value Management is an invaluable tool in the management of any project. In its simplest form, Earned Value Management represents nothing more than fundamental project management - managing a project with a resource-loaded schedule.

Furthermore, these authors also recognised the importance of the baseline schedule as *the* point of reference for project control, as they stated the following in their books:

> In its most fundamental form, employing Earned Value Management requires nothing more than managing a project with a good schedule, a schedule with the authorised budget embedded task by task.

Despite the lack of academic interest more than 20 years ago, today, a growing number of researchers rely on the EVM methodology to investigate and improve the methods to control projects, which has led to a growing number of academic *peer-reviewed* articles. I started working on this fascinating topic thanks to my collaboration with friend Stephan Vandevoorde around 2004, as I discussed in the previous chapters. After two peer-reviewed articles in academic journals co-authored with Stephan, I decided to continue working on this topic, and it gradually became one of the main research themes at my growing OR&S group. We extended the EVM methodology in our project control studies to *analytical* and *statistical* project control methods (Chaps. 5 and 8), we integrated it in a *machine learning* framework (Chap. 6), and we even decided to *collect empirical data* with real progress reports to increase the realism of our research studies (Chap. 13). This challenging process of continuously improving and extending the EVM methodology to better alternative control methods is an important part of this book and is still ongoing. For an overview of the work done in the scientific literature up to 2015, a reference is made to our article published in the *International Journal of Project Management* (Willems and Vanhoucke, 2015). This article is now old and out of date, so keep reading this book for a more detailed look at the various research efforts that have been made to date.

References

Artigues, C., Koné, O., Lopez, P., & Mongeau, M. (2015). Mixed-integer linear programming formulations. In C. Schwindt & J. Zimmermann (Eds.), *Handbook on Project Management and Scheduling* (pp. 17–41). Springer International Publishing AG.

Coelho, J., & Vanhoucke, M. (2018). An exact composite lower bound strategy for the resource-constrained project scheduling problem. *Computers and Operations Research, 93*, 135–150.

Demeulemeester, E., & Herroelen, W. (2002). *Project scheduling: A research handbook*. Kluwer Academic Publishers.

Fleming, Q., & Koppelman, J. (2010). *Earned value project management* (4th edn.). Project Management Institute, Newton Square, Pennsylvania.

Guo, W., Vanhoucke, M., Coelho, J., & Luo, J. (2021). Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem. *Expert Systems with Applications, 167*, 114116.

Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 207*, 1–15.

Hartmann, S., & Briskorn, D. (2021). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research, 297*(1), 1–14.

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 127*, 394–407.

Hulett, D. (1996). Schedule risk analysis simplified. *Project Management Network, 10*, 23–30.

Jacob, D., & Kane, M. (2004). Forecasting schedule completion using earned value metrics? Revisited. *The Measurable News, Summer: 1*, 11–17.

Kolisch, R. (1996a). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management, 14*, 179–192.

Kolisch, R. (1996b). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research, 90*, 320–333.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research, 174*, 23–37.

Lipke, W., Zwikael, O., Henderson, K., & Anbari, F. (2009). Prediction of project outcome: The application of statistical methods to earned value management and earned schedule performance indexes. *International Journal of Project Management, 27*, 400–407.

Luo, J., Vanhoucke, M., Coelho, J., & Guo, W. (2022). An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem. *Expert Systems with Applications, 198*, 116753.

Pellerin, R., Perrier, N., & Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 280*(2), 395–416.

Servranckx, T., & Vanhoucke, M. (2021). Essential skills for data-driven project management: A classroom teaching experiment. *Journal of Modern Project Management, 8*(4), 123–139.

Song, J., Martens, A., & Vanhoucke, M. (2022). Using earned value management and schedule risk analysis with resource constraints for project control. *European Journal of Operational Research, 297*, 451–466.

Uyttewaal, E. (2005). *Dynamic Scheduling With Microsoft Office Project 2003: The book by and for professionals*. Co-published with International Institute for Learning, Inc.

Vanhoucke, M. (2010). Using activity sensitivity and network topology information to monitor project time performance. *Omega The International Journal of Management Science, 38*, 359–370.

Vanhoucke, M. (2015). On the use of schedule risk analysis for project management. *Journal of Modern Project Management, 2*(3), 108–117.

Willems, L., & Vanhoucke, M. (2015). Classification of articles and journals on project control and earned value management. *International Journal of Project Management, 33*, 1610–1634.

Williams, T. (1995). A classified bibliography of recent research relating to project risk management. *European Journal of Operational Research, 85*, 18–38.

Williams, T. (1999). Towards realism in network simulation. *Omega The International Journal of Management Science, 27*, 305–314.

Part II
What Academics Do

Academic freedom is a contested issue
and, therefore, has limitations in practice.

In the next two parts of this book, my *Operations Research & Scheduling* (OR&S) group's dynamic scheduling research will be discussed to illustrate the similarities *and* differences between academic efforts and professional needs about better using data-driven methodologies in project management. Making a distinction between academic research and professional needs might be a bit artificial, but most of what I have written in Part II (this part, *"what academics do"*) and Part III (the next part, *"what professionals want"*) comes from the distinction between academia and business discussed in my keynote presentation at the *Creative Construction Conference* in Budapest (Hungary) in 2016 entitled *The data-driven project manager: Academics like what they do, and professionals know what they want*.

Before I discuss the needs of professionals in the next part of this book, I will start this Part II from my own perspective as an academic researcher trying to explain what academics do with their precious time at the universities. Academic researchers who are active in the field of Project Management (PM) belong to a very strange species. They work on a research topic that is highly relevant to the professional manager and business practice, but they often have no primary interest in reality. As you will see, many research papers are filled with algorithms and data analyses written with the primary goal of getting a publication, or pushing the boundaries of knowledge, but *not* with the hope that managers will apply them in their daily practice. Whatever most researchers claim when they talk about their research output, what is going on in reality is almost always used as a secondary inspiration, and it is much more fun to investigate a problem from a theoretical perspective without caring too much about practicalities in business. The main purpose of Part II is to describe the real goals of academic research and discuss the ambitions academic researchers pursue in their search towards creating new knowledge. I hope this part will convince the readers that academic research plays a major role in improving the decision making process of managing projects in business, even if many research studies primarily focus on theoretical problems, advanced algorithms, and detailed data analyses. From this perspective, it will be shown that academic research not only has to provide a better understanding in today's professional project management methodologies, but also has to improve these methodologies (for a better professional use) or even radically change them into new methods (even if that means that professionals will not adopt them at all). This triple ambition is described in this book part as three separate *missions of academic research* as briefly outlined in the next section.

Three Missions

Academic research is the careful study of a given problem or subject in a certain discipline, undertaken to discover facts or new principles. It should be carried out in a detailed and accurate manner, under strict and well-defined circumstances following the currently existing rules of science, and it should be directed towards increased knowledge and/or solutions for the problems under study. Since human beings like to categorise things, a distinction has been made in research too, classifying research projects as either *fundamental* research (at universities) or *applied* research (at business schools). The distinction between these two classes of research is of course a bit artificial, and a research project can often not easily be classified as either the one or the other. While both types of research aim at extending the current body of scientific knowledge, it is said that applied research aims at discovering a solution for a relevant practical problem (often in collaboration with a client/company), whereas basic research is directed towards finding solutions with a broad base of applications (and therefore not designed for a specific company). At my OR&S group, the distinction between these two types of research is not very important, but we classify our projects as *fundamental* research when it is funded by university money with no one else involved except my team members and a few very close friends with professional experience. However, when the research projects are funded by external resources, or when the research themes are inspired by a significant group of professional project managers who want to see their practical problems solved, the project is classified as *applied* research. In this case, we no longer carry out the research in isolation, but instead listen to wishes and desires from the outside world. I know it is an artificial distinction, but it works.

Ideally, any research study in the field of data-driven project management should have a strong applied component and should aim at increasing knowledge about managing *real* projects, but that does not exclude the use of theoretical concepts and does not mean that any research study should always immediately lead to practical results and business relevance. I have had hundreds of interesting discussions with colleagues from both universities and business schools about the real purpose of academic research, and I learned that every person has a slightly different definition. I noticed that everybody more or less agrees on the nature of research and how it has to contribute to a better knowledge of the topics under study, but when it comes to the practical relevance for the real world, I saw different people with very different opinions. Some believe that academic research should maximise the impact

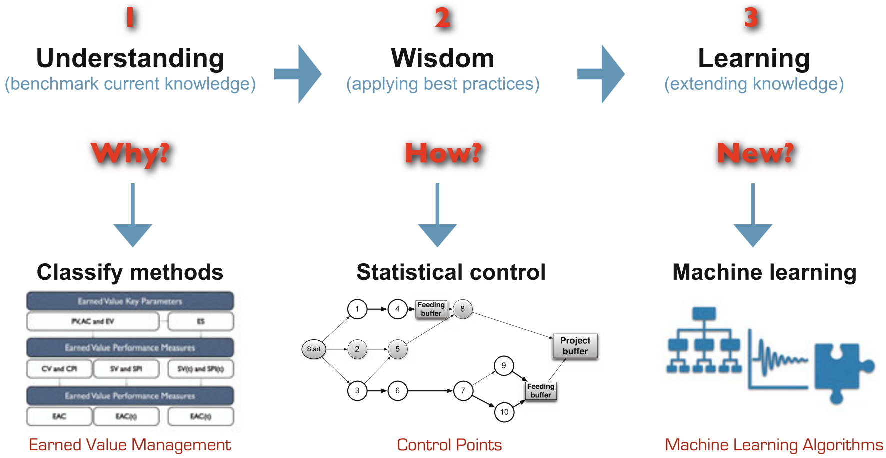


Fig. 1 Three academic missions (*understanding, wisdom, and learning*)

on managers and policy makers, while others believe academic curiosity is a goal on itself, regardless of the needs of society. In the next three chapters of this book, I will classify the purpose of academic research under three separate missions which I called *understanding*, *wisdom*, and *learning*. The titles of these three missions come from three different quotes from three different geniuses and are assembled in one quote that goes as follows:[4]

> Knowledge speaks,
> understanding explains,
> wisdom listens,
> and learning changes.

A visual summary of the three academic missions is given in Fig. 1 that will be outlined in detail in the three upcoming chapters. It will be shown that the OR&S group has worked on each of these three missions separately, each with a totally different research agenda and end goal in mind. In Chap. 4, a *classification* of existing project control methods will be provided in order to better understand why they work for some projects, and why they fail for others. Chapter 5 discusses how these existing methods can be extended to more advanced statistical methods to improve their performance and includes the extension of these project control methods to the so-called *control points*. Finally, Chap. 6 will develop totally new methodologies including *machine learning* algorithms to fundamentally change the

[4] The original quotes come from Albert Einstein, Jimmy Hendrix, and William Deming and will be used in the three chapters of Part II.

way projects could be managed in the future. It has been a fantastic journey spanning almost two decades with the highs (when a paper is accepted for publication) and lows (another reviewer with a request for changes) that typify the nature of research. Welcome to the world of academic data-driven project management research.

Chapter 4
Understanding

The first mission of academic research is to create insight into existing methodologies used by professional project managers. This Mission #1 should focus on comparing, validating, and assessing the performance of *existing* project management methodologies, rather than on developing *new* alternative methodologies to improve the management of projects. It should therefore aim at creating insights into why a given methodology works for some projects and why it sometimes fails so miserably for others, without making any changes or improvements. Translating the current knowledge into this "*why*" question will create a better *understanding* of the existing methodologies, and this is a fundamental and necessary step in the search for improvements (which will be discussed in the next chapter under the mission *wisdom*). It was Albert Einstein who highlighted the important difference between knowledge and understanding as follows:

> Any fool can know.
> The point is to understand.

I am not stating here that professionals have only knowledge about how their methods work but no understanding why they use them, but rather, that it should be one of the primary goals of academics to look at today's understanding with a critical eye. Professional project managers often use a specific methodology for managing their projects based on past experience and because *it simply works*. They have chosen a particular methodology for no good reason, just use it because others do. Apart from some fine-tuning towards the specific needs of their project, they have no real incentive to question its performance as long as it does the job accurately enough. They have, from their perspective, enough with what they have and feel no urge for creating a better understanding if these methodologies provide them with enough data to support their decisions. If it does the job, why should you criticise it?

That is not to say that professional project managers are blind to the weaknesses and flaws of the existing methodologies, but when they fail, they often criticise them as being irrelevant for *their* projects, without even thinking deeply what

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_4

the root causes are that lead to their disappointing performance. Academics think differently and want to find out what the reasons are for such failures. While these academic researchers have obviously less experience in the practical implementation of existing methodologies, the search to understanding why a given methodology works or fails lies in the nature of their job. I believe that this lack of practical experience is exactly the point where academics can help increasing a professional's understanding. A good and high-quality assessment of current methodologies should be done in a sound and rigorous way, far away from the practical problems and case-specific settings of the professional world. As a matter of fact, only academics have the luxury to stay away from the practical issues of the professional world, and with their experience in performing experiments under a controlled design, they can create a better understanding without being biased by numerous noisy inputs irrelevant for the research study. I therefore personally believe that the assessment of existing methodologies should be carried out with only a limited input from the professional world, leaving the academics in their ivory tower designing experiments that professionals would never have thought of. After my PhD graduation, I felt such an inexhaustible hunger to contribute to Mission #1 of academic research that I started a project duration forecasting study with only one professional project manager involved. The results of this study will be briefly discussed in the next section.

4.1 Measuring Time

The importance of Mission #1 (*creating understanding*) has become extremely apparent in 2004 when I started to work on the research for my first book *"Measuring Time"*. As I already briefly mentioned in Chap. 2, I almost accidentally entered the field of Earned Value Management (EVM) thanks to my close collaboration with Stephan Vandevoorde, a project manager with no research experience but with an endless interest in academia. He is one of those professionals who is never satisfied with the status quo, and he always has that critical look in his eyes when he talks about the way he manages his projects. Despite his rich experience with large and important projects, he always tried to critically scrutinise his own methodologies with which he was immensely familiar, and he did not hesitate to criticise or even change them altogether. With his academic mind and his professional experience, he was looking for someone who wanted to think along about the usefulness and importance of Earned Value Management. He could not have crossed my path at a better time, and we quickly became good friends for life after our first collaborations.

In our first joint project, we were trying to implement several EVM methodologies for controlling and predicting the timing of projects, but we were not sure which method suited best for our challenging task. There were three basic EVM methods to monitor the timing of a project in progress, and each of them predicted the expected final duration of the project using progress data in a slightly different way. Two of

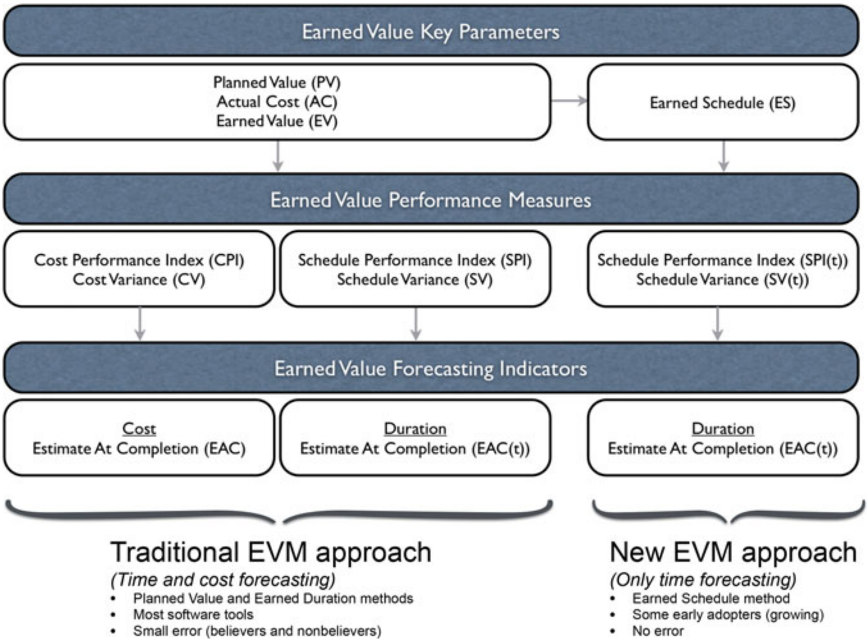


Fig. 4.1 Earned Value Management methodologies in 2003

these methods were well known in the professional control community as they both relied on the traditional three-level EVM system shown in Fig. 4.1, and we referred to them as the *planned value method* (Anbari, 2003) and the *earned duration method* (Jacob, 2003). The third method is slightly different from these traditional methods and was not well-known in the community since it relies on a new EVM key metric that no one understood at that time. This new method is referred to as the *earned schedule method* (Lipke, 2003). I could give here a full overview of the differences between these three methods by telling stories of how they have been used with variable success for different projects. These methods were criticised by some but praised by others, but since that has been the subject of most of my previous books, I will not repeat this story here in detail. Instead, I will just give a very succinct summary of the three levels of an Earned Value Management system as shown in Fig. 4.1 before continuing the story of my comparative study with Stephan.

- **Key metrics:** Any EVM system makes use of three key metrics to measure the performance of a project in progress, known as the Planned Value (PV), the Actual Cost (AC), and the Earned Value (EV). These key metrics are input values provided by the project manager at each review period at which the project progress is monitored. The *planned value* is the time-phased budget baseline and is an immediate result of the schedule constructed from the project network. Since the baseline schedule is constructed in the static phase, this key metric is completely determined at the start of the project (i.e., for each possible review period in the future, the planned values are known). The planned value is often called the *budgeted cost of work scheduled*, but I prefer to call it *planned value* because it sounds so much better. The two other key metrics are dynamic metrics, which means that their values are only known for review periods of the past. For future review periods, these values are unknown and become only visible when the project progresses. The *actual cost* is often referred to as the *actual cost of work performed* and is the cumulative actual cost spent at a given review period. The *earned value* represents the amount budgeted for performing the work that was accomplished by a given period. It is often called the *budgeted cost of work performed* and is equal to the total activity (or project) budget at completion (BAC, the total planned cost of the baseline schedule) multiplied by the percentage activity (or project) completion (PC) at the period ($= \text{PC} * \text{BAC}$).

- **Performance measures:** The project's time and cost performance at a certain point in time is determined by comparing the three key parameters PV, AC, and EV, resulting in four well-known performance measures. The *Schedule Variance* ($\text{SV} = \text{EV} - \text{PV}$) and the *Schedule Performance Index* ($\text{SPI} = \text{EV}/\text{PV}$) measure the time performance of a project. Similarly, the *Cost Variance* ($\text{CV} = \text{EV} - \text{AC}$) and the *Cost Performance Index* ($\text{CPI} = \text{EV}/\text{AC}$) measure the cost performance of a project in progress. Values for SPI and CPI lower than 100% indicate a lower-than-expected performance (i.e., a late project or over budget), while values higher than 100% indicate a better-than-expected performance (a project ahead of schedule or under budget). These performance metrics can be used as warning signals for the project manager to indicate that something needs to be done to resolve the issues in the project progress and will be the subject of several future chapters.
- **Forecasting indicators:** Predicting the final expected duration and cost of a project based on its current progress is done by forecasting indicators. The general formula for predicting a project's final cost is given by the *Estimate At Completion* indicator (EAC or sometimes abbreviated as EAC(€) to highlight the *cost* aspect of the prediction). The EAC predictor estimates the total cost of a project based on the current actual cost (AC) increased with an estimate of the expected cost necessary to finish the remaining work of the project. Similarly, the time prediction is given by the EAC(t) indicator (the *t* is added to EAC to clarify it is a time prediction) that predicts the total project duration based on the current actual time of the project (often abbreviated as AT) increased with an estimate about the expected time necessary to finish the remaining work. Stephan and I have shown that the expected time and cost of the remaining project work can be calculated under different scenarios, resulting in different alternative formulas for EAC(€) and EAC(t) that are not relevant to the current chapter. However, in Sect. 9.3 of Chap. 9 (*reference class forecasting*), I will go slightly deeper into these different scenarios and provide more details on the EVM forecasting formulas.

If you feel unconformable after seeing too many abbreviations used at each level of the EVM system, I advise the readers to stop here for a moment to consult a good EVM book (there are plenty out there) or read a few short articles at the PM Knowledge Center (www.pmknowledgecenter.com). You can also take a look at Appendix B that provides a glossary of the most well-known EVM concepts, with references to other interesting sources. The number of abbreviations can be indeed confusing at some times, certainly because not everybody uses the same terminology or symbols, leading to different formulas that basically measure the same thing. When I started working with Stephan, it had become almost impossible to find a common thread through the various articles on this interesting topic, let alone find any advice on which forecasting method would work best. It was time to bring some structure in the chaos.

Figure 4.1 provides a clear and simple three-level structure of EVM and the careful readers have noticed that the figure is split into two separate classes. The first class consists of the *traditional EVM approach* and includes the two well-known time forecasting methods (the *planned value method* and the *earned duration method*). This class is called *traditional* since these methods were very familiar to most professional project managers. The project performance is measured by the previously discussed cost performance index (CPI) and schedule performance index (SPI), and both indicators were implemented in most commercial software tools to monitor both the cost and time of projects in progress. The third method is classified as a *new EVM approach* and includes the third *earned schedule* (ES) method that was introduced in an article in 2003. This article proposed an alternative (and to some believed as a better) way of managing the time aspect of the projects. The ES method was totally unknown to most professional project managers and certainly not integrated in commercial software tools, and it was Stephan who drew my attention to this third method. The method had been proposed by Walt Lipke, a professional project manager at the Department of Defence (USA) who published his seminal paper *"Schedule is different"* in The Measurable News (Lipke, 2003). The article explains the possible error in the traditional SPI metric as it cannot predict the expected duration of a project in an accurate way. More precisely, he shows that the SPI always ends at 100% at the end of the project (denoting a project finishing on time), even when the project finishes dramatically late.[1] To solve this problem, he proposed a simply yet elegant alternative to the *earned value* (EV) key metric, which he called the *earned schedule* (ES) metric, by simply translating the monetary value of EV into a time-based ES value. Based on this new ES metric, he replaced the flawed SPI performance metric by the new SPI(t) metric which he defined as SPI(t) = ES/AT. The AT defines the actual time of the project at the current review period, which is equal to the number of days the project is in progress, and the ES is –as said – identical to the EV metric but expressed in time. Walt shows that his SPI(t) is therefore quite similar to the traditional SPI = EV/PV formula, but he argued in his article that this new performance measure is now reliable over the complete horizon of the project. As a matter of fact, when a project finishes

[1] This error occurs for all projects due to the fact that the earned value (EV) metric is always equal to PV at the end of the project, and therefore, SPI = EV/PV is always equal to 1.

late, the SPI(t) indicator will be lower than 100%, indicating the true status of the project. Some would argue that this new idea was not exactly a brilliant idea, but rather a logical consequence of a mistake in the previous formulas, but I think Walt did initiate something through his article that few others would have succeeded in doing.

Neither Stephan nor I had heard about Walt in 2004, but once we got the article, we immediately decided to include this third method in our comparison study, and it changed our life. When I first heard about the ES method, it was so similar to the original EV method that I thought including it in the comparative study would not yield impressive results. The basic idea of the new ES method was very simple, as it proposed an improved version of the classic SPI metric which, according to Walt's article, misrepresents a project's progress (he called this "*the quirky behaviour of the SPI*"). Indeed, if the SPI always ends at 100%, even if the project ends dramatically late, it will likely lead to erroneous conclusions during project progress and therefore will also result in low accurate time predictions. Instead, as the alternative performance measure SPI(t) measures the actual project progress (early, on time, or late) at each stage of the project, this should obviously lead to much more reliable predictions. Perhaps it would be a little too simple to include this in an academic article, but many thought differently.

Our article indeed showed on a small sample of three projects that the ES metric is a better time predictor than the traditional methods, but not everybody accepted this as easily as we did. I soon learned that for many professionals the *earned schedule method* was not just a simple extension of traditional EVM methods, but rather an attack to shut down the currently accepted EVM system. Indeed, you had two camps in the professional community of the newly suggested system, some were believers while others were opponents. Some project managers simply did not want to accept the findings of our article, arguing that this improved ES performance is just a coincidence (tested on only 3 projects) and cannot be generalised to other projects. Some even went so far as to call the new method *controversial* (no kidding!) that sparked tough and sometimes painful discussions between the traditional EVM and new ES believers. What struck me most was the (sometimes emotional) preference people had for the traditional EVM or the new ES methodology, without any rational arguments or solid data analysis. At a workshop where I had to give a keynote speech, the organisers even kindly asked me not to mention the new ES method and only talk about the EVM method so as not to hurt the audience too much. I completely ignored this request during my speech and was never invited to this workshop again. To make matters worse, the controversy was also present in academia. A few years after my ES study with Stephan, I submitted a paper to a scientific journal for which I received a review report recommending not to accept the manuscript. I normally never share review reports with others, but this one was so short and to the point that I can print it out in full in the following lines:

> I do not believe in Earned Schedule
> and therefore recommend to reject this paper.

What? I had no idea that scientific research was a matter of faith and belief. I have never submitted another article to this journal, and I never will, but fortunately many of my later articles on the EVM/ES topic have been published in a high quality journal, such that I can conclude that the controversy is no longer a problem. Nevertheless, it shows that creating an understanding (i.e., Mission #1) of why some methods work and why others fail is an important mission for academia. Finally, our first collaborative paper titled *"A Comparison of Different Methods for Predicting Project Duration Using Earned Value Metrics"* was published in the recognised *International Journal of Project Management* (Vandevoorde and Vanhoucke, 2006) and many other publications followed. As I mentioned in a previous chapter, this study has the highest number of citations of all the articles that I have written with my team, and I think this is mainly because the ES method was initially not widely accepted. Stephan and I met Walt several times later in our careers, and I feel privileged to say that we have become friends for life, not only sharing our passion for project management but also our love for music and art.[2]

4.2 Shedding New Light

After our first publication, Stephan and I were pleasantly surprised by the positive response and the growing number of citations, but we were not completely satisfied with the achieved results, especially since our conclusions were drawn on only three projects from Stephan's company. Rather than providing case-specific results, we wanted to draw general conclusions so that we could completely eliminate any last spark of doubt. To achieve this, we started a second comparison study, this time based not on real data from three projects, but on simulated data with thousands of artificial projects. Professional project managers do have real data, but they are often very case-specific and contain errors, and the conclusions drawn are often biased by their own experience and the way they collected the data. Academics have the freedom to generate artificial project data and rely on simulation studies to imitate any project progress under uncertainty, and this simulated reality can be created under a controlled design by including as many scenarios as possible.

For this study, we designed a totally new simulation model focusing on the performance of critical and non-critical activities of the projects, and we steered the simulation in such a way to invoke problematic behaviour in the predictions. More precisely, we tested the accuracy of the warning signals provided by the traditional SPI and new SPI(t) performance metrics by simulating true and false warning signals. We tested the impact of correct warning signals (i.e., the SPI and SPI(t) represent a *true* project progress status) and unreliable warning signals (i.e., the SPI

[2] The albums of the late singer J.J. Cale I have in my vinyl collection are recommended by Walt. Once he told me that J.J. Cale attended the same high school as Walt did, I bought some of these records. They are fantastic!

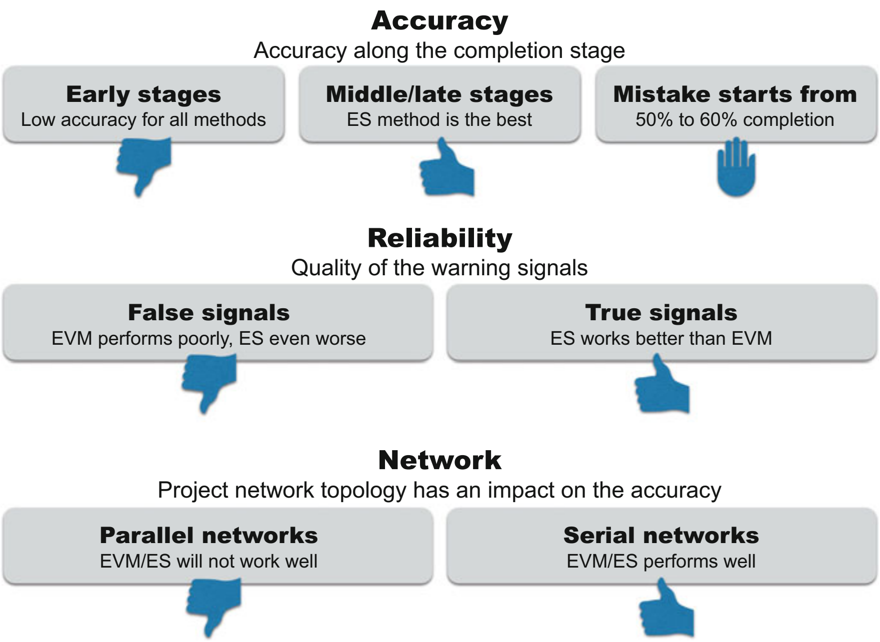


Fig. 4.2 Performance of EVM/ES methods for time forecasting

and SPI(t) report a *false* project status, e.g., a warning for a project delay, while the true status is that the project is ahead of schedule) by simulating projects that finish ahead of schedule and projects that finish with a delay. The so-called *scenario model* that we developed for this study is explained in detail in Sect. 12.4 of Chap. 12 of this book, and the artificial project database, consisting of more than 4000 projects, is presented in Chap. 11. This study is published in the peer-reviewed *"The Journal of Operational Research Society"* (Vanhoucke and Vandevoorde, 2007) and the main results are summarised in Fig. 4.2. The next paragraphs provide a short summary of the achieved results, classified along three criteria (accuracy, reliability, and project network structure).

Accuracy When making predictions, accuracy matters. Our simulation model measured the accuracy of the time predictions by periodically comparing the predicted duration with the final project durations (which is only known when the project simulation is finished) and then calculating the average absolute or relative deviations. These accuracy measures are known as the *mean absolution percentage error* (absolute) or *mean percentage error* (relative) and are used to validate the quality of each of the three time prediction methods. The results of the simulation experiments revealed that the accuracy of the three methods differs along the completion stage of the project. First and foremost, the results showed that the accuracy along the *early stages* of the project was low for all methods and no method was able to significantly outperform the other. Obviously, when the project is in the early stages, not a lot of project progress data are available yet, and none of the methods can rely on enough data to accurately predict the final project duration. However, the results also indicated that for the *middle stages* and certainly for the *late stages*, the new ES method outperforms the two other traditional EVM methods. As a matter of fact, these results confirmed the observations of Walt's paper and the conclusions drawn in our first study based on the three projects, but this time, it was tested on a huge set with thousands of projects. Of course, we were not very surprised by these results, since it was known that the SPI – used by the traditional EVM methods – is seriously flawed, while the SPI(t) – used by the ES method – does not suffer from that problem and so the ES method should logically lead to a better forecasting quality. However, less straightforward was the observation that the mistake of the SPI already starts to have a negative impact on the accuracy of the predictions from 50% to 60% project completion. This means that the predictions are not accurate during the most crucial phases of the project progress and might lead to wrong decisions and a waste of time and money for the project manager and the stakeholders relatively early in the project. Like it or not, the simulation study showed once and for all that the ES method is a clear winner when it comes to predicting the timing of a project and is therefore preferable to the traditional EVM methods.

Reliability Despite the clear advantage of using the ES method over the EVM method to predict the expected duration of the project, we also wanted to find out whether these predictions depended on the quality of the input data. More precisely, the predictions are made based on the input data (key metrics) and generated warning signals (performance metrics), and sometimes these metrics are not fully reliable. Consider, for example, a project with a lot of critical activities running on time, and only a few non-critical activities (i.e., activities with slack) with some minor delays. Any EVM/ES system will detect these activity delays and will report an SPI and/or SPI(t) value lower than the expected 100% performance, indicating that the project is suffering from a delay. This might look like a correct (true) warning signal (since some activities are indeed delayed), but since the delays are only for the non-critical activities, the delays could still be within their slack so that they will not have any impact on the total project duration. Consequently, the corresponding warning signals might be considered as *unreliable* since they measure a project delay while the project is perfectly on time. A similar but opposite example can be given for projects for which the SPI and SPI(t) metrics report a project ahead of schedule, while the true status is that the project will be delayed. Just think of many non-critical activities that are ahead of schedule and only one critical activity with a small delay, and you will probably predict an ahead-of-schedule project status while its true status will be delayed. Again, the warning signals might not be very reliable, and predictions might therefore be totally inaccurate.

The results of these reliability experiments have shown that the ES method does indeed perform less well compared to the traditional methods when the warning signals are unreliable (i.e., false). Some people might conclude that this is an advantage of the traditional methods, but I believe the opposite is true. Since the results indicate that the ES method performs best when the warning signals are correct (*true* signals) but performs very poorly when the warning signals are wrong (*false* signals), this is a clear indication of the quality of the ES method. However, the traditional methods certainly do not perform as well as ES when the warning signals are correct but perform relatively well (and better than the ES method) when the warning signs are completely wrong. As a result, the traditional methods seem to perform almost randomly, regardless of the reliability of the warning signals, which is certainly not a sign of high quality. The ES method, on the other hand, is a clear winner provided the warning signals are reliable, and this *garbage-in/garbage-out* phenomenon is inherent in any good system: if you feed it with the right data, it performs well, but without good input data, you better do not trust its outcomes.

Network Structure So far, it seems that the ES method is a clear winner and outperforms the traditional EVM method in all situations. However, if it sounds too good to be true, it probably is. There is indeed a catch: the ES method does not always work well, and the catch is in the structure of the project network. Despite the observation that the ES method outperforms the two others traditional methods along the whole project life cycle, the results were not always favourable. As a matter of fact, the main reason why we have tested these EVM/ES methods on more than 4000 projects was to make sure that we could span the *full range of complexity*, guaranteeing that any project that could possibly exist in reality was included in our project dataset (more on this *full range of complexity* concept will be explained in the artificial project data Chap. 11). The generated dataset with artificial projects contained projects with a variable network structure ranging from very parallel networks to completely serial networks, and our experiments revealed that the structure of the project network has a clear impact on the accuracy of the EVM/ES methods. We found that the serial/parallel indicator (SP), which measures the degree of how serial or parallel the project network is, was the main driver that determines the accuracy of all EVM/ES predictions. The results clearly showed that the EVM/ES methods performed best when projects are closer to a serial-structured network, with ES outperforming the EVM methods. However, for project networks closer to a completely parallel structure, none of the methods (neither EVM nor ES) performed well, and the accuracy was so low that it was better not to rely on their predictions at all. Consequently, the main conclusion of the simulation study was that EVM, and certainly ES, works well for serial networks but fails miserably for parallel networks. As I will discuss later in Chap. 7, this observation is (*in hindsight*) very straightforward, but I could never have come to this conclusion without the help of our simulation experiments. Mission #1 of academic research is accomplished.

4.3 Thank You, Tony

This chapter described my early efforts at forecasting the duration of projects with earned value methods in collaboration with Stephan and Walt who have shaped the future of my research in several ways. I have deliberately not gone into too much detail, as many of the results of these two studies will be discussed in several later chapters of this book. The two publications with Stephan were the spark of much more work, which eventually resulted in my award-winning book *"Measuring Time"* (Vanhoucke, 2010). I will never forget my *fifteen minutes of fame* when I presented my research during the award ceremony in Rome (Italy) at the World Congress of the *International Project Management Association* in 2008. Today I look back on these studies with nostalgia, remembering our regular trips to London (UK) and Geneva (Switzerland) and our thousands of emails trying to convince people of the power of the new ES methodology. Over time, my book has become my personal best seller and has given me the opportunity to present my research all over the world.

A *best-selling* book should be taken with a pinch of salt in the project management domain. I wish I could say it has sold more than a million copies worldwide, but I am happy with the several thousand copies and the positive attention the book has received. I am inclined to believe that the book has set a number of things in motion, as many academics now use the ES method in their studies, which was not the case at all before the book publication. I still experience academic research as a rewarding job which brings joy in my life, and although as a teenager I dreamed of becoming a rock star (preferred choice) or a very famous writer (second option), I am now happy with the positive feedback for my books on data-driven project management. One of the nicest complements I ever got for my research achievements came from Tony Barrett, an American Professional Engineer (PE), Earned Value Professional (EVP), and Project Management Professional (PMP) who wrote on the LinkedIn Earned Value Management discussion the following words:

> *Professor Vanhoucke's work is shedding a new light on using EVM for me. In retrospect, this has helped me understand better why EVM worked so well in some cases and failed so miserably in others.*

I am sure that Tony did not need any of my research experiments to fully realise what he was doing in his professional job, but I dare to think that the results have confirmed his gut feeling and therefore improved his understanding in his work. I think Tony's comment is a good example of how academics can help professionals to better understand the methodologies they use. However, while a better understanding of existing techniques can be a noble goal of academic research (Mission #1), it is often insufficient and brings a lot of new ideas on the research agenda. The search to new and better methods is another phase in the story of academic research, and this new step is described in the next chapter as Mission #2 (*wisdom*).

References

Anbari, F. (2003). Earned value project management method and extensions. *Project Management Journal, 34*(4), 12–23.

Jacob, D. (2003). Forecasting project schedule completion with earned value metrics. *The Measurable News, March:1*, 7–9.

Lipke, W. (2003). Schedule is different. *The Measurable News, Summer*, 31–34.

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*, 289–302.

Vanhoucke, M. (2010). *Measuring time – improving project performance using earned value management* (Vol. 136). International Series in Operations Research and Management Science. Springer.

Vanhoucke, M., & Vandevoorde, S. (2007). A simulation and evaluation of earned value metrics to forecast the project duration. *Journal of the Operational Research Society, 58*, 1361–1374.

Chapter 5
Wisdom

After some crazy years of travelling and attending workshops, a successful collaboration with two published papers and an award-winning book on Earned Value Management (EVM), it was time to shift from Mission #1 (*creating understanding*) to Mission #2 (*providing wisdom*). I felt the need to extend my team with researchers who are as passionate as I am and who are able and willing to join my (then small) OR&S group to expand the work we had done so far. We had not developed any new methodologies for EVM yet, and all we had done so far was testing and comparing the existing EVM methodologies to improve our understanding for project time forecasting. The results, discussed in the previous chapter, not only provided us with more understanding, but also with new ideas to improve the quality of the predictions. Even from the professional field, I felt the willingness (from some project managers) to incorporate some changes in the current EVM methods, and they argued that their better understanding (especially in cases when the current methods failed (mostly for parallel projects)) made them think about possible improvements. I believe this is exactly the desired side effect of Mission #1 of academic research: it makes us more ambitious and eager to learn more.

Mission #2 is indeed much more ambitious, as it aims at *extending* the existing methodologies based on the lessons learned from Mission #1. While the first mission aimed at understanding *why* EVM/ES systems work or fail, the second mission focuses on *how* this understanding can be transformed into wisdom. Wisdom is used here as a collective term to go beyond the mere understanding by making the current methodologies richer and more complete, by adding additional features or integrating them into a broader framework, or by even completely replacing them by other, hopefully better alternative methodologies. Since it lies in the nature of academic research to shift boundaries and replace current understanding by new, alternative knowledge, I knew I needed a team, and so I submitted a big research proposal that eventually resulted in the fantastic team I have today. I doubt whether the American rock singer and guitarist Jimi Hendrix had an interest in data-driven

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_5

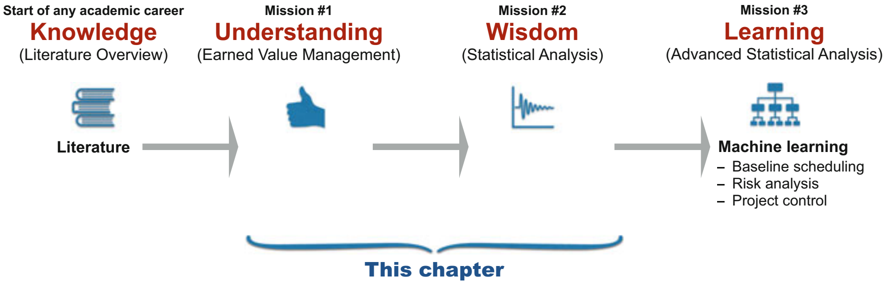


Fig. 5.1 The transition from *understanding* to *wisdom*

project management, but I will nevertheless use his quote to introduce Mission #2 of academic research in this chapter. It goes as follows:

> Knowledge speaks, but wisdom listens.

The attentive readers will probably remember that in the introduction to Part II of this book I wrote down a four-line quote[1] that would be repeated in bits in Chaps. 4 to 6 (for example, Jimi Hendrix's quote consists of lines 1 and 3 of the full four-line quote). The full quote should not be taken too seriously, as the four lines served purely to represent the general framework of academic research as the three-mission process "*understanding*", "*wisdom*", and " *learning*". Nevertheless, I believe it is now time to come back to the full quote and explain its four lines in detail in the following paragraphs. This is best done by referring to Fig. 5.1 graphically depicting the true nature of academic research:

- *Knowledge speaks* refers to the importance of knowing what the current state-of-the-art is in the academic literature. Every PhD student knows that the first few months of an academic career path consist of reading, reading and nothing but reading. By exploring the relevant papers in the academic literature and trying to make sense of them, researchers get an idea of the current knowledge in academia, and I sometimes feel that young and enthusiastic researchers underestimate the importance of understanding the current state of knowledge early in their careers. You cannot become a good researcher if you do not know what others have done before you, which is why I think a literature review is one of the most important steps in starting an academic career.
- *Understanding explains* refers to the process of investigating current ways of managing projects without adding new features or creating totally new methodologies. This Mission #1 was illustrated in Chap. 4 by the comparison study of the three *existing* forecasting methods on three sample projects. The published paper does not contain any novel idea nor any new methodology and focuses solely on the comparison of existing methods. In the current Chap. 5, a new

[1] You may recall that the full quote sounded like this: *Knowledge speaks, understanding explains, wisdom listens, and learning changes*.

existing methodology will be introduced referred to as *rules of thumb* for project control that exists of simple and straightforward methods to interpret warning signals during the progress of the project. Again, these rules do not contain any new elements but are nevertheless widely used in practice by professional project managers.

- *Wisdom listens* goes further, much further, than merely creating understanding. The nature of research is not only to provide a better understanding of the current existing methodologies, but primarily to shift the boundaries of our current knowledge and explore new territories by walking on untrodden paths and creating ideas no one else had before. This Mission #2 is the topic of the current chapter and will extend the simple rules of thumb to four types projects of control with *statistical tolerance limits*. This new approach of controlling projects will not fundamentally change the way projects are managed and still relies on the EVM/ES methods discussed earlier, but aims at providing (marginal) improvements to create a statistical project control system with a higher accuracy and reliability.
- *Learning changes* will be the theme of Chap. 6 and aims at exploring the very limits of our current understanding without caring about the practical implications or relevance. Rather than aiming for *marginal* improvements, Mission #3 searches for *radical* improvements by introducing totally new ideas and advanced algorithms that no one has used before in this research domain. It will be shown in the next chapter that machine learning algorithms can be used for the three components of the data-driven project management framework (*baseline scheduling*, *risk analysis*, and *project control*).

The story of this chapter is the story of Jeroen Colin, a researcher who joined the OR&S group in 2009 and successfully defended his PhD thesis in 2015 with six published papers on *statistical project control* methods using *tolerance limits*. When he joined the team, he expressed his interest in statistical data analyses, and he quickly developed coding skills in R and Python and had heard about the supercomputer infrastructure that Ghent University would install in the next coming years.[2] He told me that he had read the paper on the comparison study and believed that the *earned value methodology* could and should be used in a much better way than it is done now (in 2009). I warned him that not everybody is ready for accepting changes to EVM and told him about the difficulties I had when introducing the *earned schedule* method in the literature, but that could not stop him from starting his challenging (and sometimes very difficult) journey into the world of statistical project control.

Jeroen's story perfectly illustrates the challenging nature of the second mission of academic research, as it took him more than 4 years and an endless stream of suggestions for changes before his first paper was finally accepted in the flagship

[2] Ghent University announced three years later (in 2012) the introduction of the first Flemish supercomputer as a High-Performance Computing (HPC) system with a price tag of about €4.2 million.

journal *Omega—The International Journal of Management Science* (Colin and Vanhoucke, 2014). I think that not many reviewers were ready to accept the proposed changes in the existing EVM/ES systems, which is why they first objected and proposed to simplify the research study, but finally accepted because they had no other choice than admitting that this new statistical way of analysing EVM data made much sense. After his first publication, the ball started rolling, and Jeroen presented a wealth of advanced statistical project control methods, and his five subsequent papers were all accepted in a period of 2 years after the first acceptance. Once people accept the changes, they are no longer afraid for wanting more. Let us take a look at this new way of controlling projects in the next sections.

5.1 Tolerance Limits

The four different control methodologies presented in this chapter all rely on the EVM[3] metrics discussed in the previous chapter, i.e., the key input metrics (planned value (PV), actual costs (AC), and earned value (EV)) and the schedule and cost performance measures (SPI, SPI(t), and CPI). Consequently, from a professional point of view, the project manager must not add anything to the currently used control system to measure the performance of the project in progress, and the new methods presented in the current chapter rely on the same system input metrics as before. However, the major difference with the traditional control systems lies in the fact that the EVM metrics are now used to construct the so-called *control limits* or *tolerance limits* to automatically warn the user when the project is out of control. The incorporation of control limits in EVM is an interesting feature to support the decision maker with warning signals to trigger actions when the project tends to run into the danger zone. Consequently, these control limits act as *action thresholds* and tell the user when it is time for action to bring the project back on track. The traditional EVM systems also used the schedule and cost performance metrics as warning signals, but they required interpretation and did only report the current project status (e.g., early, on time, or delay) without telling anything about the possible impact on the total project duration and the necessity for actions. Therefore, the professional manager had to fall back on some arbitrary action thresholds (e.g., when SPI(t) falls below 70%, it is time for action) using intuition and past experience. Instead, the new action thresholds of the four methods presented in this chapter rely on a statistical data analysis to automatically warn the manager to take actions, and this new control approach is therefore described as *statistical project control*.

[3] I will from now on refer to an EVM system as a project control system with traditional EVM metrics *and* the new ES metric and will no longer make a distinction between the two by calling it an EVM/ES system as I did in the previous chapters.

The following paragraphs present a three-phased approach used in the statistical project control study to test the relevance of tolerance limits for predicting the necessity of corrective actions. An important difference will be made between two types of Monte Carlo simulations. A *static simulation* run will be used to construct the control limits, and a *dynamic simulation* run will be applied to test these control limits for real project control. Since this dual simulation approach (*static* and *dynamic*) will be used throughout many chapters of this book, it is worth spending some pages on it.

Phase 1. Baseline Schedule The construction of a baseline schedule acts, as always, as the point of reference for constructing control limits and monitoring the project progress, as I have discussed earlier in Chap. 3. More precisely, the baseline schedule provides the *planned value* (PV) curve as the cumulative increase of the planned cost of the activities in the schedule. The PV curve is said to be a static key metric since it is known from the very beginning, prior to the project start. It will be used to monitor the performance of the project once it is in progress by calculating the *earned value* (EV) and *earned schedule* (ES) metrics to measure the project performance at periodic review periods. Figure 5.2 shows an example project Gantt chart (i.e., the baseline schedule) in ProTrack (the software tool I introduced in Chap. 1), and the line at the bottom of this picture titled *"expected project duration"* is the planned value line that shows the cumulative increase in the planned cost according to this Gantt chart. The two other EVM key metrics (AC and EV) are obviously not shown in the picture since they are unknown at the project start.

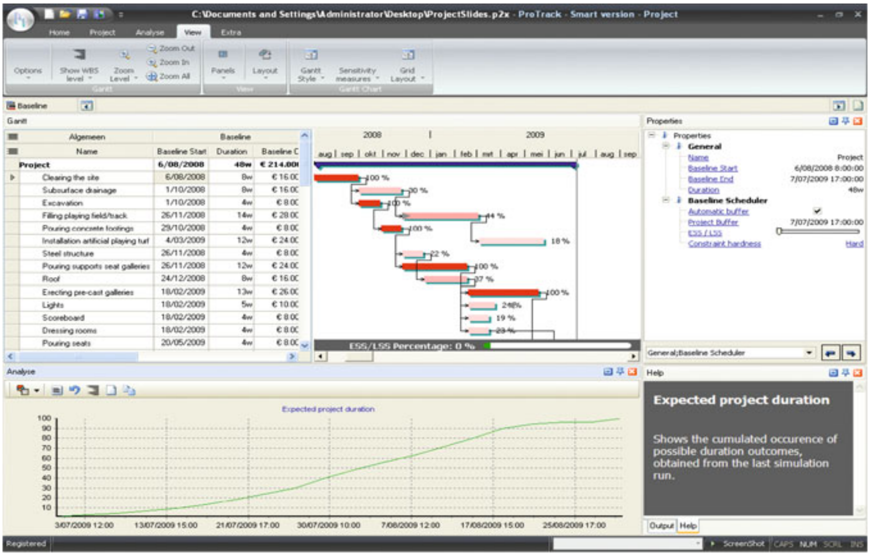


Fig. 5.2 A baseline schedule (Gantt chart and planned value line)

In the next two phases, two different types of Monte Carlo simulations are used to imitate the progress of the project, and it is important to understand the difference between these two types. The *static simulation* of Phase 2 imitates the project progress prior to the real project start and only serves as a simulation run to gather data to construct the tolerance limits. This simulation is identical to the simulation runs of a schedule risk analysis (Chap. 3) for obtaining the values for the activity sensitivity metrics, but it now gathers data for the schedule and cost performance metrics to construct the tolerance limits. The *dynamic simulation* of Phase 3 is different, as it is a way to imitate the *real* project progress and test the quality of the tolerance limits in a real-life setting. This second type of simulation therefore assumes that the project is now in progress and makes use of the tolerance limits of Phase 2 to find out whether they are indeed doing what they should do, i.e., warning the project manager when problems require actions. It is important to understand that this second simulation run is only relevant to academic researchers to test the control limits but will never be used in a real professional setting. Indeed, real projects are not dynamically simulated but are subject to the real project progress with its inevitable problems and necessary actions. Since the real world is not the ideal playground for academic researchers to test their ideas, they have to fall back on dynamic simulation runs to imitate this reality under uncertainty. These two types of simulations runs are discussed in detail in the following paragraphs.

Phase 2. Construct Tolerance Limits (*Static Simulation*) The aim of running a static simulation is to create a wealth of data to construct the tolerance limits for project control. At each run of these static simulations, the two dynamic key metrics of EVM (*actual cost* (AC) and *earned value* (EV)) are periodically generated and compared with the static PV metric to calculate the schedule (SPI or SPI(t)) and cost (CPI) performance metrics. Since each simulation run imitates a slightly different project progress, the values of the performance metrics will differ in each run and each review period, and after a couple of thousands of runs, the static simulation run ends with a range of values for each review period of the project that will be used to create the tolerance limits. These tolerance limits can be calculated as simple $1 - \alpha$ confidence intervals (e.g., a 95% interval for $\alpha = 0.05$), which results in lower and upper tolerance limits containing most of the simulated data (only $\frac{\alpha}{2}$ percent of the simulated values fall under the lower tolerance limit and $\frac{\alpha}{2}$ percent lie above the upper tolerance limit).

Since these tolerance limits are calculated on the cost and schedule performance metrics (CPI, SPI, and SPI(t)), the static simulation simply simulates variability in the project by drawing random numbers from predefined probability distributions for the activity durations. It recognises that the baseline schedule is only a reference point that will be subject to natural variability during the project progress. The concept of *natural variability* comes from the quality control literature to accept that any process (including a project) is always subject to inherent uncertainty as the cumulative effect of several minor factors that cannot be predicted. Consequently, describing this natural variability as *acceptable variation* requires the definition of a *desirable state of project progress* expressed by the parameters of the probability

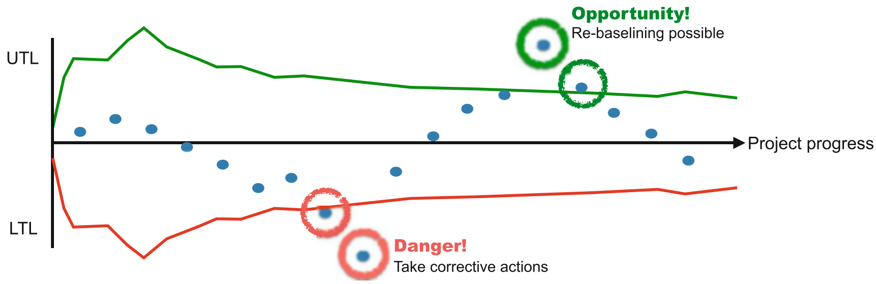


Fig. 5.3 A project control chart (control limits and periodic measurements)

distributions used in the static simulation run. Any variability coming from these distributions is considered as natural and should not require any actions, and all values within the tolerance limits will therefore contain acceptable values for the cost and schedule performance indicators for each review period of the project. Any value beyond these limits is assumed not to be acceptable and must have an unnatural cause of variability.

Figure 5.3 displays an illustrative *control chart* with the lower tolerance limit (LTL) and upper tolerance limit (UTL) for one of the performance metrics (e.g., for SPI(t)). These tolerance limits are not set as straight lines but have an irregular shape since they are calculated for each review period from the simulated data. This irregularity illustrates the power of the static simulation runs and shows that the tolerance limits differ along the stages of the project (between start and end). At certain stages of the imitated project progress, a different number of activities can be in progress, each having a different duration and cost resulting in very different values for the cost and schedule performance metrics. Hence, choosing the right (i.e., most realistic) parameters for the probability distributions for defining the desirable project progress is of the utmost importance and should be done with care and knowledge about the expected behaviour of the project progress.[4] The dots within the control chart are periodic observations of the true status of the project progress and are the subject of Phase 3 (dynamic simulation or real project progress). It clearly shows that the static tolerance limits can serve as action thresholds in Phase 3, which means that exceeding these limits definitely indicates an abnormal project behaviour (a good one or a bad one) that should be carefully investigated, and when necessary, solved by corrective actions. In Jeroen's first paper (Colin and Vanhoucke, 2014), two types of control charts were presented in line with the traditional quality control literature. A so-called *X chart* will be used to monitor the individual observations of the project performance indicators (e.g., the dots in the figure represent the CPI or SPI(t) values along the project progress), while an *R chart* monitors the difference between two adjacent observations of these

[4] Chapter 14 presents a *calibration procedure* to find realistic values for the parameters of a log-normal distribution to model variability in activity durations using empirical data of past projects.

project indicators. The tolerance limits of both charts can be calculated using the data generated by the static simulation runs.

Despite the simplicity of running static simulations for creating the control charts with tolerance limits (from an academic point of view), not many project managers will easily take the step to this new approach for different reasons. First and foremost, they will argue that the parameters of the probability distributions to model activity duration variability can only be realistically estimated after analysing past projects similar to the new project. While I fully recognise that this is indeed not an easy task, I would argue that this should not be an excuse to *not* use this technique for controlling real projects. As will be shown in later chapters, a good analysis of past projects has multiple benefits and should be done regardless of whether you want to use Monte Carlo simulations or not. A second reason why project managers will not be inclined to embrace this new technique is the need for the Monte Carlo simulations to create these control charts. Nevertheless, I try again and again to convince project managers that running simulations has a lot of value and is not as difficult as they sometimes think. Moreover, many books have been written (outside the domain of project management) where the use of Monte Carlo simulation brought many advantages. One of the most enjoyable books I have ever read, and one that I definitely want to recommend to you is Nicholas Taleb's book titled *Fooled by Randomness* (Taleb, 2012). Besides the fact that I am a huge fan of the author and his amazing writing style, after reading this book, you cannot help but conclude that the use of simulation is a necessity to better understand risk. I also want to add a last quote here to end this second phase of the static simulation runs from the American academic and computer scientist Nicholas Negroponte who perfectly summarised the value of simulations as follows:

> Learning by doing, peer-to-peer teaching, and computer simulation
> *are all part of the same equation.*

I advise every project manager to try out the simulation tool for themselves. It is a small step for a manager but a giant leap for project management.

Phase 3. Project Progress (*Dynamic Simulation*) In the third phase, the project manager is responsible for managing and controlling the progress of the project using periodic observations of the project's status and comparing them with the tolerance limits of Phase 2. More specifically, the manager collects data for the EVM key metrics (PV, AC, EV, and its time-indexed version ES) at each review period to calculate the schedule and cost performance indicators. These periodic observations define the current project status and must be plotted in the control charts, as shown by the dots in Fig. 5.3. When these observed performance indicators exceed the tolerance limits, it is a clear indication that the project no longer is subject to acceptable variability and much more is going on. At that point, the project manager should be careful and use this indication as a warning that there is unacceptable variability in the project, requiring additional attention and possibly some actions. Consequently, control charts must be used as a data-driven project control system to report warning signals that serve as triggers for corrective actions. The system is based on a series of consecutive hypothesis tests using the following hypotheses at each review period (i.e., for each dot in the control chart):

H_0: The project is executed as planned (acceptable variation).
H_a: The project is not executed as planned (unacceptable variation)

When the observed performance (dots) falls within the tolerance limits, the null hypothesis is accepted and no real problems have occurred. In that case, it is assumed that so far the project is executed as planned with some acceptable variation (defined in Phase 2). However, when the observed dots exceed these tolerance limits, the alternative hypothesis must be accepted denoting that the project progress is no longer subject to acceptable variability. More precisely, when an observation falls below the lower control limit, the performance has likely dropped below an acceptable margin of variability compared to the baseline schedule, in which case the project manager should consider taking corrective actions. Likewise, when the observed performance indicator exceeds the upper control limit, it must be seen as a signal to exploit opportunities, as the performance is better than initially planned. In such case, the project manager might consider to re-baseline the project and exploit these opportunities to sharpen the future expected outcome of the project.

As I have outlined earlier, the project monitoring process of real-life projects (Phase 3) does not make use of dynamic simulations, since the project manager is in charge of a project in progress and monitors its performance with the control charts until it is finished. However, academic researchers do not manage and control real projects but want to test the quality of control limits under a wide range of possible circumstances. Therefore, the project progress of Phase 3 must be dynamically simulated to imitate the real project progress. This dynamic simulation must be carried out under a wide range of settings (e.g., early projects and late projects with just a few or a lot of problems) to test whether these control charts indeed generate warning signals when they are really necessary. The specific design for this dynamic simulation is outside the scope of this chapter and will be outlined in detail in Chap. 12 in which three dynamic simulation models will be proposed.

Practical Implementation Our research on project control using control charts with tolerance limits has received positive reviews from academia but also attracted the attention of some professional project managers with whom I was working at the time. Based on their request, we implemented these control charts in ProTrack to test them with real project data to find out whether they could add value in reality. Since the method works for every possible performance measure (not just the SPI or CPI), we had to make a choice during implementation for which performance measure we would calculate the tolerance limits to monitor project progress. As a matter of fact, the power of the static simulation runs specifically lies in the fact that they can generate an overwhelming wealth of project data for *any* performance metric without much additional effort. Consequently, any performance metric available in an EVM system can be used to construct the control charts. Figure 5.4 displays our control chart dashboard in ProTrack based on twelve different performance indicators. Each control chart represents another indicator coming from the same

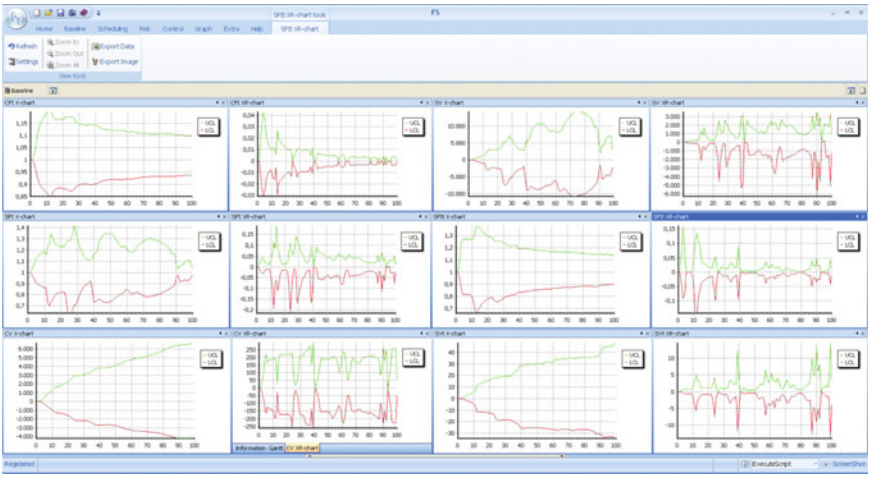


Fig. 5.4 Control charts for different performance measures

static simulation runs, and despite their difference in interpretation, their use in a control chart remains exactly the same: When any of the thresholds are exceeded, the project manager must treat this as a warning signal for unacceptable problems that might require corrective actions.

Apart from the choice of which performance metric to display, an additional choice had to be made. The different performance metrics in the control charts can be measured at various levels in the project network, ranging from individual activity control (*constructing a chart for each activity*) to general project control (*constructing one control chart for the project*). Both extremes are unlikely to lead to the best results, and the truth, as always, lies somewhere in between. I have already discussed in Sect. 3.3 of Chap. 3 that the *right level of detail* is an extremely important choice for an EVM system. Therefore, this choice should also be taken with some caution for the control charts, and this is why I will spend a detailed discussion on this topic in the next section.

5.2 Control Points

Recall that the control chart approach with statistical tolerance limits has been classified under the general umbrella of *wisdom* to refer to the fact that this method is an extension of the traditional EVM control method normally used by professional project managers (which were classified under the *understanding* mission of Chap. 4). There is no doubt that the use of control charts to generate warning signals for corrective actions does not belong to the current toolset of project managers for controlling projects, but that does not mean that the idea is completely new. As a matter of fact, any project manager implicitly relies on threshold limits to indicate whether actions are necessary, but they do not do it using the static simulation runs (Phase 2) discussed in the previous section. This section will review five different control methodologies, consisting of one well-known method in practice that everybody uses (*understanding*) and four implementations of the control chart methodology discussed in this chapter (*wisdom*). This section is a summary of a follow-up paper published in Colin and Vanhoucke (2015) to provide an answer to the controversy about the right level of detail for project control.

The first method consists of the traditional use of EVM used by most professional projects managers and is briefly described in the following paragraph.

Rules of Thumb (ROT) Most project managers who use EVM for project control measure the project status by the classic performance indicators (SPI and CPI) and set some arbitrary values (thresholds) based on their intuition, experience or using simple rules of thumb (e.g., when SPI(t) falls below 70%, it is time for action). This straightforward way of controlling projects makes use of EVM data during project progress without any statistical analysis or simulation runs. Instead, these control limits are set manually by the project manager and will therefore look less irregular than the statistical tolerance limits of Fig. 5.3. In our study, we have defined three different control limits to imitate how project managers would define arbitrary rules of thumb to define their control thresholds. In a first method, the control limits are defined as straight horizontal lines, e.g., to express that the SPI cannot drop below a level of 70%. These so-called *static control limits* assume that the thresholds have fixed values along the project progress and will never be updated. Two other methods are referred to as *narrowing control limits* or *widening control limits* as they change the values for the thresholds during the project's progress. The narrowing limits start with extreme values (e.g., 40%) to denote that schedule deviations are acceptable in the early stages of the project, but the values are narrowed towards the later stages (e.g., 90%) to denote that any slight deviation from the schedule requires immediate action. The widening limits work exactly in the opposite way and allow no major deviations early in the project but become more relaxed in the later stages of the project. This simple ROT method is the easiest way in which EVM can be used for project control without the need for statistical data (probability distributions) and static simulations and is therefore classified under Mission #1 (*understanding*). Most project managers that I have met implicitly use this method, but they do not know how to define the best values for the thresholds. I have met people who argue that the performance should never drop below 90% during the early stages of a project, but no one could give me a reasonable argument as to why the 90% value is so important. I often feel that managers use a default value as some kind of proof that their system will detect any problem, but often no data analysis has preceded it. Too much trust in such a system results in a tendency to overestimate how much control someone has over the outcome of uncontrollable events and creates an *illusion of control*.[5] When I had to choose an appropriate title for this book, I suddenly thought of a quote from the fiction writer Fred Bubbers that fits perfectly into the discussion of this first control method. The quote goes like this:

> Control is merely an illusion we construct
> to cope with the chaos that is reality.

Illusions should be avoided at all costs, and the use of statistical tolerance limits instead of arbitrary rules of thumb can play an important role in this. Creating control charts using static simulation runs followed by a sequence of hypothesis tests to control projects is obviously a better and more objective method than using arbitrary rules. However, this method must, as mentioned earlier, be applied with the correct level of detail. The level of detail may vary according to the number of control charts used to monitor the performance of the different components (activities or groups of activities) of the project. In the remainder of the section, the number of control charts is defined by the number of *control points* in the project network. A control point is defined as a certain place in the project network that tracks the performance of a subset of the activities of the whole project. A separate control chart is then constructed for each control point, based on the statistical approach of the previous section, and each chart only monitors the progress of a subset of the project network. Consequently, the number of control points defines the level of detail in the project control process. Figure 5.5 displays an example project network to illustrate how four different control points can be defined in a project network. Since the activity durations are displayed above each activity (node) in the network, it is not difficult to see that the critical path of this project consists of activities Start - 2 - 5 - 8 - End. The four different control points are shown in the networks below the project network and discussed along the following paragraphs.

Project Network Control (PNC) In the description of the previous section, it was implicitly assumed that only one control point was set at the end of the project such that the performance of the complete set of activities is monitored at each review period of the project (shown by the dots in Fig. 5.3). This system is the most straightforward implementation of statistical project control since the control chart monitors the progress of the project for *all* activities that are in progress. It requires only one control chart[6] to periodically monitor the project performance from its start to end and is therefore the system with the lowest level of detail. Figure 5.5 displays this so-called project network control point at the end of the project network.

Feeding Path Control (FPC) The concept of feeding path control is inspired by the critical chain/buffer management methodology of Goldratt (1997), in which

---

[5] In case you have not figured it out yet, this is the title of the book you have now in your hands.

[6] I argued earlier that for any project performance indicator, one can use two control charts (X and R chart), and since EVM proposes several performance indicators (e.g., 12 in Fig. 5.4), a single control point can still result in multiple control charts.

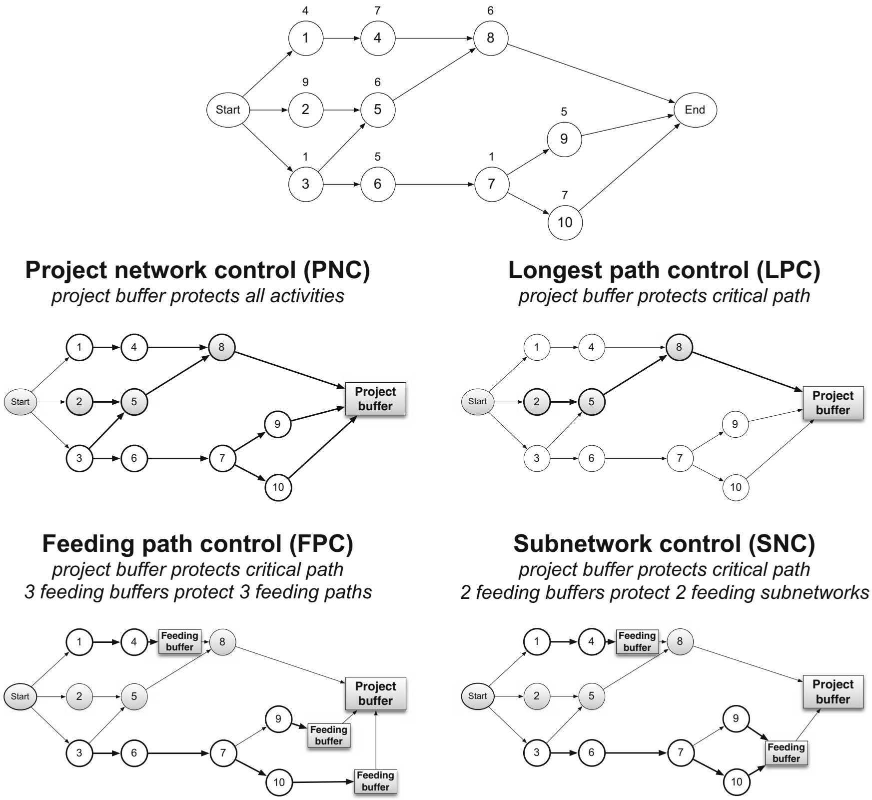


Fig. 5.5 Control points to construct control charts

feeding paths are defined as any path in the project network that enters the critical path. The feeding path control method calculates the tolerance limits for each feeding path separately, as well as for the critical path, and constructs several control charts to monitor the performance of the different paths in the project network. Figure 5.5 shows the four different control points for four different network chains. One control chart is displayed at the end of the project and monitors the progress of the activity on the critical path (*project chart*), while the others monitor the progress of the activities on the feeding chains (*feeding charts*) entering this critical path. This approach represents a more granular control method than the project network control method, as each control chart only monitors the performance of a subpart of the project network, requiring more observations and performance metrics to be measured. The number of control points can grow rapidly for large project networks, making it more difficult for this method to get an overview of the actual performance of the project due to the large number of control charts that must be checked during execution.

Subnetwork Control (SNC) The subnetwork control method tries to find a middle ground between the two previous extreme methods by using more control charts than the single chart of the PNC method but less than the large number of charts in the FPC method (for the critical path and different feeding paths). More specifically, the method sets control points at each subnetwork instead of each feeding path, which will significantly reduce the number of charts. A subnetwork is defined as a collection of paths that enter the critical path and thus collectively form a subnetwork in the total project network. In the example of Fig. 5.5, the two feeding paths (path 3 - 6 - 7 - 9, and path 10) that get in the FPC method each a separate control chart are merged into one subnetwork with only one remaining chart. The figure displays three control points in total, one for the critical path (project chart), and two feeding charts for the two subnetworks entering the critical path.

Longest Path Control (LPC) The longest path control method is a special case because it no longer monitors the *entire* project with several control points but only a part of the project with a single control point. However, the single control chart of the LPC is used in a different way than for the ROT and PNC methods. Indeed, these methods also only use one control point, but these were used to monitor the entire project (including all activities), while the FPC method only monitors the activities on the longest path, completely ignoring the other activities. The logic of the LPC method is inspired by the results of the research study discussed in Chap. 4 in which it was shown that the network structure has a major impact on the accuracy and reliability of EVM systems. Indeed, in Sect. 4.2, I stated that EVM-based schedule control metrics are much more reliable for *serial networks* but have a poor performance for *parallel networks*. My friend Walt took this idea a step further in a new study (Lipke, 2012) and suggested to just look at the longest path in any network (critical path) by ignoring all the remaining (non-critical) activities. Since the critical path in a network is by definition always completely serial, the EVM methods should work much better on this serial part of the network. The other activities of the network are then superfluous because they make the network more parallel, and we know that a parallel network lowers the accuracy and reliability of the system. Figure 5.5 displays this control chart at the project network but only monitors the critical path activities. While the obvious advantage of this method is that it only uses one single control chart, the difficulty lies in the fact that the critical path continuously changes along the project progress. Therefore, this LPC method requires a continuous update of the critical path along the project progress, which gives the usability of this method a big blow.

In the next section, the performance of these five project control methods will be tested using three well-known quality metrics from statistical hypothesis testing, and some comparative results are shown based on a computational experiment on a large set of artificial projects.

5.3 Signal Quality

In the comparative study of Chap. 4, the quality of an EVM system was validated using two straightforward metrics that measure the *accuracy* and *reliability* of predictions. The *accuracy* was defined as the average deviation between the predictions and the real project outcome (known once the project is over), while the *reliability* measured the impact of true or false warning signals on the quality of the predictions. These two metrics are easy to understand but are not well-defined and therefore not usable in the study discussed in this chapter. Since the study on control charts takes a statistical point of view, they should be validated with good and well-defined metrics to measure the signal quality of the charts in an unambiguous way. This section presents three quality metrics to assess the performance of the different control charts of the previous section, along with some results of the performance of the five control chart methods. Since the use of control charts with statistical tolerance limit methods can be reduced to a sequence of hypothesis tests, the quality of their generated signals should be validated by the *classic* criteria used in this statistical field. It is known that the conclusions drawn from a hypothesis test can be subject to two errors (Type I and Type II) that will be used to evaluate the five control methods discussed in the previous section. A summary of the quality metrics is given in Fig. 5.6 and discussed along the following paragraphs.

Probability of Overreactions The probability of overreaction is equal to the frequency that the control method generates a warning signal (because the tolerance limits have been exceeded) when in fact there is no problem at all and the project activities are still running according to plan (within the allowable variability). This probability is known in statistics as the *Type I error* or as the *false positives* and measures the probability that the null hypothesis is rejected when it should actually be accepted. Obviously, this Type I error should be as low as possible, since it creates a false warning signal to the project manager that actions are necessary while actually they are not. Since the control charts are set up for control points to monitor a group of activities (using 5 different versions as discussed earlier), each warning signal requires a search for the cause of the problem by looking for the activities that actually are in trouble, and actions should be taken to resolve the problem of these activities. But when there are actually no problems, the project manager has

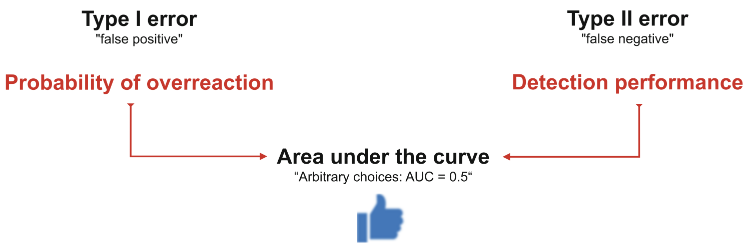


Fig. 5.6 Hypothesis testing for statistical project control

needlessly carried out this search, which entails valuable time and effort, which should be avoided at all times.[7]

Detection Performance The detection performance measures the frequency with which the control chart generates a warning signal to report a project problem (by exceeding the tolerance limits) that actually proves to be a correct signal because some activities are no longer performed according to the planned acceptable variability. This detection performance should be as high as possible, and ideally, each problem should be detected by the control charts. Of course, no system is perfect, and a number of problems sometimes remain under the radar and are not detected by the control charts. The detection performance is measured in statistics by the Type II error; more specifically, the performance is equal to 1—Type II error. The error is also referred to as the *false negatives* and measures the failure to reject the null hypothesis when the alternative hypothesis is actually true.

Both types of errors are in conflict with each other, meaning that decreasing one error increases the second error. An increase or decrease in error can easily be achieved by narrowing or broadening the tolerance limits, but since they cannot both be improved at the same time, a difficult choice has to be made when composing the tolerance limits. Widening the tolerance limits will likely result in fewer warning signals being generated, which also means that real problems will be less easily detected. Thus, the Type II error will increase, but the Type I error will decrease. An inverse reasoning can be made for narrowing the boundaries. This trade-off between the two error types can be perfectly measured by a third quality metric that can be visualised as a curve showing the relation between the two errors, as explained in the next paragraph.

The graph showing both errors is called the *receiver operating characteristic* (ROC) curve and is illustrated in Fig. 5.7. The curve shows the trade-off between the two errors that should be as high as possible. The perfect system is in the top left corner of the graph, where the detection performance is 100% (any error is detected) and the probability of overreactions is 0% (no signal is ever given if there is no actual error). A real system does not achieve this ideal situation, which is the reason why this curve can be used to measure the quality of the system. This quality is measured by the *area under the curve* (AUC), a metric that perfectly integrates both types of errors. As noted earlier, the perfect system has an AUC = 100%. A completely random system has an AUC of 50%, which makes the system not better than flipping a coin as a warning signal (head gives you a signal, tail means everything under control). The quality of a sound warning system is thus measured by the AUC value, which must be greater than 50% and as close to the perfect 100% value as possible.

The five control methods have been experimentally tested on a large set of artificial projects under different settings with different parameters for the probability distributions used for the two simulation runs (Phase 2 and Phase 3).[8] In each experiment, the tolerance limits have been varied for the control charts to obtain

[7] In Chap. 7, this search for problems will be called a *top-down project control* approach, and the effort taken to search for problems will be integrated in a new *efficiency of control* concept.

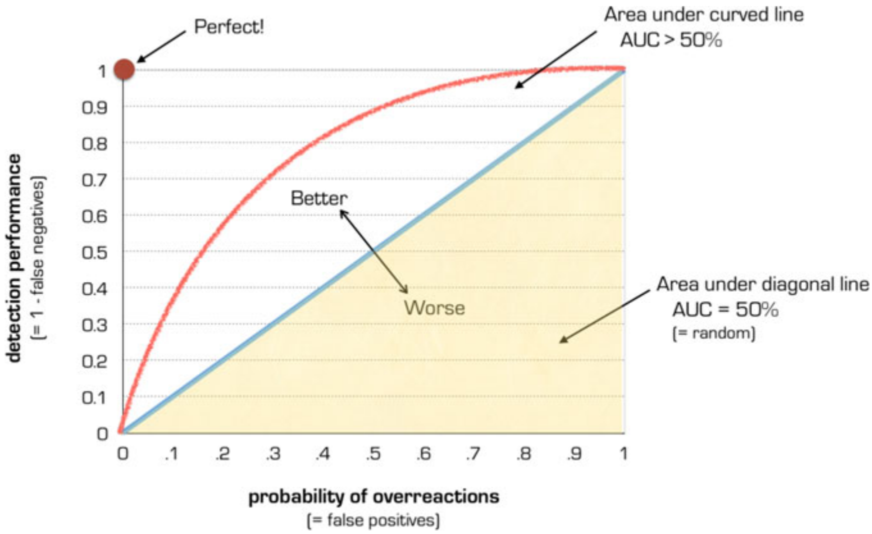


Fig. 5.7 An illustrative receiver operating characteristic (ROC) curve

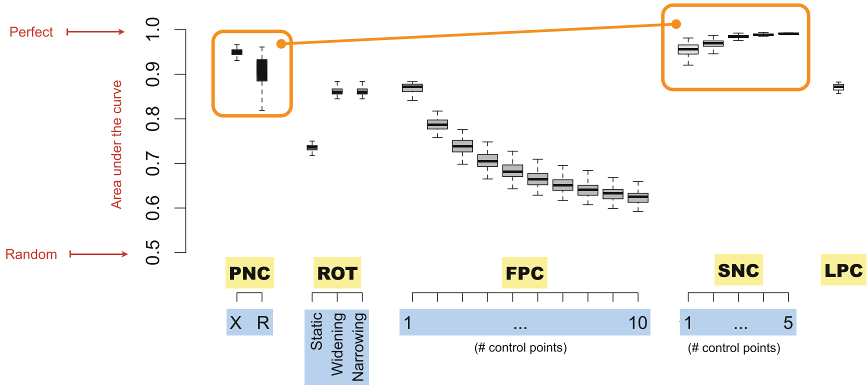


Fig. 5.8 Results of the statistical project control study

many different values for the Type I and Type II errors. After a huge number of experimental runs, these errors can be used to draw the ROC curve and calculate the AUC metric. The results of these experiments are summarised in Fig. 5.8 for the five control methods of the previous section (displayed on the X-axis).

These AUC values are displayed as box plots on the vertical axis and show the *interquartile range* (IQR) (the box) and the difference between the minimum

[8] The relevance of artificial projects for computational experiments will be explained later in Chap. 11, and the specific details of the simulation model will be the topic of Chap. 12.

and maximum AUC values (the lines). The *project network control* (PNC) method shows results for both the X and R control charts, while the *rules of thumb* approach (ROT) shows the three types of straight lines. The methods working with feeding paths—the *feeding path control* (FPC) and *subnetwork control* (SNC) methods—show results for a different number of control points. This number depends on the structure of the project network and on the number of feeding paths in each network. The *longest path method* (LPC) only displays one value, since it only uses one control point to monitor the project performance.

The graph clearly shows that the most basic version of statistical project control using only one project network control point (PNC) outperforms the rules of thumb (ROT) approach, certainly when the static thresholds are used for ROT. These results are, of course, as expected, as the statistical methods are much more powerful than the simple rules of thumb, but they indicate the relevance of using static simulations to set the correct values for the tolerance limits. The longest path method performs equally good as the best ROT methods and therefore should not be considered a worthy alternative as it takes too many changes along the project progress to continuously update the critical path. An interesting observation is that the subnetwork control (SNC) method performs better than the feeding path control approach (FPC) that uses up to 10 different control points (charts). These results indicate that setting a control chart on each linear path entering the critical path is not only too time-consuming (too many control points to monitor), but also results in a weaker performance, and the FPC method should therefore not be taken into account. The SNC method follows a similar logic as the FPC method but uses less control points (up to 5) and performs the best of all methods. This result indicates that the use of statistical control points with a relatively small number of control charts can significantly increase the quality of the warning signals generated during project progress. I therefore advise every researcher to investigate this SNC method in order to further improve the quality of the statistical project control by adding extra features and a richer data analysis.

5.4 Mission Accomplished

As I noted at the beginning of this chapter, this chapter builds the bridge between the first two missions of academic research. More specifically, it was argued that the ROT approach belongs to Mission #1 (*understanding*), in which the researchers examine the existing approach (based on rules of thumb) and validate it with computer experiments. The four other methods went a step further and used statistical methods that are not yet (much) used in practice, and these methods were therefore classified under Mission #2 (*wisdom*). The ultimate goal of academic research is, of course, to convince practitioners to start using these new (statistical) methods, as the experiments have shown that they work a lot better. That is why Mission #2 is also defined as making small, sometimes even marginal improvements, so that the step towards practical use is not very high.

But maybe I am just naive, and this step might never be taken. The results have shown that the SNC is the best method, which is not very difficult and only requires a maximum of 5 control charts. Why would not everyone start using this method? As a professional, it is probably much more relevant to give up some of the excellent performance for SNC and go for the second best option (LPC) to take advantage of the convenience of using only one control chart. In addition, the ROT method, the easiest method of all, does not perform too badly either and requires no statistical simulation at all. Perhaps this is the method that professionals will use now and forever, and the statistical extensions of project control may quietly disappear and never be used in professional settings.

This is a possibility, but I am very hopeful that things will change. It may all be slow, but it will happen some day. I have already mentioned in Chap. 4 that the *earned schedule method* was also accepted by almost no one in the beginning, but that gradually, partly thanks to the academic research, opinions were adjusted. Today, almost no one doubts that this method works much better for time management than the classic EVM method. Above all, I think that an academic researcher should never give up hope that research results will one day find their way into practice and researchers should therefore do everything they can to continue convincing the world of the power of statistical methods over intuition. Perhaps the step towards statistical project control is too big, and we should all look for intermediate forms between the two extreme missions (*should I call it Mission #1.5 then?)*, as is suggested in Fig. 5.9. In Chap. 8, I will look for such an intermediate form to facilitate access from research to practice. Nevertheless, I continue to defend Mission #2 as a fully fledged and important task of academic research. I would even like to go a step further and argue that sometimes it is not even necessary to try to convince the real world. Sometimes research can be an end in itself, in which the researcher goes completely wild in the search for the limits of our knowledge.

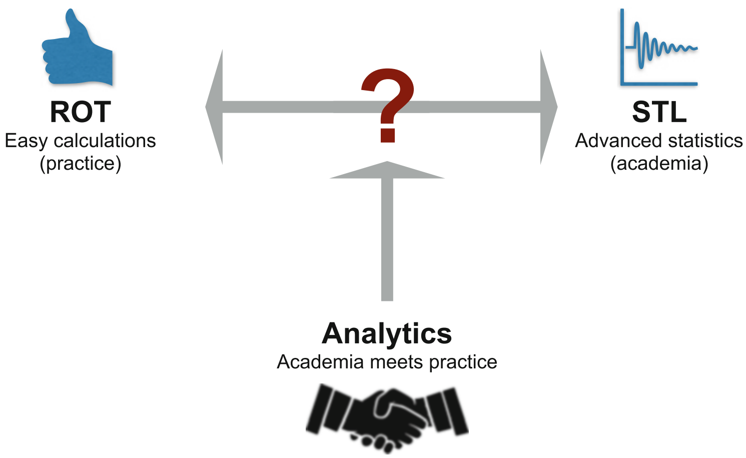


Fig. 5.9 Mission #1.5 (*academia meets practice*)

This quest can be challenging, sometimes leading to strange and incomprehensible results, and is presented as the third mission of academic research in the next chapter. Fasten your seatbelt … it will be a challenging ride!

References

Colin, J., & Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega The International Journal of Management Science, 49*, 107–122.

Colin, J., & Vanhoucke, M. (2015). A comparison of the performance of various project control methods using earned value management systems. *Expert Systems with Applications, 42*, 3159–3175.

Goldratt, E. (1997). *Critical chain*. Great Barrington, MA: North River Press.

Lipke, W. (2012). Speculations on project duration forecasting. *The Measurable News, 3*, 3–7.

Taleb, N. N. (2012). *Antifragile*. New York: Random House.

Chapter 6
Learning

In the previous two chapters, the concepts *understanding* and *wisdom* were used to refer to the two research missions to investigate *existing* and *new* methodologies for managing projects, respectively. I have shown in the previous chapter how the currently existing and generally accepted EVM methods (*understanding*) have been extended to statistical methods (*wisdom*) that are—despite their simplicity—not (yet) widely used in professional environments. As a matter of fact, the five statistical control methods discussed in the previous chapter are not very easy and straightforward to implement in the current PM practices, and I have not met many project managers that were as enthusiastic as I am to use them in their project management systems. Nevertheless, despite this lack of implementation, the underlying idea of these statistical methods is not very complex and not new at all, since it is merely an extension of a generally accepted data-driven methodology used in quality management known as *Statistical Process Control*. Statistical process control is a method of quality control that employs statistical methods to monitor and control a process and has been used in many companies since its introduction by Walter Shewhart at Bell Laboratories in the early 1920s and later popularised by William Edwards Deming, first in the Japanese industry, and later all over the world. Statistical process control is appropriate to support any repetitive process, and as a project does not follow that definition (but is instead assumed to be a unique one-time endeavour), not much effort has been done to translate the statistical *process* control methodology into a statistical *project* control methodology. This translation lies in the core of research Mission #2 of the previous chapter. It aims at extending the current methodologies with new proven principles borrowed from other environments (e.g., manufacturing) and applied to another environment (project management) where it has not been used so far. While these methods are not yet widely used, it is—I hope and believe—only a matter of time before practical use cases will show their relevance in business as some use cases already exist.

Despite the noble second research mission to develop new and not-so-complex methodologies that can be used by professionals without much additional effort,

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_6

academic research should strive for more. Academic research should not only provide new insights and methodologies ready for immediate use, but should also aim at shifting boundaries, explore untrodden territories, and come up with totally new ideas, without thinking too much about the practicalities of implementation or the relevance for business. Instead, academic research should be defined, at some times, as the formalised search for new knowledge for the sake of the knowledge. It is the *freedom to explore* and *search to the unknown*, and it is the primary reason why I have chosen to stay in academia for the rest of my life. This exploration of new ideas is exactly the goal of the Mission #3 of academic research, which is referred to in this book as *learning*. While some people might say that such a type of research is totally irrelevant and a waste of time (and money), others would refer to the wise words of the African–American novelist Zora Neale Hurston who stated the following:

> Research is formalised curiosity.
> It is poking and prying with a purpose.

This chapter extends the project management and control methods to advanced *machine learning* techniques to push boundaries and shift our knowledge, without aiming at reaching immediate practical relevance. The use of machine learning methods in computer science is to create a technology that allows computers and machines to function in an intelligent manner. This chapter applies these advanced statistical learning techniques to monitor and control projects in order to test whether they could potentially be useful in a project management environment. Of course, I fully realise that not many project managers will be able, or willing, to use such advanced methodologies in their projects, but that does not matter, and it does not make this research less interesting or exciting. The previously mentioned quality management guru William Edwards Deming also noted the following:

> Learning is not compulsory... neither is survival.

In this chapter, I will use the words *artificial intelligence* and *machine learning* to refer to advanced statistical techniques to access and process large amounts of data used by fast and efficient computers to improve the quality of the project control decision making process. To that respect, this section is not as exciting as the title suggests and I may disappoint my readers. It is not a chapter about robots with mechanic brains learning to imitate our behaviour, nor is it about science fiction stories as often told in artificial intelligence stories. Instead, it is a chapter about processing an overwhelming amount of project data using advanced analytical methods to improve our knowledge and bring it to new levels. In the next sections, these techniques will be used for the three components of the dynamic scheduling framework (*schedule*, *risk*, and *control*). Welcome to the power of data.

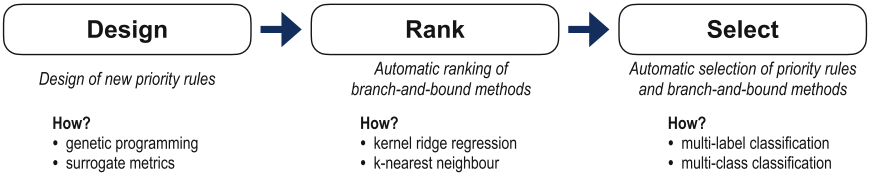


Fig. 6.1 Summary of machine learning methods for the RCPSP

6.1 Schedule

Since *baseline scheduling* is the point of reference for risk analysis and project control, it might seem obvious to use machine learning algorithms for these challenging scheduling problems first. However, the scheduling problems are so challenging that most researchers developed an impressive amount of algorithms to solve these scheduling problems and no one really cared about extending them to more advanced machine learning methods. However, this changed when two international PhD students, Weikang Guo and Jingyu Liu from China, joined my team and expressed their desire to apply learning algorithms to my favourite research theme. Figure 6.1 summarises the work on using machine learning algorithms for the construction of a baseline schedule for projects with limited resources. We worked in three phases, each of them resulted in a publication in a peer-reviewed journal, and each phase will be briefly discussed along the following paragraphs.

Design: Genetic Programming Genetic programming (GP) is a powerful and data-intensive method that can be used for the automated design of priority rule heuristics for the *resource-constrained project scheduling problem* (RCPSP). The RCPSP is a scheduling problem that is notoriously hard to solve (NP hard), and many researchers (and software tools) fall back on simple yet efficient priority rules to schedule these projects without aiming to find an optimal solution[1] (which consists of a project schedule without resource over-allocations and a minimal total project duration). In the academic literature, the potential of applying a so-called *genetic programming hyper-heuristic* to design priority rules has only been scarcely explored in the domain of project scheduling, and there is plenty of room to improve its efficiency and performance.

The term *hyper-heuristic* is used as a high-level approach that can select or generate low-level heuristic algorithms. The *heuristic selection* chooses or selects existing heuristics from the academic literature, and then the *heuristic generation* module generates new and better heuristics from the components of existing ones. The *genetic programming* is widely used as a heuristic generation methodology, and in recent years, this technology has attracted the attention of many researchers in the field of operations research. Instead of relying on human experts, researchers

[1] I have briefly described the various solution methods to solve the RCPSP in Sect. 3.4 of this book.

can now resort to genetic programming to automate the process of heuristic design and discover efficient priority rules that may not be easy to construct manually. The use of priority rules fits perfectly in the approach, as there exist plenty of simple and effective rules to solve the RCPSP, and they can therefore be used to design new and better ones to solve this challenging scheduling problem. Jingyu Liu, one of my Chinese PhD students, accepted this challenge and wanted to design new improved priority rules to solve the RCPSP for large projects up to thousands of activities. For such projects, none of the advanced solution methods (such as meta-heuristics or exact branch-and-bound procedures) work very well, and so the priority rules are most likely the only algorithms to schedule these projects in a *fast* and *relatively good* way. Jingyu is an enthusiastic and hard-working researcher, but often underestimates how much computer effort is necessary to train and test new research ideas. It is known that GP requires intensive computing effort, carefully selected training and test data, and appropriate assessment criteria, and although he initially thought that this was all readily available, it took him quite some time to carry out all his experiments before his paper could be published.

After struggles with an overwhelming amount of project data, limited computer memory, and too many computational experiments, he eventually published his first paper (Luo et al., 2022). He used 4320 project instances from well-known existing project datasets (which will be discussed further in Chap. 11) consisting mostly of projects with 30, 60, 90, and 120 activities, and the biggest ones up to 300 activities. He tested the new GP-based priority rules on a test set of projects and concluded that the newly designed priority rules outperformed the existing ones. Despite this good news, these new rules will never be used for such small projects, which is why he also trained and tested his GP approach on newly generated datasets with projects with more than 1000 activities. The GP-designed rules performed significantly better than the best traditional priority rules for these large projects, which clearly shows the benefits of using the data-intensive GP heuristic to improve the schedule quality of real-sized project instances. Happy with his first publication and with a renewed motivation due to the interest he got from some Belgian companies that wanted to schedule huge projects with up to 5000 projects, he extended his GP algorithm to a so-called surrogate-GP method, which he summarised in Luo et al. (2023).

Rank: Regression Models Weikang Guo, another Chinese PhD student working in the same office as Jingyu, was more interested in finding (near-)optimal solutions for small projects instead of finding relatively good solutions for big projects. Rather than replacing the fast priority rules from the literature by better ones, she decided to rely on the exact and often complex *branch-and-bound* (BnB) methods available in the literature that can—for relatively small projects—solve the RCPSP to optimality.

The development of branch-and-bound procedures for solving the RCPSP dates back to my own PhD period, in which working on exact algorithms was considered as an honour. The BnB method is an algorithmic approach that consists of a systematic enumeration of candidate solutions by exploring the search space in a systematic way. The algorithm constructs a tree, which initially starts with the full set of all possible solutions at its root node. It then explores branches of this tree, which represent subsets of the solution set, and each branch gets an estimated value for the optimal solution of the problem in the form of a lower bound. When the node in the tree has a bound value lower than any feasible schedule found so far, this tree must be branched further. However, the node is discarded if it cannot produce a better solution than the best one found so far. Any branch-and-bound algorithm consists of a number of different components that can be implemented in different ways. The number of possible settings for each component is given between brackets in the following description:

- Search strategy (2): The search in the branch-and-bound tree can be explored from two fundamentally different points of view. In the so-called *upper bound strategy* (U), the search aims at improving the solution of the schedule until no better solution can be found. It starts with a feasible schedule with a certain value for the total project duration and gradually aims at improving it (reducing the duration) until the best one is found. The *lower bound strategy* (L) works exactly the opposite way around and starts with a lower bound value on the total project duration (which does not represent a feasible schedule) and then gradually aims at increasing this lower bound value until it can be proven that it corresponds to a feasible project schedule with a minimum project duration.
- Branching scheme (3): This scheme determines the way the nodes are constructed at each level of the tree, i.e., it determines how the solution space is split into different parts at each level of the tree. Three well-known branching schemes are tested, known as the *serial*, *parallel*, and *activity start time* branching rules.
- Branching order (2): Once the branches are constructed at a new level of the tree, the algorithm has to decide the order in which the nodes are selected for further exploration. These nodes can be ordered according to the best lower bound value (which gives an estimate of the project duration of a feasible solution) or according to the ID of the activity.
- Lower bound (4): The algorithm depends on efficient estimation of the lower bound calculations, and if no bounds are available, the algorithm degenerates to an exhaustive search of the solution space that will—for large projects—take forever. The study has used 4 composite lower bounds as an assembly of 13 lower bounds from the literature.

Any BnB algorithm must select one setting for each of the four components, and so it is not very difficult to see that in total, $2 \times 3 \times 2 \times 4 = 48$ different configurations are possible, which results in 48 slightly different BnB algorithms.

My Portuguese friend José Coelho has programmed these BnB algorithms in 2018 by implementing all possible configurations of the four components in a so-called *composite lower bound branch-and-bound procedure*, and this allowed Weikang to use this algorithm in her study (the development of this 48-component algorithm is discussed in detail in Sect. 11.5 of Chap. 11). However, not all of these 48 configurations had been presented in the academic literature yet, and between 1974 and 2000, only 12 high-performing BnB procedures were proposed to solve the RCPSP.[2] Each algorithm makes use of only one or two settings for the four components described earlier, and in total, we could classify these 12 algorithms in 9 different configurations, which cover less than 20% (9/48) of all possible configurations. Weikang used the solutions obtained from the 48 BnB algorithms to investigate which configuration performed best. To that purpose, she made use of two regression models to predict the ranking of the best performing configurations based on easy-to-calculate features of the project. More precisely, for each individual project instance, the regression methods must rank the performance of each of the 48 configurations based on project characteristics such as the number of activities, the network structure, or the use of resources by the project activities. Consequently, this ranking is very project-specific and can differ from one project to the other. Her ultimate goal was to automatically rank the configurations and then select the best ones to solve the RCPSP hoping that the solution would be better than the solutions found by the 12 existing algorithms in the literature.

A detailed description of these ranking methods would lead me too far, and the readers are referred to her published paper (Guo et al., 2023b) for more details. The two regression models used are the *kernel ridge regression* (kRR) and *k-nearest neighbour* (kNN) algorithms, which are well-known methods in the machine learning literature. She performed a computational experiment on a set of 3,143 projects (80% for training and 20% for testing) under different stop criteria of 1 s, 60 s, and 1 h, which took her months of hard work and a lot of CPU time. Results were impressive, though, showing that the regression models could predict the ranking of configurations pretty well. In one of her experiments, 490 hard instances were selected and solved by the two regression models (kNN and kRR) and the 12 existing branch-and-bound procedures from the literature (using only 9 of the 48 configurations). Figure 6.2 shows the results of the two regression models for the 12 branch-and-bound algorithms (no author names are mentioned, just the year of publication). Each method is truncated after 1 h of processing time, and the best project duration is reported. For the two machine learning methods, the 48 configurations were automatically ranked, and then the first 10 configurations were sequentially called (running for 6 min) to report the best found solution after 1 h. The existing BnB procedures consist of only one specific configuration, which was run for 1 h to report their best solution. The figure clearly shows that the two regression models outperform the existing BnB procedures, which demonstrates that ranking the different configurations from the literature into one integrated prediction model is worth the effort. Despite the complexity of this ranking method, it nevertheless makes the use of machine learning algorithms an interesting future research avenue in the context of project scheduling.

Select: Classification Models Academic research can be intense, stimulating, and rewarding. It requires an unsatisfying hunger for knowledge and a commitment to

[2] After the year 2000, meta-heuristic solution methods became so popular and powerful that almost no one wanted to work on exact algorithms anymore.

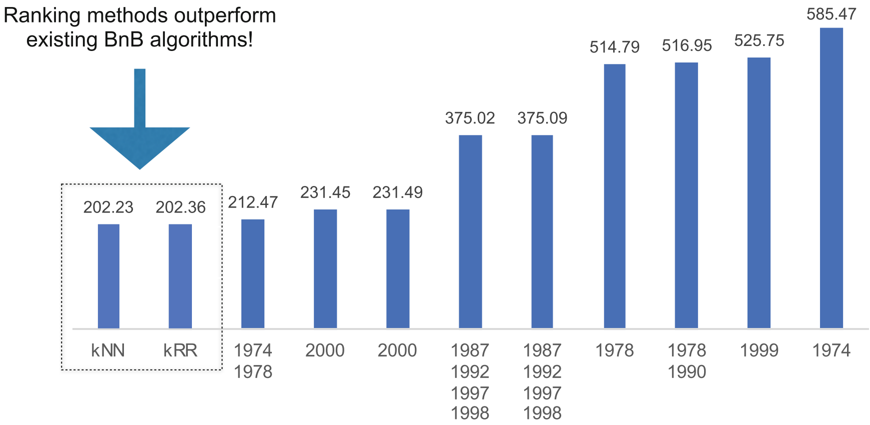


Fig. 6.2 Performance of two learning models and nine existing BnB algorithms

exploring the limits of our understanding in search of impossible improvements. Weikang has that spirit and did not stop after the impressive results of her previous study. She wanted more, and replaced, in a new study, the two ranking algorithms by a set of machine learning algorithms that automatically select the best performing configuration(s) instead of ranking them in decreasing order of performance. The rationale behind this new approach is that a prediction of the relative ranking between algorithms (configurations) does not really matter and an automatic project-specific prediction that determines which configuration(s) could provide the best solution is sufficient. Such prediction can be provided by the classification models used in machine learning.

Classification is a supervised learning task in which the computer learns from the labelled training data and uses this learned information to classify a new unseen object or sample of objects. With a given dataset, attributes that are used to describe each sample are referred to as *features*, and the category identifiers are referred to as *labels*. The features and labels are often used as the inputs and outputs of the model, respectively. The goal of the classification is to learn a function (or mapping) from inputs to outputs. The various machine learning techniques that implement the function are known as *classifiers*. Based on the number of labels to be predicted for each sample, the classification can be categorised into single-label classification methods and multi-label classification methods. The *single-label classification* methods assign an object to exactly one class when there are two or more classes available, while *multi-label classification* deals with objects having multiple class labels simultaneously. In the context of project scheduling, an object will be represented by a project instance, a feature is represented by well-known project indicators (measuring the network structure and the resource use), and the labels are given by the solution of the 48 BnB configurations or by a set of priority rules. Since it is obvious that for each project instance there may be more than one BnB configuration or priority rule that can generate the best possible project duration among all the procedures, the learning task to be addressed is a typical *multi-label classification task*.

Weikang tested the classification methods with both priority rules (Guo et al., 2021) and the 48 branch-and-bound configurations (Guo et al., 2023a) and relied on six machine learning base classifiers (decision trees, random forests, neural networks, support vector machines, k-nearest neighbour search, and naive Bayes). The difference between single-label and multi-label classification is explained along the following lines using *decision trees* as a classifier (but the words "decision tree" can be easily replaced by the other five classifiers without changing the underlying classification model):

- **Multi-label classification with binary classification.** This method transforms the multi-label classification task into a set of single-label models, which can be done in two different ways. In the first model, the *binary relevance* (BR) method is used to decompose the multi-label classification task into multiple binary classification tasks, which means that for each label (priority rule or BnB configuration), a separate classifier (e.g., a *decision tree*) is built. For the BnB procedures, this approach results in exactly 48 decision trees during the training process, and the ultimate desire is that only a small number of these trees results in a positive prediction. If that is the case, each test instance is then solved by only this small number of solution algorithms, i.e., only with the BnB configurations or priority rules that return a positive prediction among the complete set of possible trees. A second method, known as the *classifier chain* method, works in a very similar way but adds correlations between the single-label models to better classify the data and make the different decision trees depending on each other.
- **Single-label classification with multiple classes.** This method transforms the multi-label classification into a single-label classification with multiple classes. More specifically, the multi-label classification task is converted into a single-label multi-class classification task by treating a *set of labels* as a new class. Hence, if multiple labels (e.g., a *set* of priority rules or BnB algorithms) give the best possible makespan for a particular instance, then this instance is labelled with the newly defined class containing all these positive labels such that this instance gets a single newly defined class label. The obvious advantage is that this approach results in only one decision tree, but since the number of created classes during the model training is not known in advance, it can result in one enormous decision tree with many newly created classes. Nevertheless, each instance will make use of the single (huge) tree during the testing phase, and for each new project instance with known feature values, only one leaf node will be selected (which means that only one priority rule or BnB configuration will be used to schedule this project instance).

The computational results showed that all classification models outperform the performance of using any single solution procedure (either priority rule or BnB configuration). Moreover, it was shown that the multi-label method with classifier chain (which incorporated correlation between the single-label models) works better than the multi-label with binary relevance method, which illustrates the importance of adding correlations between the different classifiers (decision trees). However, much to our surprise, the results also showed that this performance is (slightly) worse than the results of the ranking models discussed in the previous study (as shown in Fig. 6.2).

Conclusion The results of the previously discussed machine learning studies illustrate that the machine learning framework can help select algorithms to solve this challenging problem, but also that much more research is necessary to put these insights into practical applications. As a matter of fact, this research should be read with a critical eye, since the algorithms used in this study require not only much implementation effort for incorporating all BnB strategies, but also rely on a learning process that consumes an enormous computational time. Not many professional project managers are ready to take this step to construct their project baseline schedule. However, the lack of immediate practical relevance should not be seen as a weakness of such research, as the summary of these research studies is written in the *learning* chapter that was defined as the third mission of academic research. As a matter of fact, every research study presented in this chapter consists of a search to the unknown, trying to extend knowledge for the sake of the knowledge. The academic freedom to explore gave Weikang the time and opportunity to test advanced machine learning methods without really knowing where she could and would end, and it brought us to new unknown but exciting territories. Nevertheless, her exploration has shown that combining the best components from the literature improves the solution quality for resource-constrained project scheduling problems, and therefore, the study has shown that preference learning, in general, and label ranking predictions, in particular, are interesting research avenues in the context of project scheduling. Since the project *schedule*, the first of three components of the data-driven project management framework, serves as a point of reference for the other two components (*risk* and *control*), it might be worth the effort to rely on the advanced methods, not only in academia, but even—one day in the future—in practical situations.

6.2 Risk

For the use of machine learning methods to analyse project risk, I worked together with Izel Ünsal Altuncan, a PhD student from Turkey living in Brussels (Belgium) with her husband and child. She joined my team with an interest in *risk networks* and expressed her desire to work on project time and cost forecasting based on the papers I discussed in Chap. 4. I explained in that chapter that the research on project *time forecasting* began with a comparison between traditional *Earned Value Management* methods and the new *Earned Schedule* methods, and the performance of these methods to predict the final project duration was measured by their *accuracy* and *reliability*. Predicting the total duration and cost of a project in progress is important,

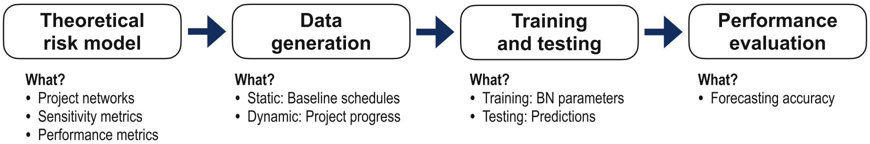


Fig. 6.3 Stepwise approach for project predictions with Bayesian networks

not only for academics, but also for professional project managers. However, none of the studies in academia made the link between *time forecasting* and *schedule risk analysis*, and so that became the focus of the new research study. She investigated the use of *Bayesian networks* (BN) to predict the expected total duration of a project based on known input parameters from the project scheduling and risk analysis literature. A Bayesian risk network model is said to provide a *static prediction* since it aims at forecasting the project's time performance prior to the project start, which is different from the time predictions discussed in Chap. 4 that predict the duration of the project during its progress. As we have already referred to the differences between the static and dynamic phases, the BN method uses both a static simulation (to determine the sensitivity values for the activities) and a dynamic simulation (to simulate the real progress of projects). The design of the study consists of four phases that are summarised in Fig. 6.3 and briefly explained along the following paragraphs (Unsal-Altuncan & Vanhoucke, 2023).

Phase 1. Theoretical Model In this first phase, a theoretical model has to be built in which relationships are established between the characteristics of projects and their possible impact on the quality of forecasts. The creation of such a theoretical model can only happen if there are pre-existing insights into such relationships, and this shows the beauty of academic research as a discipline where one researcher builds on the results of another. Specifically for this project, we could use the results and insights described in Sect. 4.2 of Chap. 4 that showed that the network topology of a project and the sensitivity of its activities have a major impact on the quality of time and cost forecasts. These results were published years before Izel's arrival, but she used them to build a so-called theoretical risk model shown in Fig. 6.4. The circle nodes represent the *risk variables*, while the arcs between these nodes indicate the direction of *causality* between the risk variables. These risk variables can be independent or dependent, and a dependent risk variable is causally preceded by at least one other risk variable in the model. The figure shows that the network structure (NT), time sensitivity (TS), and cost sensitivity (CS) of activities are all independent risk variables, while the time performance (TP) and cost performance (CP) of a project are dependent risk variables. In the cases where risk variables cannot be directly measured, *observable indicators* must be used as measurements, represented by the rectangles. Two risk variables in the model are set as target variables, because they are the variables that the model wants to predict. In the study, both time performance (TP) and cost performance (CP) were set as target variables (represented by the grey nodes) because the research goal

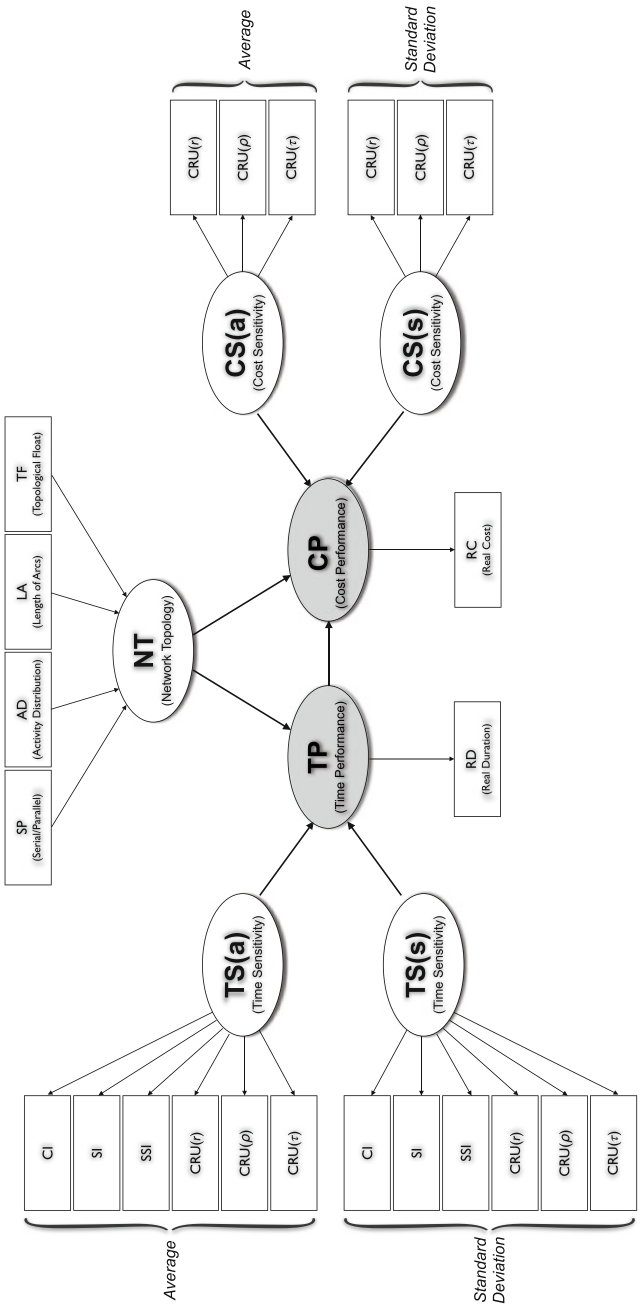


Fig. 6.4 Risk model used for Bayesian network predictions

of this study is to ultimately predict the actual duration (RD) and cost (RC) of the project. This theoretical model gives a clear picture of how the total duration and cost estimates depend on a number of project-specific factors, and the model should now be validated through a data-intensive process using Bayesian network analysis to investigate whether this is indeed the case or not.

Phase 2. Data Generation Since a Bayesian network analysis, like all machine learning methods, requires a lot of data, the second phase consists of generating a large heap of different values for the observable variables (*rectangles*) that define the values of the risk variables (*circles*). These values can be obtained by using a large database of artificial projects and by using static and dynamic simulation runs very similar to the simulation runs discussed earlier in Sect. 5.1. The generation of the data for the three classes of *observable variables* works as follows:

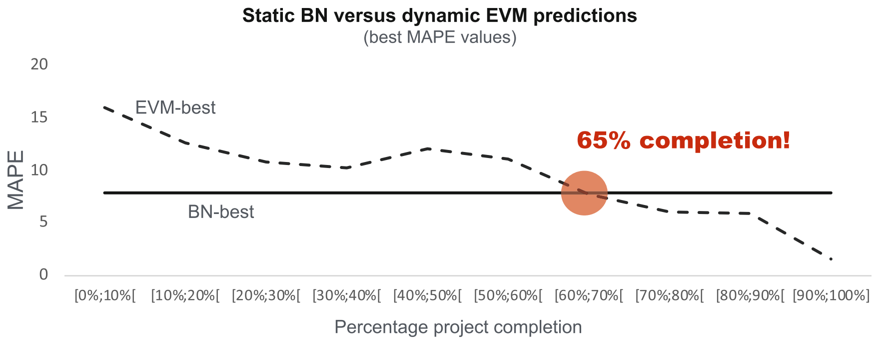
- Network topology (NT). The study makes use of a set of 900 artificial projects generated under a structured design to maximally vary the topology of the networks. The network structured is controlled during the generation by using the four so-called network topology metrics, known as the *serial/parallel* (SP) indicator, the *activity distribution* (AD) indicator, the *length of arcs* (LA) indicator, and the *topological float* (TF) indicator. The generation process of these networks, along with a brief presentation of the four indicators, is the topic of Chap. 11 (Table 11.1).
- Time (TS) and cost (CS) sensitivity: The values of the observable variables for the time/cost sensitivity are obtained by *static simulations* using a *schedule risk analysis*. This methodology is presented earlier in Sect. 3.4 and generates *sensitivity metrics* for the project activities based on Monte Carlo simulations. These metrics measure the sensitivity of variability in the duration and cost of the activities of a project as a percentage between 0% (not sensitive) and 100% (highly sensitive). This sensitivity can be expressed by the *criticality index* (CI), *significance index* (SI), or *schedule sensitivity index* (SSI) to measure *time* sensitivity, and by three versions of the *cruciality index* (CRI) to measure time *and* cost sensitivity. Since these metrics provide a percentage for each activity individually, they should be integrated into one single percentage on the project level. Therefore, the average values of these metrics (TS(a) and CS(a)) as well as their standard deviation (TS(s) and CS(s)) are used as the two observable variables for each metric.
- Time (TP) and cost (CP) performance: The performance of the project is measured by the real duration (TP) and real cost (CP) of the project, and these values are only available when the projects are completed. Since the algorithm makes use of artificial projects, these two values must be determined using *dynamic simulations* to imitate artificial project progress.[3]

[3] As I have already mentioned earlier, the specific design of these dynamic simulations to imitate project progress will be discussed in Chap. 12.

Phase 3. Training and Testing The third phase involves constructing multiple risk models by splitting the huge artificially created dataset from Phase 2 into several training and test sets and applying the BN algorithms to predict the time and cost of these projects. In model training, parameter learning algorithms are used to analyse whether the theoretical model of Phase 1 is consistent with the generated data of Phase 2. This training step results in an appropriate risk model with optimal model parameters. The so-called *k-fold cross-validation* approach is used to arrive at the optimal parameter set, as is common in machine learning, which sequentially divides the database into training and validation sets. This work is largely done as a black box analysis, which is one of the dangers of machine learning research, and the details of the BN algorithm are therefore beyond the scope of this book. In fact, the magic happened by using the package "bn.learn" in R, which—without giving details—works as follows: The parameter set for a Bayesian network is estimated using *maximum likelihood estimation* algorithms that take the form of local probability distributions expressed as Gaussian linear models consisting of regression coefficients. In the first step, the project data are entered into the model and the values for the variables are determined. In the second step, the initially estimated local probability distributions for the remaining variables are updated (re-estimated) using Bayes' theory, based on the fixed values for the evidence variables and the regression coefficients in the parameter set. After these runs, the estimates of the network risk parameters are available, which can be used to predict the time and cost performance of new unseen projects. In our testing phase, the model uses the fitted risk models for predicting the time and cost performance of a large set of projects under widely varying conditions.

Phase 4. Forecasting Output Finally, in the fourth phase, the prediction accuracy is evaluated for the time and cost performance of the test predictions to verify whether the BN model does provide improvements over the traditional forecasting methods. This prediction accuracy is measured using the same criteria that we used in Sect. 4.2 (*mean absolute percentage error*) (MAPE) and is therefore not discussed further here.

Figure 6.5 displays a summary of the results of the study using two time prediction models on a set of 19 empirical projects. The graphs show a comparison of the MAPE values (y-axis) for different percentages of project completion (x-axis) ranging from 0% to 100%. The first forecasting model makes use of the static predictions with a Bayesian network algorithm (BN). The results are shown by a horizontal line since the forecast is made at the start of the project (and is never updated). The second method makes use of the nine dynamic EVM forecasts presented earlier in Chap. 4 that are updated along each growing percentage of project completion. The top graph shows the results of both forecasting models using the best obtained prediction, while the bottom graph displays average results for each model (using different versions). The graphs show the impressive behaviour of the risk models with BN and indicate that machine learning and data training using the model of Fig. 6.4 can provide better forecasts for most of the project phases. When average results are used, the nine EVM methods cannot provide better



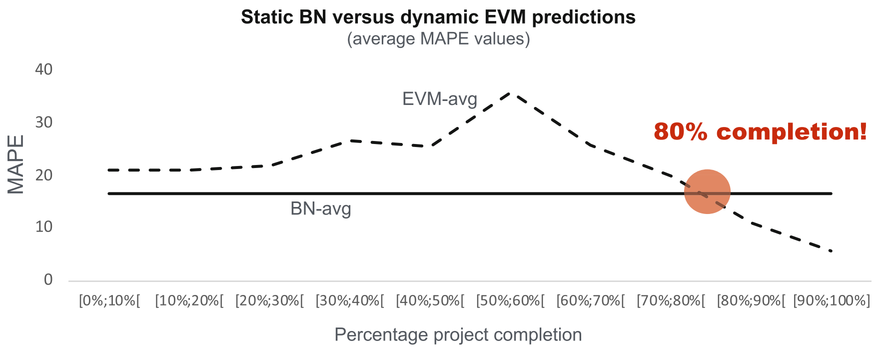


Fig. 6.5 Time accuracy of Bayesian networks and EVM predictions

predictions, on average, for the first 80% of the project completion. When the best performing EVM method is used (which is, as explained in Chap. 4, the earned schedule method), then the best performing risk model (BN) still outperforms the EVM forecasts in the first 65% of the project. These results are a clear illustration of the power of machine learning and the importance of using network topology (NT) and risk sensitivity (TS and CS) metrics to predict a project's time performance (TP). It is also an experimental proof that a composite forecasting approach (combining static risk models with dynamic progress data) can significantly improve the overall quality of project predictions, and future research should focus in developing and improving such composite methods.

6.3 Control

Since controlling projects constitutes the main theme of this book, it would be ridiculous not to test the machine learning methods on the *project control research* studies of the previous chapter that all rely on the *Earned Value Management* methodology. Much of the experimental work is done in collaboration with Mathieu Wauters, then a PhD student at my research group, and now a consultant in one of the biggest consultancy companies in Belgium. Our collaboration has resulted in three scientific publications on the use of different machine learning methods to control projects, and a brief summary is provided in this section. Figure 6.6 summarises the computational design used to test various machine learning project control methods and shows that it consists of a three-phase approach very similar to the project control study of the previous chapter. The inputs (Phase 1. *Baseline schedule*) and outputs (Phase 3. *Dynamic simulations*) are almost identical, but the second phase is completely different and relies on advanced machine learning methods trained on a large set of artificial project data. The aim of the research experiments was not to test control charts (as was the case in the study of Chap. 4), but to improve predictions of the total duration of a project. Such a study was already presented in Chap. 4 in which three basic methods of EVM were used. The current study uses the same methods but builds on them by applying machine learning algorithms to the data to significantly improve the accuracy of the predictions. It goes without saying that such algorithms probably will not be used by many companies, but it was nevertheless interesting to see to what extent improvements were possible at all. The most important components of Fig. 6.6 will briefly be discussed along the following paragraphs, followed by a brief description of the main results.

Phase 1. Input Data The input data for the machine learning model of Phase 2 consist of a large set of artificial projects and probability distributions to model the uncertainty of the activities in the baseline schedule. These inputs will be used for the three-step approach shown in Phase 2 of Fig. 6.6.

Phase 2. Machine Learning Model The machine learning model consists of three steps as discussed along the following lines:

- *Generating project data.* Machine learning is a data-intense methodology that requires literally gigabytes of data for training and testing. The data consist of *static project data* as a large set of artificial project networks with activity time and cost estimates. The *dynamic project data* contain project progress data at regular time intervals to monitor the project performance. These project progress data can be obtained by the so-called *static simulation runs* that I discussed previously using the parameters for the predefined distributions. Consequently, for each project, literally hundreds of project tracking periods must be simulated under different settings of these parameters, each of them providing a wealth of data in the form of EVM performance metrics such as the schedule performance index (SPI) and cost performance index (CPI). These metrics are now called *attributes* to be in line with the terminology used in machine learning and will be used to construct control thresholds (similar to the ones of the previous chapter) and provide time and cost predictions. As should be clear by now, the generation of static and dynamic project data is key in most of the project control studies of this book, and special attention to these data generation processes will be given in Part IV of this book.

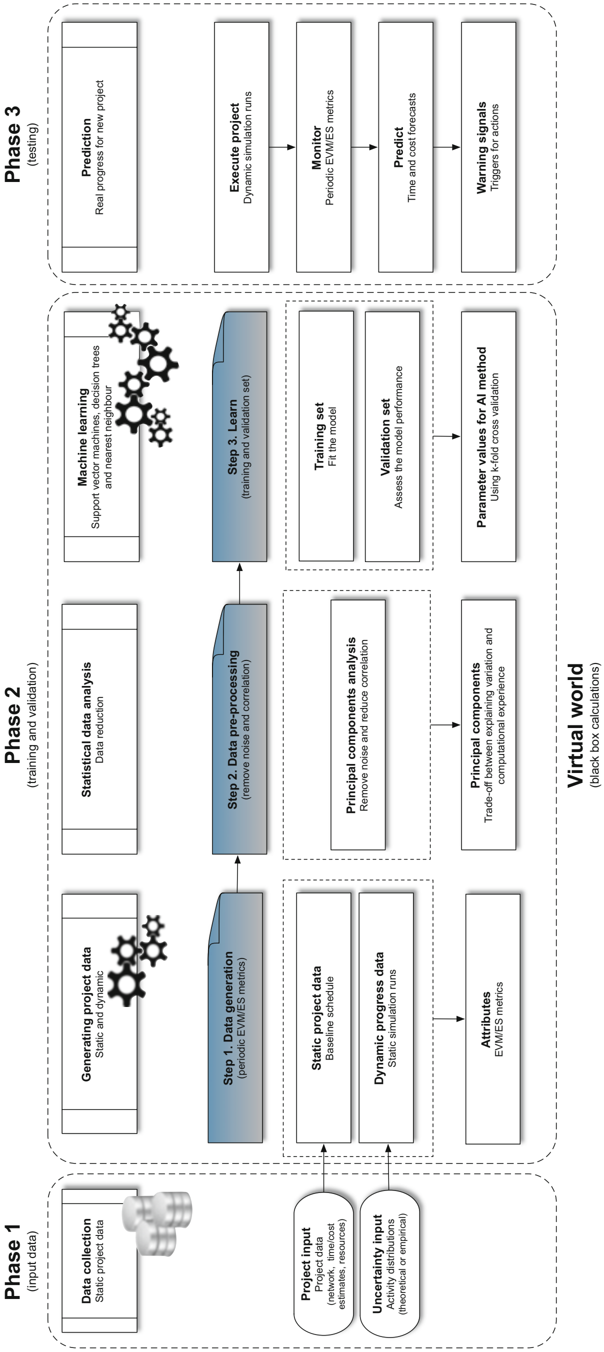


Fig. 6.6 Machine learning model for project time forecasting

- *Statistical data analysis.* Before the created data can be used for machine training, they should first be pre-processed. Typically, the overwhelming amount of data is first processed using different statistical techniques that aim at distinguishing between explanatory variation and noise. Using a *principal components analysis*, a traditional statistical technique to convert a set of data of possibly correlated variables into a set of values of linearly uncorrelated variables, the overwhelming amount of data is pre-processed and made ready for further analysis by the machine learning algorithms.
- *Machine learning.* The training of the data with various machine learning algorithms aims at finding relations in the data to improve the overall forecasting accuracy of EVM methods. Traditional, the discovery of such relations is done by dividing the generated data into two sets. In the *training set*, potential predictive relations are detected and used to fit the parameters of the machine learning algorithms and find their best possible values. In a *validation set*, it is tested whether these discovered relations are robust enough, and whether the quality of the forecasts, based on the found relations, is general and widely applicable under different settings (*parameter tuning*). Mathieu has tested different machine learning methodologies, including *Support Vector Machines* (Wauters & Vanhoucke, 2014), *Decision Trees*, and *Random Forests* (Wauters & Vanhoucke, 2016) as well as the *Nearest Neighbour* method (Wauters & Vanhoucke, 2017) for predicting the final expected duration and cost of a project in progress.

Phase 3. Predict the Project Duration (*Dynamic Simulations*) Based on the trained relationships, the method should be tested on new unseen projects to investigate whether the models can indeed improve the accuracy of project predictions. As I mentioned earlier, the third phase is identical to Phase 3 of the previous chapter and simulates the progress of a new unseen test project using the *dynamic simulation runs* to predict its time and cost. The difference between the predictions and the real (i.e., simulated) time and cost of the project is measured by the *mean absolute percentage error*, which also has been presented earlier.

The computational experiments in this study have shown that the machine learning methods could improve the forecasting accuracy of the traditional EVM methods discussed in Chap. 4, which is—given the intense use of data—not a surprise. Nevertheless, the study illustrates the power of machine learning and its ability to improve the current state-of-the-art knowledge of project time and cost forecasting. Despite the promising results of these studies, it was (and still is) not very clear why these methods performed so well (high accuracy), and no one of my team could really explain the differences in performance between the different machine learning techniques. However, one technique got my special attention, and I became increasingly interested in the *nearest neighbour method*. This method was not the best performing machine learning method in the list, but it is undoubtedly the easiest one as it "simply" compares the current status of a test project in progress with the huge amount of generated data to find the closest neighbours (i.e., the set of data points that most closely resemble the current status of the project). It then makes use of the prediction in the generated database of these neighbour projects to forecast the time and cost of the current project in progress, hoping that the accuracy will be high. The experimental results of the study showed that such an easy approach can lead to high-quality predictions, which has motivated other members of my team to start a new research study using a similar idea. More precisely, the well-known and widely used *reference class forecasting* approach relies on past project data and aims at finding similarities between a project in progress and these historical finished projects and then uses these similarities to predict the future of the current project. This interesting technique is discussed later in Chap. 9, and the current machine learning studies show that, although Mission #3 (*learning = freedom to explore*) does not always directly lead to easy results or methods used by professionals, it can nevertheless be inspiring to start new research studies that eventually could and will result in practical methods and business relevance.

6.4 Torture

A chapter about research in machine learning is not complete without a critical note on the danger of analysing data with learning algorithms. The use of machine learning algorithms in academic research has grown in the last decades and now finds applications in almost every management discipline. Implementing these advanced methodologies has become very easy these days, with free access to tools such as R or Python to give the user with some simple coding skills access to the most advanced methodologies. This simplicity comes at a price. It is the price of *not knowing what you are doing*. As a researcher, you are supposed to guide the readers of your paper in the complex world of machine learning algorithms and create a better understanding of your topic under study. However, with these advanced algorithms and easy access to tools, the researcher quickly enters a very dangerous territory in which one hardly understands what is going on behind the learning machine. Often times, defining the right inputs and interpreting the outputs are much more complex than the analysis of the data itself, reducing the learning process to a black box process that may result in better results but few new insights. The British economist Ronald Harry Coase already referred to this danger in the following quote:

> If you torture the data long enough, it will confess.

I am not advocating that the application of machine learning algorithms should be discontinued, but I do think that researchers should be careful and always put the creation of new knowledge and insights first. Despite this potential danger, I strongly believe that the use of these black box algorithms offers many opportunities for academic researchers to provide more and better insights into the management of projects that can then lead, directly or indirectly, to applications and methods relevant to the real world. I have experienced the challenges that this kind of research entails, as well as the numerous opportunities it offers, when I created the new platform *P2 Engine* for my research group. I like to share in the following paragraphs this experience with my readers to conclude this chapter.

When I was working on our commercial project scheduling software tool *ProTrack* with my friend Tom Van Acker (cf. Chap. 2), we saw some potential for using the algorithms of this software tool for academic research. While Tom was coding the graphical user interface, I was coding various algorithms for scheduling projects, analysing their risk, and controlling their performance (in C++). It was a time where my research group was growing with young and enthusiastic PhD students, all of them very eager to test these algorithms on artificial project data in order to find improvements. Machine learning was becoming increasingly popular at these times, and so the algorithms could be interesting to generate a lot of static and dynamic data without re-coding them for every new study. For this reason, Tom and I decided to use the algorithms of *ProTrack* for research purposes, and we assembled all the coded algorithms in a new tool, tailor-made for researchers, and called it *P2 Engine*[4] (www.p2engine.com, Vanhoucke (2014)). The tool is a command line utility tool based on the LUA scripting language (www.lua.org) to generate and analyse gigabytes of project data with a simple coding process to have access to the wide range of algorithms of *ProTrack* without using its graphical user interface. The tool is made in such a way that it is (relatively) easy to extend the algorithms with other features without much need for additional coding. Moreover, the tool is a platform-independent software tool that runs on Windows, Mac as well as on Linux, and most importantly, it also runs on the supercomputer of the Flemish Supercomputer Centre (VSC). In the fall of 2012, Ghent University announced the introduction of the first Flemish supercomputer as a High-Performance Computing (HPC) system with a price tag of about €4.2 million. It was, based on the ranking at number 118 in the Top500 list of June 2012, also the biggest supercomputer in Belgium. With a powerful supercomputer, a growing OR&S project database, many new and fast algorithms, and a young team of PhD students, it would have been stupid not to use our ProTrack algorithms for academic research purposes.

By using simple LUA scripts, our researchers could now generate a lot of project data and solve difficult and critical dynamic project scheduling optimisation problems using the fast and intelligent algorithms of *P2 Engine*. Since then, it has been used by many students who had access to the supercomputer for constructing project baseline schedules, for performing schedule risk analyses as well as for creating dynamic project progress data to test and validate new research ideas. Most of the research presented in this book has been experimentally tested using the *P2 Engine* tool, and I am still very proud of what Tom and I have done for my research team. More than 10 years after its introduction, *P2 Engine* is still widely used by some of my research members. I now realise that I would never have come up with the idea of making this tool if I had not experienced that most machine learning algorithms require so much data that it would be almost impossible for my PhD students to program all the algorithms themselves to generate these data.

[4] P2, or two times P, is used to refer to **P**roject **P**lanning.

Machine learning research has been an interesting journey so far, and I am sure my research group is not at the end of this journey yet. With more project data, powerful computers, and advanced machine learning methods, it will be relatively easy to test any new research idea, and I hope that some of these ideas will be challenging enough to start a new research project. Nevertheless, I will continue to be careful that we do not go in a very extreme direction in which we just rely on these advanced algorithms to yield new research results without providing new insights. I have therefore solemnly promised myself and my team members that I will repeatedly ask myself whether the research has, or can have, enough practical relevance. I have somewhat arbitrarily classified the research of the three chapters of Part II (*what academics do*) into separate missions of academic research, but the translation of this research into practice remains a necessary challenge. In the next Part III (*what professionals want*), I will describe in detail which studies have ultimately led to insights that are not only relevant for scientists, but also for professional project managers in their daily practice of managing projects.

References

Guo, W., Vanhoucke, M., & Coelho, J. (2023a). Detecting the best performing branch-and-bound procedures for the resource-constrained project scheduling problem via machine learning. *Working Paper* (under submission).

Guo, W., Vanhoucke, M., & Coelho, J. (2023b). A prediction model for ranking branch-and-bound procedures for the resource-constrained project scheduling problem. *European Journal of Operational Research, 306*, 579–595.

Guo, W., Vanhoucke, M., Coelho, J., & Luo, J. (2021). Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem. *Expert Systems with Applications, 167*, 114116.

Luo, J., Vanhoucke, M., & Coelho, J. (2023). Automated design of priority rules for resource-constrained project scheduling problem using surrogate-assisted genetic programming. Swarm and Evolutionary Computation, forthcoming (doi: 10.1016/j.swevo.2023.101339)

Luo, J., Vanhoucke, M., Coelho, J., & Guo, W. (2022). An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem. *Expert Systems with Applications, 198*, 116753.

Unsal-Altuncan, I., & Vanhoucke, M. (2023). A hybrid forecasting model to predict the duration and cost performance of projects with Bayesian networks. *Working Paper* (under submission).

Vanhoucke, M. (2014). *Integrated Project Management and Control: First comes the theory, then the practice*. Management for Professionals. Springer.

Wauters, M., & Vanhoucke, M. (2014). Support vector machine regression for project control forecasting. *Automation in Construction, 47*, 92–106.

Wauters, M., & Vanhoucke, M. (2016). A comparative study of artificial intelligence methods for project duration forecasting. *Expert Systems with Applications, 46*, 249–261.

Wauters, M., & Vanhoucke, M. (2017). A nearest neighbour extension to project duration forecasting with artificial intelligence. *European Journal of Operational Research, 259*, 1097–1111.

Part III
What Professionals Want

Academics should be encouraged to conduct research
relevant to the decisions faced by managers.

In Part II of this book, I outlined the three missions of academic research and illustrated them by three different studies in project control. In the first study, a comparison of three different forecasting methods to predict the project duration was proposed, and the *earned schedule method* was selected as the most accurate method. The insights of this study were used to develop five project control methods using statistical tolerance limits, and two methods were said to outperform the other three based on two statistical criteria for hypothesis testing. Finally, it was shown that improvements are still possible by adding advanced machine learning techniques, leading to very advanced, but nevertheless very good methodologies to schedule projects, analyse their risk, and monitor and control their performance when they are in progress. This continuous and never-ending search for improvements lies in the nature of academic research and makes me think of a song by Daft Punk:

Harder, Better, Faster, Stronger.

Indeed, contrary to popular belief, things can and should move quickly for researchers. Better results, faster algorithms, and stronger methodologies lead to publications, new insights, and ideas for further research.

Professionals do not always need these faster and stronger tools, are often unimpressed by the marginal improvements made in academia, and are usually happy with the tools they have. Every improvement comes with some costs (e.g., for implementing the new methodology), and it is often much easier (cheaper) to stick with the current methodologies, even if they do not work as well as the new alternative methodologies. However, even in the most impractical research, there are often hidden gems that, with some effort and persistence, can lead to practical improvements without entailing enormous cost or loss of time. The three chapters of Part III elaborate on a number of research projects that were based on theoretical academic research, but ultimately resulted in a number of insights from which professionals can draw conclusions to improve their project management approach.

In the upcoming chapters, I will discuss the project management research themes from a practical point of view and put the academic improvements into the right perspective. Chapter 7 discusses a new project control concept called *control efficiency* that aims at measuring relative improvements seen through the eyes of a professional project manager. I will not focus only on project control with Earned Value Management but integrate the three components of Chap. 3 (baseline scheduling, risk analysis, and project control) in one decision support system. Chapter 8 expands on the discussion of Chap. 5 and tries to combine the simple project control method (rules of thumb) with the four advanced statistical control methods to search for an ideal combination of both. By merging the best methods of both worlds (academia and practice), we can present a new way to control projects. This new method, called *analytical project control*, is almost as simple as the previously discussed rules of thumb but has similar performance to the statistical tolerance limits discussed earlier. Finally, Chap. 9 discusses a new study on project duration forecasting that is highly relevant and applicable in practice. It uses historical project data to predict the progress of future projects and is known as *reference class forecasting*.

Chapter 7
Control Efficiency

Open a dictionary or search for a definition of the word *efficiency* on the Internet, and you will easily find that it is defined as the ratio of the useful work performed in a process to the total effort (energy, time, . . .) expended. Being efficient means paying the most attention to the most important things, while not wasting time on the minor details. Efficiency requires performing or functioning in the best possible way with the least waste of time and effort. The simplest and most elegant definition of efficiency is given by the American–Canadian psychologist Daniel Levitin:

> The obvious rule of efficiency is you don't want to spend more time organising than it's worth.

Indeed, it is not always worth paying much attention to small details, and that is something that academics do not always understand. While presenting algorithms and new methodologies in the literature, they rarely think about the ease of use or the amount of work that these new methods entail. These new ways of managing projects may, at least according to their studies, perform better than the existing methods, but they do not necessarily make the project manager more efficient. However, efficiency is a more important goal in practice than being able to perform better. After all, no one wants to spend more time than necessary to use advanced data-driven methodologies. If the effort to collect, process, and analyse project data does not lead to better decisions, better actions, and ultimately higher project success, then using these new methodologies is certainly not worth it. While working on some studies on improving control methods (as discussed in the previous part of the book), I encountered quite a bit of opposition from the practical field because implementing and using these methods would simply require too much effort. This is why I decided to approach my research not only from a performance point of view, but also from an efficiency point of view, as will be discussed in this chapter.

I should have known better and realised that the pursuit of efficiency is an important goal in practice. When I mentioned in Sect. 3.3, the controversy about the correct level of detail in an Earned Value Management (EVM) system, this was

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_7

actually already a discussion on how to use this system very efficiently. This chapter builds on this discussion by determining the appropriate level of project control, thus attempting to compare the methods for achieving the optimal level of efficiency. The idea of adding efficiency to project management is a simple one, but the way it should be brought into academic research was a little less obvious. In this chapter, a very general definition of *control efficiency* is used, balancing the time the project manager spends measuring the project performance to identify problems (*effort of control*) and the results achieved after taking actions to solve these problems (*quality of actions*), as follows:

$$\text{control efficiency} = \frac{\text{quality of actions}}{\text{effort of control}}.$$

In the next two sections, the numerator (Sect. 7.1) and denominator (Sect. 7.2) of the control efficiency formula are explained in detail. Afterwards, some results of our computational studies on artificial and empirical project data will be discussed, in which not the performance is measured, but the efficiency as proposed in the previous formula. I will close the chapter with an integrated project control system that combines the three components of data-driven project management (*baseline scheduling*, *risk analysis*, and *project control*) into one integrated decision making system to improve the overall efficiency of project control.

7.1 Effort of Control

A central concept in defining the effort of control is the well-known *work breakdown structure* (WBS). The level chosen in the WBS defines the starting point of project control and can be used to assess the effort of control from two very different points of view. On the one hand, one can define the control point at a very low level of this WBS, which is not recommended for an EVM system as I argued earlier in Sect. 3.3. It would lead to too much detail and would make the EVM system hopelessly complicated. The control point can therefore best be put at higher WBS levels, so that the project control methods are carried out with a lesser degree of detail. These two extreme forms of control (low or high levels of the WBS) lead to two types of project control that can be implemented in two fundamentally different ways, each with its own underlying methodology, but both aiming to solve project problems at an early stage, finding the roots of the detected problems, and (hopefully) solving them so that the project can be put back on track. These two alternative project control methods are the subject of this section.

The preparation of a WBS is an important step in managing and mastering the inherent complexity of the project. It involves the decomposition of major project deliverables into smaller, manageable components until the deliverables are defined in sufficient detail to support development of project activities (PMBOK, 2004). The WBS is a tool that defines the project and groups the project's discrete

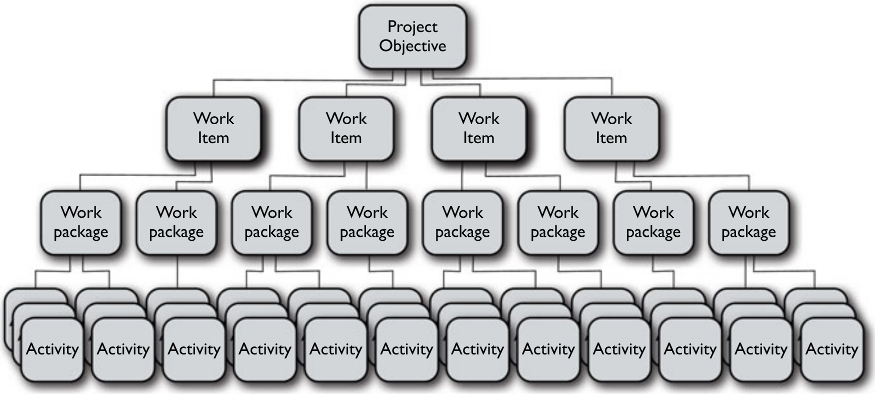


Fig. 7.1 The work breakdown structure

work elements to help organise and define the total work scope of the project. It provides the necessary framework for detailed cost estimation and control along with providing guidance for schedule development and control. Each descending level of the WBS represents an increased level of detailed definition of the project work. The WBS is often displayed graphically as a hierarchical tree. Figure 7.1 displays an illustrative WBS consisting of three levels and a root node, as follows:

- Project objective: The project objective consists of a description of the scope of the project. A careful scope definition is of crucial importance in project management and contains a list of specific project goals and deliverables to satisfy the needs of the stakeholders.
- Work item: The project is broken down into manageable pieces (items) to be able to cope with the project complexity. The work item level is often seen as the level where responsibilities are assigned by giving people the ability to act independently on one or more work items and take decisions to further subdivide the work items into the lower level components.
- Work package: Work packages are the result of the subdivision of work items into smaller pieces and are important for project control. Ideally, the monitoring and collection of time and cost data occur at this level to indicate problems in groups of activities.
- Activity: The lowest level of the WBS is the level where the accuracy of cost, duration and resource estimates as well as the precedence relations must be defined for the construction of the baseline schedule.

The remainder of this section will explain the two previously mentioned alternative and opposing project control methods using two very extreme starting points of control in the WBS (one starting at the very top of the tree and working downwards, and another method starting at the bottom but gradually working upwards). These two methods are extreme ways of controlling projects, and the

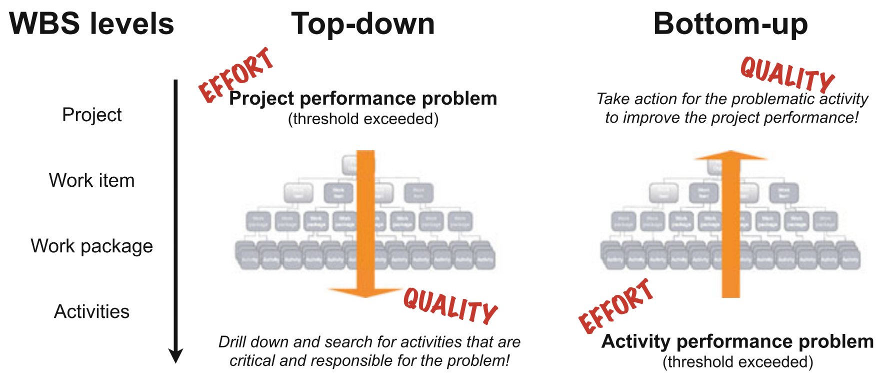


Fig. 7.2 Top-down and bottom-up project control *(separate view)*

real practical implementation of a control system will, of course as always, be somewhere between these two extremes. Figure 7.2 graphically illustrates the two extreme ways of controlling projects on the WBS. The so-called *top-down project control* method works in a downward direction, while the *bottom-up project control* method starts at the activity level and works in the opposite, upward direction. Most of the discussion in the following sections of this chapter is based on two studies that I published in the journal *Omega—The International Journal of Management Science* (Vanhoucke, 2010b, 2011), and it is with some pride that I say that many of the studies subsequently conducted by my research team build upon the foundations of these two studies.

Top-Down Project Control

All control methodologies discussed in Part II make use of *Earned Value Management* metrics and belong to the class of *top-down project control* methods. They monitor the progress using three key metrics (*planned value*, *actual cost*, and *earned value*) and measure the overall performance of the project with schedule and cost performance indices (SPI, SPI(t), and CPI). The performance metrics do not include any information about the root cause of project problems since they are not calculated for each activity individually, but for a group of activities at higher levels of the WBS (e.g., for work packages, or even at the higher work item levels or at the root node of the WBS). Consequently, the performance of a project is measured by only a few (one for each work package) or only one (at the root node) performance metric to express the time and cost performance of the project, and it is up to the project manager to decide whether this performance metric indicates a problem or not.

This top-down approach was presented earlier in Sect. 5.2 where control points were set at strategic places in the project network, and each control point measured the performance of only a set of activities (a feeding path, a subnetwork, etc.) to indicate project problems. In this view, EVM control is seen as a much simpler alternative to the detailed *critical path based* planning tools (which monitor each individual activity separately) and therefore provides a quick and easy helicopter view of project performance at the higher levels (or root node) of the WBS. When these metrics indicate a project problem, they act as warning signals to drill down to lower levels of the WBS, possibly to the detailed activity level, to find the root cause of the problem. The specific set-up of such system ensures that the project manager does not need to follow every small detail when managing the entire project. The fact that the performance metrics are measured at a higher general level, and a drill-down to the level of activities is only necessary when the performance no longer meets the predefined requirements, gives the manager room to save time and costs and reduce the *effort of control* to the absolute minimum. After the drill-down, corrective actions should be taken on those activities that are in trouble (especially those tasks that are on the critical path). However, such a system also entails some dangers, since detailed project control is no longer performed. It is therefore perfectly possible that the control limits give a wrong signal, and so the drill-down is performed (requiring *effort*) without any real project problem (overreaction). The reverse situation can also occur, where the performance metrics do not indicate a problem, while problems are actually accumulating in the underlying levels. No system is perfect, which is why Sect. 5.3 discussed the probability of overreaction and the detection performance as metrics to measure the signal quality.

In my research on top-down project control, the *effort of control* is measured in several ways, each time trying to evaluate how much time the project manager should spend finding activity problems after a drill-down due to a warning signal. More specifically, the effort depends on a number of characteristics such as (i) the number of control points in the system (more control points require more effort) and (ii) the number of times the system generates a warning signal (more warning signals mean more effort for drilling down). In order to keep the effort of control at an acceptably low level, it is therefore important to choose the number of control points very carefully, but also to set the control limits of the control charts at the correct level. Since the control limits now act as *threshold values* to drill down and search for problems (and take actions if necessary), a control chart should be prepared with the utmost care. Figure 7.3 displays an example control chart with a static lower control limit set at 60%.[1] The graph shows the periodic performance measurements with the schedule performance index (SPI) and the cost performance index (CPI) to assess the time and cost performance of the project. When the threshold values are exceeded (below the acceptable threshold of 0.60), the charts give a warning signal to drill down to search for problems (requiring effort) and, if necessary, to take corrective actions. This graph perfectly illustrates how important

[1] Such static threshold corresponds to the static control limits of the rules of thumb (ROT) approach in Chap. 5. Recall that these control limits can be variable along the planned project progress, as illustrated in Fig. 5.3.

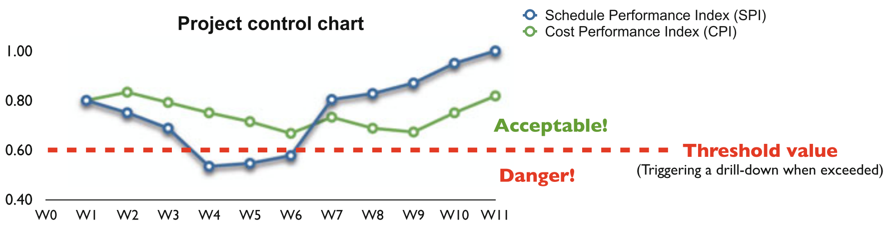


Fig. 7.3 Example control chart (static thresholds)

it is to put the threshold values at the best possible level. It is clear that if a project manager sets the threshold values too low (for example at a value of 0.40), no project problem will be detected. When problems arise, they will quickly escalate and put the project in very dangerous territory. If the system does not detect these problems, it will not trigger the project manager to take appropriate action, which could lead to catastrophic consequences. However, such an approach has the advantage that the control effort will be very low, since the project manager will only have to drill down into the WBS in very extreme situations when the project performance is extremely low. Most likely, this lack of effort will result in the project not being properly monitored and the chances of success are very slim. However, setting the threshold too high (e.g., to a value of 0.95) will lead to the opposite behaviour. In this case, a multitude of warning signals will be generated every time a minor delay or cost overrun is detected, and the project manager will have to continuously zoom in on issues that have little to no impact on the project objectives. Such a control system is a total waste of time and costs, and consequently too much effort for nothing.

Bottom-up Project Control

The top-down control method only looks for problems in activities after warnings have been generated and, of course, only requires actions if these activities actually endanger the project. An alternative approach is to start with the activities and only monitor those activities that could potentially have a major impact on the project should problems arise. The so-called *schedule risk analysis* methodology already discussed in Sect. 3.4 of Chap. 3 can be used to build such a control system, and we will refer to this method as a *bottom-up project control* method because it uses activity information (at the bottom level of the WBS) to control the project.

A schedule risk analysis uses Monte Carlo simulations to measure the sensitivity of each activity of the project. Probability distributions are used to model activity duration variability, and the simulation runs generate different sensitivity values, expressed as values between 0% and 100%, such as the criticality index, the significance index, or the schedule sensitivity index, as discussed earlier in this book. These values can be plotted in a sensitivity graph, as illustrated in Fig. 7.4 for

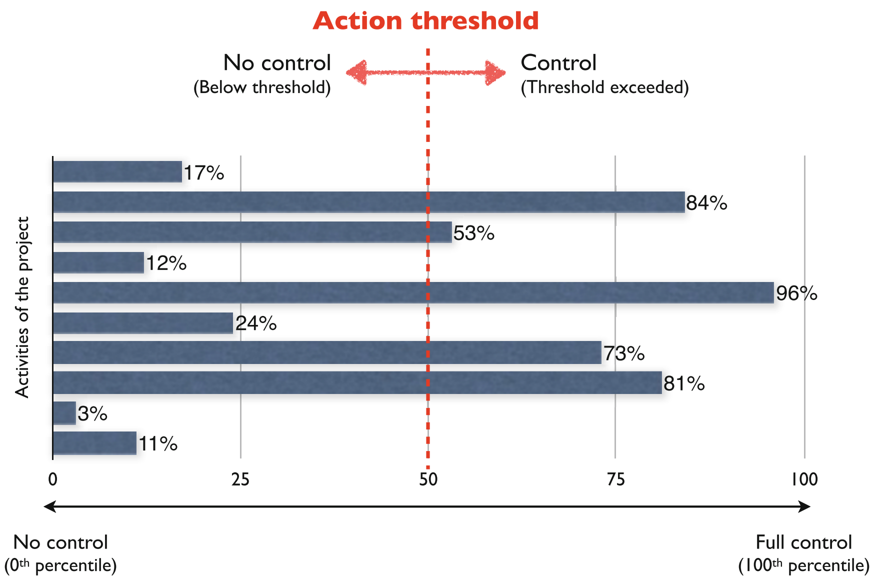


Fig. 7.4 Example sensitivity graph (10 activities)

a project with 10 activities. Activities with high values for the metrics represent very sensitive activities, meaning that a problem in these activities can cause significant damage to the project. These activities must therefore be closely monitored as any small delay can have a major direct impact on the project. However, the activities with a low value for the sensitivity metrics are likely to have a smaller impact on the project should a problem arise, and so these activities are less important to monitor very closely. It is therefore very important to make a distinction between activities that require little follow-up and activities that must be followed up very closely. As is the case with control charts, a value for the action threshold must be chosen that reduces the effort of control to a minimum, without major problems going unnoticed. The threshold value is represented as a vertical line on the figure, where all activities on the left of the line should not be followed at all, while all other activities are closely monitored. The more the line is shifted to the left, the greater the number of activities that are monitored. This not only means that problems are detected faster, but also that the effort of control becomes much greater. This method is called a bottom-up control method because the project monitoring now starts from the activity level and only controls a part of the (most sensitive) activities. However, when such activities are in trouble, the project manager will have to take actions to prevent the total project from getting into trouble, and thus the actions must have a positive impact on the total project (represented at the root node of the WBS). So there is no drill-down looking for problems, but rather an upward movement in the WBS looking for a positive outcome of corrective actions.

Setting the action threshold to the correct values is essential for the bottom-up control system, just as setting the correct tolerance limits is for a top-down control system, and this is best illustrated by introducing three types of project managers. I am sure everyone knows a person in their company or university who belongs to each of these three types. Figure 7.5 shows these three types of project managers, with each type taking a different approach in setting the thresholds for actions. Each type of project manager defines their desired effort level in a completely different way.

The *control freak* puts the action threshold close to the left side of the graph and is not afraid to spend almost every hour of the day monitoring the performance of most activities. In the sample graph, the threshold is set to 15%, and therefore, 70% (7 out of 10) of the activities are classified as potentially hazardous with high sensitivity values exceeding the threshold. The control freak performs poorly on the control effort, as 7 out of 10 activities are under the control freak's control (*too high!*). Controlling almost all activities during project progress is not very practical, and there will be little to no time left to focus on other aspects of managing the project. However, thanks to the control freaks' desire for intensive control, this approach is likely to result in early detection of potential problems and corrective action may be taken the moment that problems occur.

The *lazy manager* is not the control freak's best friend and takes a completely opposite approach. Lazy as this manager is, the threshold is set to the far right of the graph (as shown in the right graph, the threshold is set to 88%). A lazy manager performs very well in terms of control effort (*very low!*), as almost no activity exceeds the threshold. In the example, only one activity requires intense monitoring, while the others can be ignored, leaving a lot of time to do things other than monitoring activity performance. Due to this little effort of control, not all activity problems will be detected in a timely manner, and therefore many of the project problems will remain hidden until they suddenly appear. Often, when they show up, it will be too late to fix them, leading to loss of time, money, and possible project failure.

The *efficient manager* is a data-driven project manager who wants to combine data analysis and experience and strikes a balance between the intensity of a control freak and the relaxed feelings of the lazy manager. Ultimately, the efficient manager wants to be able to detect the problem in advance, just like the control freak, but at a much lower effort, closer to the effort level of the lazy manager. Such a manager does not want to concentrate on controlling almost every activity of the project, but only wants to concentrate on the most dangerous activities in order to keep the effort at a reasonable level. Therefore, the efficient project manager will set the action thresholds somewhere in the middle of the graph, between the thresholds set by the control freak and the lazy manager. Using this approach, the central question is whether this manager can achieve comparable project results to the control freak (i.e., detecting project problems in the early stages so that corrective actions can get the project back on track) with a much lower effort than the control freak, preferably as close as possible to the effort of the lazy manager. If so, this manager is much

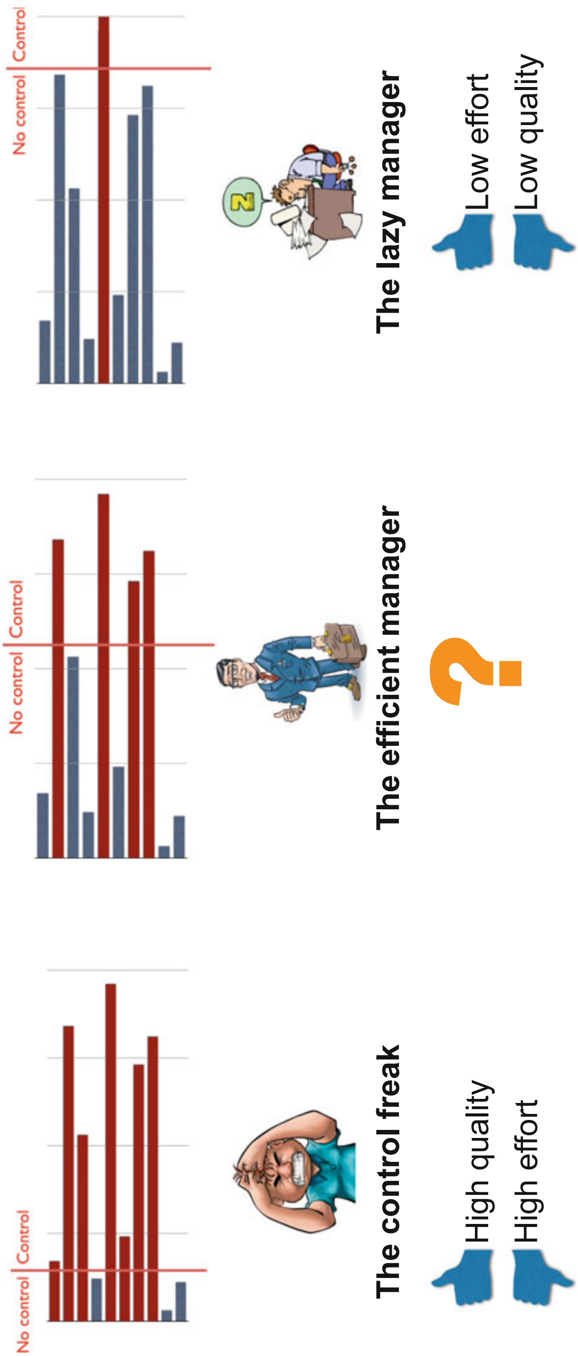


Fig. 7.5 Choosing a desired level of effort (to be efficient)

more efficient than the other two project managers. The research to answer that question is the subject of the next section.

7.2 Quality of Actions

The two control methods use action thresholds that determine how much and which activities will be monitored and thus determine the effort of control of the project manager. Minimizing this effort is of course not the primary goal, and these methods aim to detect project problems in a timely and correct manner in order to be able to take the right actions. The quality of these actions is discussed in this section and is in the denominator of the control efficiency formula. It is very important to note that the concept of *quality of actions* does not refer to very specific ways in which project managers solve problems such as shifting resources, working overtime, or changing the project scope. Instead, it does refer to the impact that the actions have on the performance of the project. In other words, it measures whether the actions have actually had a positive impact on the project leading to beneficial results and solved problems that brought the project back on the right track. Such corrective actions can only have a positive impact if (i) the problems are detected in a timely manner and (ii) if the actions are taken on the right activities, i.e., on the activities that caused the problems. Problems detected too late (because they did not generate a warning signal) or actions taken on the wrong activities (because they had a wrong value for the sensitivity metric) can have a major impact on the quality of the actions taken by the project manager and can put the project in a danger zone where it is impossible to recover. Consequently, the concept of *"quality of actions"* refers to the timely detection of a project problem for the activities in difficulty such that the positive impact of the actions is maximised.

This particular definition of *action quality* cannot be properly understood without simultaneously considering the concept of the *effort of control*. Indeed, if one does not care about the level of effort, the highest quality will be easily obtained by setting the thresholds to very extreme values. For top-down project control, this means that the thresholds are simply set to 99.9%, so that every small deviation results in a drill-down to look for problems. For bottom-up project control, it means that the thresholds are set to 0.01% such that every activity is considered as very sensitive. In this case, every activity will be part of the control set (just like how the control freak would work), and every action will result in an improvement in project performance, which will likely lead to overall project success. Every problem will therefore be noticed in time, and the quality of actions will be very high. However, the effort will also be very high, so that the control efficiency will be too low. This is exactly what the efficient project manager wants to avoid. With lower effort levels, the manager can become more efficient, possibly with the risk that the quality of actions also decreases somewhat, but as long as the efficiency is high, the efficient manager is better than the control freak and the lazy manager. Accordingly, the best performing control system is the system that leads to the highest corrective

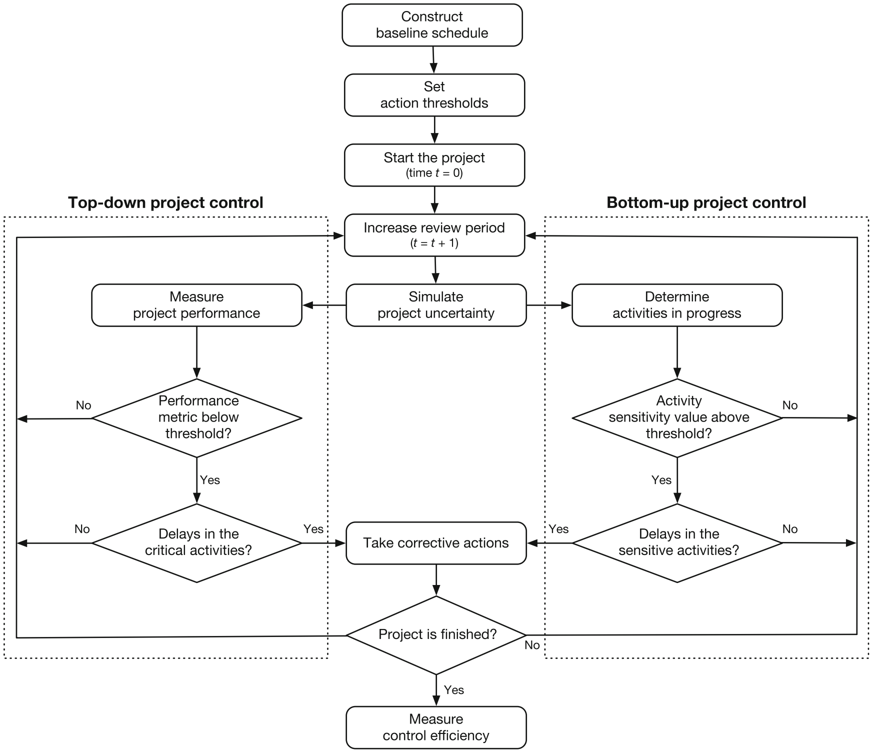


Fig. 7.6 Simulation experiment for *top-down* and *bottom-up* control

action value (numerator of the *control efficiency* formula) with the lowest possible effort (denominator of the formula). A system that can distinguish between real problems and false problems, so that the activities in difficulty can be detected in a timely manner, will probably lead to the highest efficiency, and this was precisely the subject of the study described in this chapter.

In the following paragraphs, I will give a general overview of how this study was conducted, without going into technical details. The experiment was performed on a set of artificial projects. Static and dynamic simulation runs were used to construct the control charts and the sensitivity graphs (*static*) as well as to simulate the project progress (*dynamic*), which is very similar to Phases 2 and 3 of Sect. 5.1. Figure 7.6 shows the design of the study and shows how the effort of control and the quality of corrective actions are measured during the simulated progress of the project. The experiment starts with the construction of a project baseline schedule (which, as always, is considered a reference point) and setting predefined action thresholds on the metrics (e.g., on the SPI(t) for top-down control and the SSI for bottom-up control). After that, the project progress is started, and the algorithm checks for each review period (t in the figure) whether or not these action thresholds have been exceeded. Whenever an action threshold is exceeded, a drill-down (for top-down project control) or a search for the expected impact of the sensitive activities (for bottom-up project control) is performed to solve the problems, and as this takes effort, the *effort of control* (initially set to 0) is increased each time. When actions are taken, the quality of the actions is also measured. For top-down project control, the corrective actions are automatically taken on the critical activities when they have too much delay, while bottom-up project control only takes actions on the activities with a delay that exceed the minimum value of the sensitivity metrics (i.e., the action threshold). Several runs are performed, each time with different levels of effort by changing the action thresholds set at the start. This results in different values for the quality of the corrective actions, and thus, different values for the efficiency of the control. The two alternative ways of project control are indicated in the two halves of the figure with the top-down project control method on the left side and the bottom-up project control method on the right side. This simulation process is repeated for each review period until the project is completed, and the control efficiency is reported and compared to previous runs that were performed with different values for the action thresholds.

Since the simulation study of Fig. 7.6 is a fully automatic project control study, the algorithm must automatically choose between a series of possible actions when thresholds are exceeded. To this end, three different types of corrective actions have been implemented, each of which can be used under different settings:

- Reducing activity durations: Shortening the activity duration is the most logical action to solve delays, but this results in an increasing cost. For this reason, the duration reduction should not be taken arbitrarily, but only if the cost increase is still within the allowable budget. Such an action is known in the project management community as *activity crashing* and refers to the allocation of additional resources to delayed activities to make up for the work.
- Parallel execution of activities. The project network defines the logical sequence of activities and is used to construct the baseline schedule. Any precedence relation between two activities means that one cannot begin until the other is completed, but this logic is not always followed as the project progresses. Even the logic of the project network can be subject to unexpected changes, and activities that were previously thought to be performed in a sequence can be performed in parallel if necessary. These unexpected activity overlaps resolve delays, but can incur costs such as additional risk of rework or set-up time, and are known in the project management community as *activity fast tracking*.
- Changing the baseline schedule: In some cases, the additional unexpected work or changes to the original project objective are so significant that the original project baseline schedule must be replaced or updated with a new plan to define a new reference point. Although I always advise not to change the baseline schedule too often, sometimes the new project reality is so different than initially thought that a project manager has no other choice. A re-baseline of the project means removing all planned and progress data from the past and includes the decision to start from scratch with the modified project. It is a drastic, but

possible, action that should be avoided if possible and is known in the project management community as *project re-baselining*.

The *control efficiency* algorithm was initially tested on 4400 artificial project instances, the main conclusions of which are discussed in Sect. 7.3 and published in Vanhoucke (2011). One year later, the results were validated on a series of 48 empirical projects from 8 different companies in Belgium in 13 sectors, the results of which are presented in Sect. 7.4 and published in Vanhoucke (2012a). Since then, the use of both artificial and empirical project data has become a standard in my research team, and this will be further discussed in Part IV of this book.

7.3 Accuracy Pays Off

In a previous chapter, it was already shown that the network structure has a major impact on the *accuracy* of predictions, and the insights we gained from it even led to the best compliment I ever received about my research from Tony Barrett that I quoted on the end of Chap. 4. More specifically, it was shown in Fig. 4.2 of this chapter that the accuracy increases as the project becomes more serial. The closeness of a project network to either a completely serial or parallel network can be measured by the so-called *serial/parallel* (SP) indicator[2] that ranges between 0% (completely parallel project) and 100% (completely serial project) and is calculated as follows:

$$SP = \begin{cases} 1 & \text{if } n = 1 \\ \frac{m-1}{n-1} & \text{if } n > 1 \end{cases} \tag{7.1}$$

with n the number of activities in the project and m the number of activities on the longest path in the project network.[3]

The study of the current chapter does not look at the accuracy of the predictions, but at the efficiency of control. Figure 7.7 displays the results of the control efficiency experiment on the large set of artificial projects and makes use of the SP indicator. The graph displays the control efficiency values (higher values are better) for different projects with different topologies of the network. More precisely, the projects are classified by their network structure, ranging from very parallel projects (where most activities can be scheduled and executed simultaneously) at the left to very serial projects (where activities mostly are executed sequentially) at the right, as measured by the SP indicator. The graph contains four different lines, labelled as

[2] You might recall that I introduced this SP indicator already in Chap. 6 to model the network topology (NT) in the Bayesian network study.

[3] Note that this longest path is not the critical path (which is the path with the longest duration), but rather the path with the maximum number of activities.

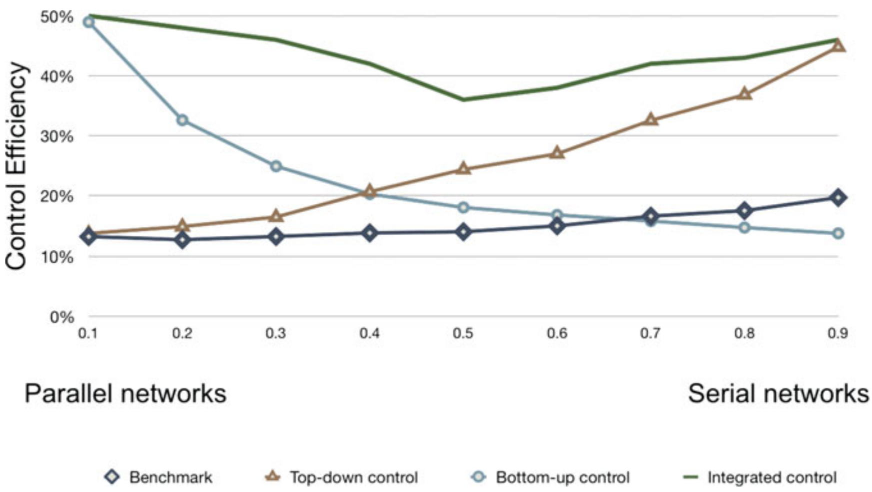


Fig. 7.7 Control efficiency—Results of an experimental study

benchmark, *top-down control*, *bottom-up control*, and *integrated control*. Each line expresses a different way of controlling projects, which is briefly outlined along the following paragraphs.

Benchmark The line with the diamonds is referred to as the *benchmark* line as it represents the simple rules of thumb (ROT) approach of Chap. 5 using the traditional *schedule performance index* (SPI) to monitor the timing of projects. I have discussed before that this SPI performance measure is known to be flawed (because it always ends at 100% at the end of the project), but it is nevertheless implemented in most commercial software tools. In Chap. 4 of Part II of this book, I argued that the quirky behaviour of the SPI is a direct result of an error in the formula and therefore does not accurately measure the time performance of a project in the second half of the project's progress. I thought this error was pretty straightforward, and I thought everyone knew it could lead to odd results, but I was surprised how many project managers blindly relied on it when using project planning and control software tools. What struck me most was that many of these project managers were aware of the flaw in the SPI formula, but thought it would have little effect on their control efficiency. However, the control efficiency graph shows the impact of the SPI mistake and shows that the ROT approach performs relatively weak compared to the other control methods (other lines). In addition, the project network structure, measured by the SP, has little to no impact on this low control efficiency. Fortunately, the chart contains three other lines.

Top-down Control The line with the triangles is the control efficiency for the *top-down project control* approach using the improved version of the schedule performance index, the SPI(t), which is known to be reliable from the start to the finish of the project. Although the formula of SPI(t) is not fundamentally different from the SPI formula,[4] it allows the project manager to reliably assess the time performance of the ongoing project at each stage of progress. The graph shows that the control efficiency is always higher for the SPI(t) top-down control method than for the SPI control method (benchmark line), and the difference between the two methods increases with increasing SP values. The finding that the top-down project control method with a reliable indicator (SPI(t)) outperforms the one with a flawed indicator (SPI) is not surprising, but the positive impact of the network structure, especially for serial projects, was unknown in the scientific literature. In Chap. 4, it was already shown that the *earned schedule method* (which relies on the SPI(t)) outperforms two other traditional methods that depend on the SPI (*planned value method* and *earned duration method*) for predicting the total duration of an ongoing project, but now the graph shows that the higher prediction accuracy also results in a higher control efficiency. Accuracy pays off!

I have to admit that I did not quite expect this result during the research, but in retrospect I was actually not surprised that more serial projects were more manageable with a top-down project control method. That is how it often goes in research: Unexpected results seem quite logical in retrospect, and it works a bit like Julian Casablancas from The Strokes puts it:

> The best solutions are often simple, yet unexpected.

Nevertheless, the outcome of this research is very special to me, as it has given me access to the practice of some very interesting UK consultancy projects, but also led to the aforementioned (in Chap. 1) *more than one million Euro* research project. This project was the start of my team's significant growth and multiple collaborations with great people around the world. The icing on the cake was the keynote speech at the EVM World Congress in Florida in 2012 together with Stephan Vandevoorde with the title *"When time is money, accuracy pays dividends!"*

Bottom-up Control The line with the circles represents the control efficiency for the *bottom-up project control* approach and shows the exact opposite behaviour. Indeed, the graph shows that for the projects with poor performance for top-down control (i.e., the low value SP projects), the bottom-up project control offers a valuable alternative. The bottom-up control approach performs best when SP values are low (i.e., for parallel structured projects), but the control efficiency gradually decreases when the project networks have a more serial structure. I previously explained that bottom-up project management relies on sensitivity metrics obtained through a schedule risk analysis, and the experiments showed that the best results could be obtained by relying on the *schedule sensitivity index* (SSI). Three other sensitivity measures (criticality index, significance index, and cruciality index) underperformed, sometimes even worse than the benchmark method, but the SSI

[4] I argued in Chap. 4 that the calculation of the SPI(t) requires no additional input and is therefore nothing more than a corrected version of the SPI.

outperformed all existing metrics and is therefore the most reliable sensitivity measure for bottom-up control.

Integrated Control The straight line at the top of the graph was initially not included in the original study and will therefore be discussed later in this chapter. The line represents the *integrated project control* approach, using top-down control (via SPI(t)) and bottom-up control (via SSI). Integrating both methods into one integrated control system increases the control efficiency considerably and is therefore the most recommended method. Before explaining this integrated control approach in Sect. 7.5, I first want to show some empirical results from a small series of real projects in the next section.

7.4 Empirical Evidence

Despite the unexpected enthusiasm and interest of professional project managers in corporate workshops and professional conferences, many project managers continued to question the usefulness and validity of the results due to the fact that they were obtained from experiments with artificial project data. It is true that the previous experiments were conducted on a set of artificial projects, where the values of the SP indicator were artificially controlled without any link to empirical project data. The advantage of artificial data is that researchers have full control over the project parameters (e.g., the network structure measured by the SP) to obtain general results, but the danger, of course, lies in the fact that little or no link is made with real project data. Partly because of this, many professional project managers wanted to see how this control efficiency study would perform on real projects, and I was almost obliged to use real project data for this study. While I have often tried to convince them that using real data would not provide additional insights, I have to agree that research should not just focus on delivering generic results (on a wide range of artificial projects), but that the results should also be applied to real data. It is interesting anyway to see which types of projects have a rather parallel structured network and which other projects are closer to a full serial network. I started collecting real project data early in my career, but I had initially very much underestimated the difficulty of collecting real project data. It took me a few years to collect a sample of just 48 projects from just 8 companies, which I used in the study. I quickly realised that collecting real project data needed a more structured approach, and I will introduce the readers to a formalised approach on how academics can collect and analyse empirical data later in Chap. 13 of this book.

At the time of the control efficiency study, I could not use this formalised approach, and all I could do was visit several companies to get some data. I got some partial project data with a lot of unknowns, and I worked my way through the unstructured mess until I was able to present some results. I ended up with 13 classes of projects with a similar network structure from 8 different companies, a total of 48 projects on which I tested the different control methods. Figure 7.8

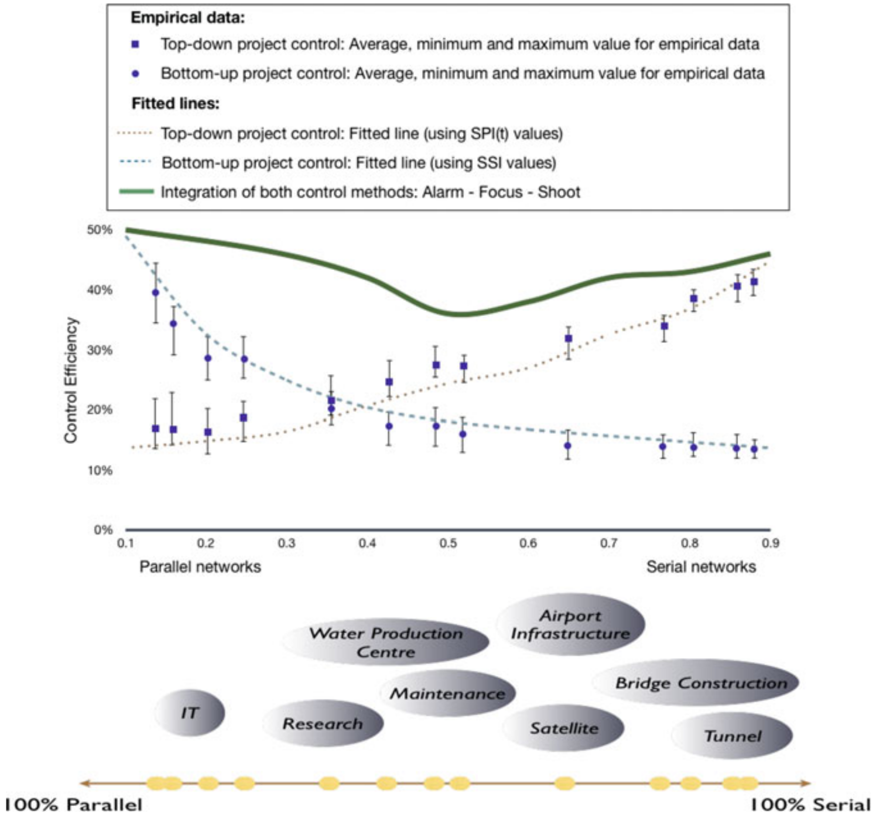


Fig. 7.8 Control efficiency: Results of empirical study (Source: Vanhoucke (2012a))

shows the 13 classes of projects with their corresponding SP values on the x-axis of the bottom image, ranging from projects with low SP values (almost completely parallel, e.g., IT projects) to projects close to a complete serial network (SP values high, e.g., construction projects). Fortunately, the results were completely in line with the theoretical results from the previous graph. The graph just below the text in Fig. 7.8 is almost identical to the graph in Fig. 7.7 and shows the two lines for top-down (SPI(t)) and bottom-up (SSI) project control as well as the integrated line at the top. Box plots are shown for each control method showing the results of the simulation study on the 13 classes of empirical data. Since simulations are performed under different settings (each time generating different values for the uncertainty about the duration of the activity and the possible corrective actions taken when thresholds are exceeded), different runs often show slightly different values for the efficiency of the control. The box plots show the mean values for each project as well as the minimum and maximum values and show that these values are surprisingly close to the theoretical results. The lines not only show that the results of the experimental study of the previous section are relevant for professionals who want to rely on these control methods, but also that the use of artificial data is often good enough to explain real phenomena.

The results of this empirical validation study have helped me to convince professional project managers that artificial project data are just as good as empirical project data for academic research. In addition, this study was published in the *International Journal of Project Management* (Vanhoucke, 2012a), which is typically a journal that is widely read by practitioners. The final step to fully convince professional project managers that the experimental results make sense for *their* projects was to go one step further by merging the two alternative methods as one integrated project control system. This integrated project control system is represented by the solid line at the top of Figs. 7.7 and 7.8 and discussed in the next section.

7.5 The Control Room

As I wrote before, I had the opportunity and privilege to present these research results at workshops and conferences. I was also fortunate enough to have many professional project managers talk to me about these results that were now based on both artificial project data and empirical validation. However, there was one problem that still bothered me: I always presented the results of the control efficiency study as two separate control systems (bottom-up and top-down) and never offered the public an integrated view. I could easily tell professionals which of the two control methods would work best for them, depending on the structure of the project, but I never offered them an integrated project control approach that combined both methods into one decision support system. Of course, I knew the best approach was a combination of these two control methods, as the graphs showed that the solid line outperforms all other methods, but I could not really translate this integrated approach into a clear, easy-to-understand project control methodology. Nevertheless, the real practical way to control projects is not to choose one method or the other, but to combine both methods in a smart way to reach the highest possible values for the control efficiency. The only question: how do you do such a thing?

Lisbon has never disappointed me, and I have lost my heart to this city not only because of the good food and sunny weather, but also because it is where I find the most inspiration for my research. After spending a year in Lisbon during which I could write a new book, I was finally able to come up with a suitable answer to the above question. As often, the answer was hidden in the methods, namely in the specific starting location in the work breakdown structure and the direction (up or down) in which these methods work. Keep in mind that the top-down control method starts from the root node of the WBS and generates warning signals to go to the lowest level looking for problems. This lowest activity level is then the starting level for the bottom-up method and distinguishes between low-sensitive and highly sensitive activities, the latter requiring more intensive monitoring. This sequential process of starting from a top-down approach (to reach the lowest WBS levels) and then moving to a bottom-up approach (by focusing on the most sensitive activities) offers a solution to integrate both methods as a single-project management method. I have called that approach the *alarm–focus–shoot* approach in my book *"The data-driven project manager: a statistical battle against project obstacles"* (Vanhoucke, 2018), which is graphically represented in Fig. 7.9. The approach consists of the following three phases:

Phase 1: Alarm The top-down project control method merely acts as an alarm system to alert the project manager that problems have happened in the project. Indeed, the EVM performance metrics (CPI and SPI(t)) of the top-down control method are used to monitor the costs and time of the ongoing project at the project level and serve as a warning signal (the *alarm*) to identify project problems without giving any detail about the nature of the problem at the activity level. Their purpose is merely to provide quick and easy metrics to assess the health of the project, and thresholds can be set as simple rules of thumb or by using statistical tolerance limits (cf. Chap. 5) to move to the lower levels of the WBS. Consequently, the top-down approach is used as the first step in the integrated system to descend into the WBS.

Phase 2: Focus After the alarm has gone off, this second phase requires a detailed examination of the activity level. Since the first phase clearly gave an indication that something went wrong, the project manager is now at the activity level of the WBS to look for the causes of this project problem. Such a search should begin immediately, aimed at finding the root causes of the problem as quickly as possible. However, investigating each individual activity can be tedious and takes too much effort, so this search should be optimised and limited to the most *dangerous* activities, which will most likely have the greatest impact on the project objectives if delayed. This is precisely the purpose of the bottom-up project management method that uses activity sensitivity metrics to enable the project manager to distinguish between high-sensitive and low-sensitive activities. These sensitivity values allow the project manager to limit the search and focus only on the activities with the highest values of the *schedule sensitivity index* (SSI) and pay less or no attention to the other activities that—in case of delay—will not have a big impact on the project anyway. Therefore, the second phase should consist of the bottom-up control method that can be used as a tool to *focus* on the most sensitive activities that require the project manager's attention every time an alarm is given.

Phase 3: Shoot Once the problems are detected in one or more activities, the project manager must decide when and how to respond to the project problems. The project manager must come up with an appropriate action to get the project back on track and decide how to use the limited resources (time, money, resources) to aim correctly and then shoot exactly where the problem lies. After this action, the project manager should see if the project recovers and continue with this integrated project control approach until the project is completed.

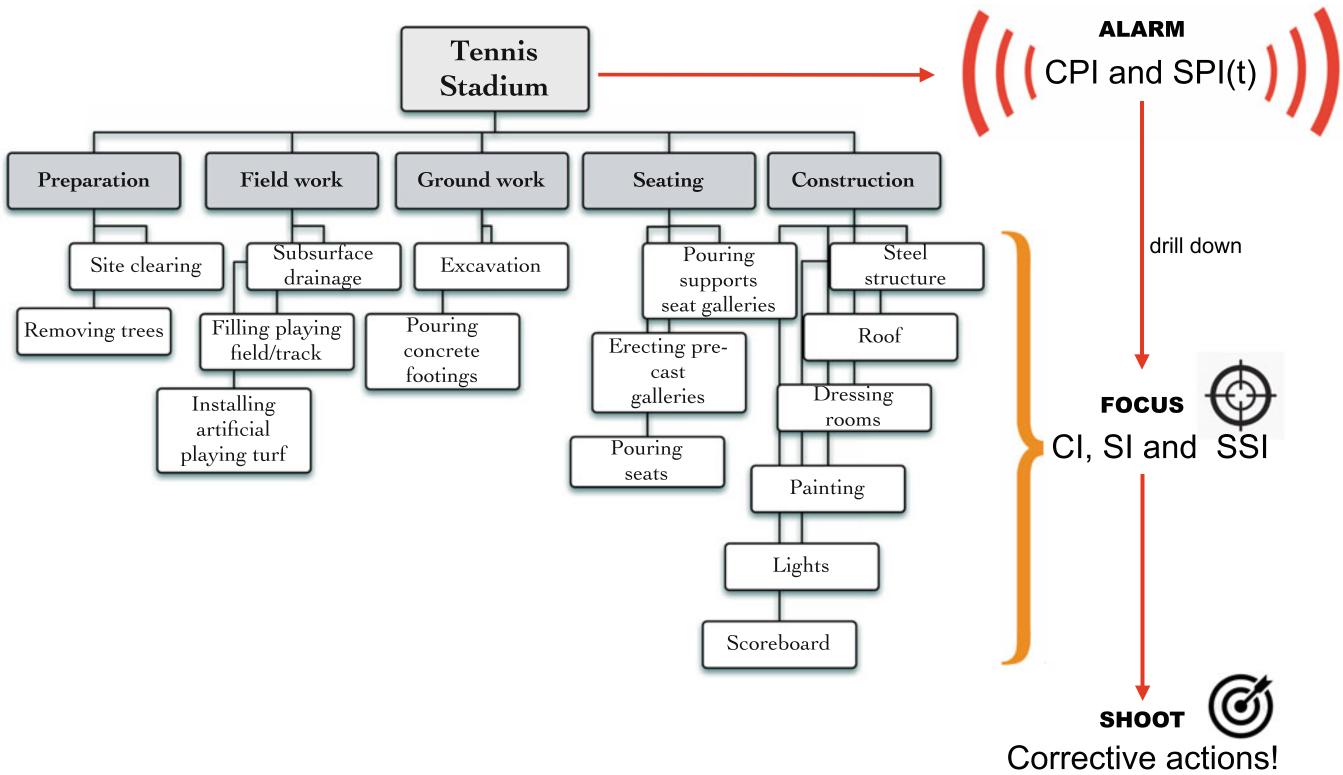


Fig. 7.9 Top-down and bottom-up project control *(integrated view)*

Afterthought

Before concluding this chapter, let me reiterate that this *alarm–focus–shoot* project management approach fits perfectly within the three components of data-driven project management and the dynamic planning methodology of Fig. 3.1 of Chap. 3. First of all, the approach requires the construction of a *baseline schedule* (first component) as a reference point for the other two phases. Additionally, the *focus* phase includes a *schedule risk analysis* (second component) to determine the sensitivity of activities, and the *alarm* phase consists of setting up a *project control* system with EVM (third component). I hope that the presentation of this new integrated system arouses the interest of professional project managers and convinces them that the three components should ideally be considered as one system for managing projects and making better decisions for corrective actions.

The first phase of this integrated model (the *alarm*) is perhaps the most difficult phase to implement in a real project environment, since it requires designing control limits with statistical tolerance limits as action thresholds (discussed in Chap. 5). The statistical methods that use tolerance limits work much better than the arbitrary rules of thumb for setting the control limits, but they can be considered too complex to implement in a practical project environment. I fear that most professional project managers will stick to the easy (but less good) rules of thumb to generate project warning signals, and for most of them the research results of statistical project control will forever remain within academia and never be given practical translations to the business world.

Unless we as academic researchers do our best to make the difficult methods a bit easier, and remove a number of basic components from the difficult methods in order to increase the implementability. What would happen if we looked for an intermediate solution that has the convenience of the simple rules of thumb, but achieves the performance of the difficult statistical project control methods? Such a middle ground is not easy to find, but it would undoubtedly open many doors to the practical world, making research not only attractive from a theoretical perspective, but also practically orienting it towards improvements in the real world. The search for methods that are both simple *and* powerful is the subject of the next chapter.

References

PMBOK (2004). *A guide to the project management body of knowledge* (3rd ed.). Newtown Square, Pa.: Project Management Institute, Inc.

Vanhoucke, M. (2010b). Using activity sensitivity and network topology information to monitor project time performance. *Omega the International Journal of Management Science, 38*, 359–370.

Vanhoucke, M. (2011). On the dynamic use of project performance and schedule risk information during project tracking. *Omega the International Journal of Management Science, 39*, 416–426.

Vanhoucke, M. (2012a). Measuring the efficiency of project control using fictitious and empirical project data. *International Journal of Project Management, 30*, 252–263.

Vanhoucke, M. (2018). *The data-driven project manager: A statistical battle against project obstacles*. New York: Apress.

Chapter 8
Analytical Project Control

This chapter revisits the statistical control methods from Chap. 5 and attempts to link their way of working with the control efficiency concept discussed in the previous chapter. Recall that the *control efficiency* was defined as the division of the *quality of actions* by the *effort of control*. It is a very general and easy-to-understand definition of efficiency that compares the impact of the project manager's decisions (quality of actions) with the effort required to achieve these decisions (effort of control). These decisions can only be made by continuously collecting, processing, and analysing project progress data to identify project problems in a timely manner. Consequently, this concept of control efficiency is a very general one that can be applied to all kinds of project control methods, from the very easy rules of thumb to the most advanced systems with all kinds of complex statistical calculations. The idea of comparing the necessary inputs (progress data, control limits, static simulation runs, etc.) with the desired outputs (warning signals and corrective actions) forms the basis of this chapter to evaluate and possibly simplify the statistical project control methodologies of Chap. 5 in order to increase their efficiency of control.

In fact, I already gave a hint at the end of the previous chapter by pointing out that most statistical project control systems are often too difficult for practical use because they require so much data and statistical analyses leading to too much effort of control and thus a control efficiency that is too low. After all, the excellent manager from the previous chapter wants a system that makes it possible to achieve very good results (high quality of actions) with relatively simple systems (low effort of control) and so every researcher must ensure that the newly proposed systems are not being too hard for practical use. As I have already concluded in the previous chapter, I fear that the statistical control methods from Chap. 5 are too complex for practical use and so we have to look for a middle ground, a new way to make the advanced project control systems simpler without sacrificing power and performance.

This new way of project control is presented in this chapter as *analytical control methods* and tries to make the statistical project control methods more user-friendly

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_8

without losing their power and ability to detect problems in a timely manner and take appropriate actions. In order to be able to properly explain these new methods, this chapter first revisits the use of statistics in project control in more detail. More specifically, the statistical *project* control methods introduced in Chap. 5 are once again critically examined and compared with other existing methods of statistical *process* control. Once the notion of statistics in project control is better understood, the chapter moves on to Sect. 8.2 and introduces a new analytical project control system that is a combination between the simple rules of thumb and the advanced statistical project management methods. Finally, this chapter will also show that the trade-off between ease of use (*effort*) and the quality of actions can be measured in different ways, very different from the simple *control efficiency* formula of the previous chapter, but with the same goal of making the project manager more efficient.

8.1 Project Control Methods (*Revisited*)

In Chap. 5, the use of *control limits* was introduced for the first time, and five different ways of controlling projects with these limits were proposed. Each of these methods aimed to build control charts with upper and lower control limits to indicate an out-of-control situation that serves as a warning signal and a trigger for actions. Four of these five methods use simulation runs to generate progress data to calculate tolerance limits as confidence intervals. The actual project control is then nothing more than a series of hypothesis tests aiming at determining whether the actual observations lie within or outside those tolerance limits. Therefore, these project control methods are no more than an application of the classical hypothesis testing theory from statistics and are therefore referred to as *statistical project control* methodologies. A fifth method consists of simple *rules of thumb*, without the use of statistics, but its use for controlling projects works in a very similar way. The only difference is that this method sets the control limits to arbitrary values based on the project controller's experience without using confidence intervals, and this easy method is therefore further referred to as *standard project control* methods. In the current chapter, a third way of controlling projects is added, referred to as *analytical project control* methods, trying to combine the best features of these two previous project control methods in a new easy *and* powerful control system. A summary of these three methods is given in Fig. 8.1 and discussed along the following paragraphs.

Standard Project Control The easiest and most straightforward way to use EVM progress data in a project control system is to set arbitrary values on the performance indicators without any statistical data analysis. These methods make use of so-called *static control limits* as thresholds on the project performance indicators based on simple rules of thumb using the project manager's experience. I argued earlier in Chap. 5 that this way of managing projects is the standard way usually used by

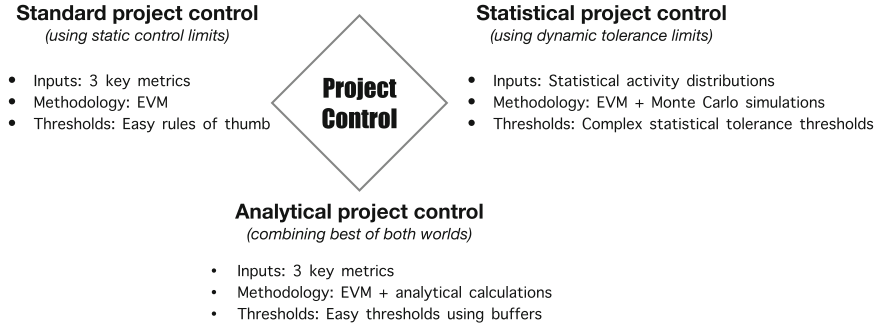


Fig. 8.1 Three types of project control methods using control limits

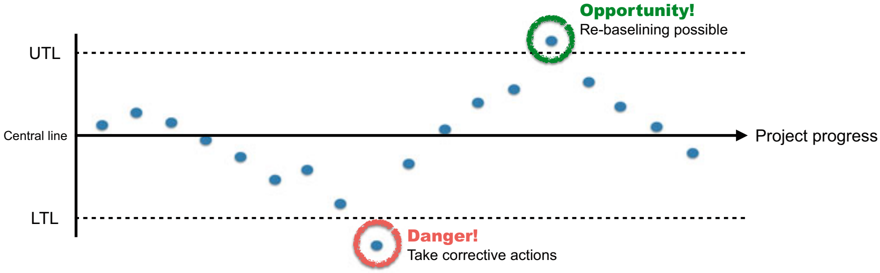


Fig. 8.2 A project control chart with static control limits

professional project managers. Since most professionals do use EVM metrics to measure the project performance but often do not take the next step to statistical data analysis, they use implicit rules of thumb to determine control limits. Using simple rules in project control makes sense but is not without danger. Rules such as *"when the performance falls below x% it is time to act"* are quickly seen as generally applicable rules, and the value for x is often set without any kind of analysis and eventually becomes a fixed value generally regarded as the standard for good project management.[1] Figure 8.2 shows an illustrative control chart with static (upper and lower) control limits. It displays the progress of a project over time (on the X-axis) and shows the upper and lower *control* or *tolerance* limits (UTL and LTL) to indicate if and when a drill-down is required. When the performance indicators for time and costs (e.g. SPI or SPI(t) for time control and CPI for cost control) exceed these thresholds, a warning is given that the project is at risk (under LTL) or that the project is progressing better than expected (above UTL). In either case, the out-of-

[1] When I ask practitioners what the generally tolerable deviation of project progress from the project plan is, I often hear percentages as multiples of 5% (some swear by 10% as the maximum deviation, and others say their projects are sometimes allowed to have up to 20% or 25% deviation before actions are taken). I do not know why it always has to be this way, but I suspect it is because of our way of counting, using our five fingers and counting the number of hands.

control signal is an indication that the current project performance is significantly different from the expected performance (given by the central line) and that further research is needed to solve problems or seize opportunities.

Of course, not every project manager makes very explicit use of such control charts and often very implicitly controls the project with arbitrary threshold values in mind. For such implicit control chart system, these control limits are mostly set as fixed values (horizontal lines), but they can vary, as described earlier in Chap. 5 to the ascending and descending values along the project progress. Despite these possible variations, the use of this method is still limited to the arbitrary determination of threshold values for the control limits and is never based on a proper statistical analysis, as is the case with the statistical project control systems discussed next.

The best way to understand the difference between these two classes of project control methods (*standard* versus *statistical*) is to compare Fig. 8.2 with Fig. 5.3 of Chap. 5. This last figure shows a control chart where the control limits vary according to the stage of the project and such control limits can never be determined on the basis of simple rules of thumb, but instead require a thorough statistical analysis that forms the basis of the statistical control methods.

Statistical Project Control (*Basic*) The statistical project control methods use static simulation runs to determine control limits in advance. Based on a predefined *desired state of progress*, these simulations will generate normal variation to model that the future real project execution is allowed to have a certain degree of variation. As long as the real project progress is within these limits, the project does not appear to have a significant problem, and no action is necessary. The specific design of the control limits has already been discussed in detail in Chap. 5 and will therefore not be repeated here. In the previous chapters, I took the liberty of using the words *tolerance limits* and *control limits* interchangeably, but I should have been more careful. While they both serve as action thresholds to trigger actions, their differences are important for understanding how the project progress data can be used to construct these control charts. In the following paragraphs, I will therefore explain the important difference between control limits and tolerance limits and illustrate this difference using two completely different ways of data generation to track the project. The difference stems from the use of statistics to monitor and control *processes* (instead of projects), and the term "*statistical project control*" is therefore often used for two completely separate ways of monitoring projects.

Statistical project control
statistical tolerance limits $\neq$ statistical control limits

Statistical Project Control with Tolerance Limits (SPC-TL) This first class of control methods consists of the methods discussed so far in this book. These methods use control charts that are created in advance (prior to the project start) based on the predefined distributions (to model the desired state of progress) and the static simulation runs (to generate the fictitious progress data). This means that the central line and the upper and lower control limits are known in advance, calculated as tolerance limits on the generated data. At the actual start of the project, the charts will remain immutable and are used to calculate the difference between allowable variation (within the thresholds) and unacceptable (outside the thresholds) variation in the project progress. Whenever the project enters a new phase, the project manager must measure the current status of the project, and this is then added into the control chart as a hypothesis test to see whether actions are necessary or not. Since these control charts consist of tolerance limits based on fully pre-generated data, this way of project monitoring is referred to as statistical project control with tolerance limits.

Statistical Process Control with Control Limits (SPC-CL) The second class of statistical project control methods has not been discussed earlier in this book and is very similar to the classic view of statistical *process* control that is often used for monitoring production processes. The fundamental difference from the previous SPC-TL method is that the SPC-CL method assumes that the data will gradually become available over time. This means that no desired state of progress must be defined and no static simulation runs will be required, since the central line and the lower and upper control limits are calculated based on the gradually incoming data points during project execution. The SPC-CL methods measure the deviation from a *normal* project progress defined by the available data points and do not require an *acceptable* project progress defined in advance. It should also be noted that these classical statistical control methods often reject the idea that control charts are used as a form of hypothesis testing. Since these methods are based on the same underlying assumption of statistical process control, they are further referred to as *statistical process control with control limits*.

The use of SPC-CL methods will not be discussed further in this book, but I would nevertheless like to give a brief overview in the following paragraphs of a number of studies that have proposed these methods, not only to draw the readers' attention to their existence but also to show that their way of working is fundamentally different from the SPC-TL methods used throughout this book. A more detailed discussion is given in Vanhoucke (2019). To the best of my knowledge, not many statistical process control methods have been presented in the academic literature that perfectly fit into the SPC-CL framework. These methods have their origins in the use of traditional *statistical process control* (SPC) methods from manufacturing. Indeed, these traditional methods assume that they are trying to control an *infinite* process, and they measure deviations from *normal* progress as defined by the observed data. Such a model focuses on a process that "*speaks for itself*", which means that the control charts are constructed along the project progress. It is as if the control charts are listening to the *voice of the process*, which means that they gradually become more and more reliable as more and more progress data become available. Since a process is viewed as a continuous, never-ending process, its control charts can actually be trusted on the data generated after a sufficiently long run-in period. But as projects are defined as *unique* efforts (rather than continuous processes), they often have no time enough to "listen to the voice" during their progress since not enough data will be generated. As a matter of fact,

Table 8.1 Three methods to construct control charts (*the voice of the project*)

Progress data (SPC-CL)	Historical data (SPC-CL)	Simulated data (SPC-TL)
No data	Some data	Enough data[b,c]
Same project	Defining similarity[a]	Same project
Lipke and Vaughn (2000)	Bauch and Chung (2001)	Colin and Vanhoucke (2014)
Wang et al. (2006)	Leu and Lin (2008)	Colin et al. (2015)
Aliverdi et al. (2013)		Vanhoucke and Colin (2016)

[a] Reference class forecasting can help to define the similarity between projects (Chap. 9).
[b] Simulated data for artificial project progress can be obtained with project progress models (Chap. 12).
[c] Realistic distributions to model project uncertainty can be obtained with calibration procedures (Chap. 14).

due to the uniqueness, collecting data on the progress of the project is difficult because there is no data at all at the beginning, and as the project progresses, the data are often unreliable at first. Thus, for a project, one must wait too long for sufficient or reliable data before the control process can be considered more or less ongoing (except if the project is extremely large and has a very long duration, then one could assume that the project resembles an ongoing process). Therefore, researchers have proposed the use of statistical process control methods for project control (SPC-CL) that are the middle ground between the classical SPC view (ongoing infinite process, no data available at the beginning) and the unique project view modelled by the SPC-TL approach (unique project, all data simulated in advance). An overview of the different methods, along with some references to original studies, is given in Table 8.1. The table divides the control methods into three columns, depending on where they get their data from to construct the control limits (or tolerance limits) for the control charts (the voice of the project).

- The first class retrieves the data in the traditional way while the process (in this case, the project) is running. This control method follows the classical view of statistical process control and assumes that the data come from the progression of the project and become more and more reliable as the project progresses. However, as mentioned earlier, these methods are often practically unusable because there is insufficient data present in the early stages of the project, and therefore they cannot construct reliable control limits for the control charts.
- A second method attempts to solve this flaw by providing sufficient data at the beginning of the control process. More precisely, this method suggests extracting data from the early stages of *similar* past projects so that control limits can be determined very early in an accurate manner. This method is still considered a classical SPC-CL method because no tolerance limits are calculated before the start of the project, but these are determined as the project progresses. Despite the fact that this approach has the advantage of having sufficient data, the challenge lies mainly in defining project *similarity*, something that is not at all obvious. A

subsequent chapter will delve deeper into how historical data can be obtained and how similarity between projects can be measured.

- The last method belongs to the class of SPC-TL methods that use simulated data to construct the tolerance limits of the control chart. This method does not need to use (but does not exclude) historical data as it relies on the static simulation runs on a set of artificial projects. Moreover, this method can easily generate a lot of data since the static simulation runs are made to obtain these data in a very easy way. Of course, the challenge of this method lies in generating these data as close as possible to the expected real execution of the project. This requires the use of probability distributions that reflect the variability in the project, and these are preferably not set completely randomly. The best way to set these to realistic values is to analyse historical data, and a procedure that uses such data to determine distributions will be proposed in Chap. 14. Moreover, it is also necessary that the static simulation experiments are performed very accurately so that the artificial project progression is approximated as realistically as possible. For this purpose, one of the progress models that will be presented in Chap. 12 can be used.

As I mentioned earlier, I will not return to the SPC-CL methods in this book and will only use the SPC-TL methods. This is not to say that I want to rule out one method or another. Diversity in statistical methods to measure project progress is always a good thing. However, because of these fundamental differences, comparing and benchmarking these different methods is not an easy task, and I prefer to stay in the familiar realm of tolerance limits. In a study presented in Colin and Vanhoucke (2015b), Jeroen and I presented a possible framework to fairly compare these different methods and test their ability to detect project problems in early stages of control. The study showed that there is no agreed framework on how to validate the strength of the different methods, nor is there any consensus on how the required normality (or transformations to normality) is performed on the project data. Therefore, we presented a framework based on the statistical tolerance approach (SPC-TL) and concluded that the SPC-CL method of Leu and Lin (2008) is closest to the results obtained with the SPC-TL methods. For more details, I refer the readers to the original study. In the next section I will briefly discuss a number of extensions of the SPC-TL to show that a lot of further research can be done in this area by using more complex statistical techniques.

Extended Statistical Methods (*Extended*) I have mentioned before that the SPC-TL methods are probably too difficult for many project managers and therefore they are not going to be implemented very soon. However, these methods do not use the most advanced statistical techniques (*hypothesis testing*) and the control charts are limited to X charts (to measure the average performance of a particular project period) and R charts (to measure the performance of successive periods). However, it can all be made much more complex, and instead of measuring performance on control charts using a single metric (e.g., SPI or CPI), the generated data can first be manipulated by using more advanced statistical methods. These new advanced metrics can then be used as new schedule control metrics to construct tolerance limits, as discussed earlier. A brief summary of this so-called *multivariate statistical project control* approach is given as follows.

Extension 1: Multivariate Statistics Colin et al. (2015) present a multivariate project control methodology using as much data as possible coming from the EVM system. More precisely, rather than only focusing on the *schedule performance index* (SPI) or *cost performance index* (CPI), this new control method also incorporates other EVM metrics such as the schedule variance (SV) and cost variance (CV) and can possibly be extended to metrics such as the p-factor (Lipke, 2004), the to-complete-index (Fleming & Koppelman, 2010), and many more EVM-related metrics if necessary. More formally, the project progress is now represented by a vector x of multiple EVM metrics measured along the lifetime of the project (i.e., at each percentage completion). The underlying idea is that incorporating more performance data will provide a more accurate view on the real performance of the project and will more accurately detect real problems in the project progress. However, the authors also warn that more data might be dangerous and might lead to problems such as *data overload*, *redundancy*, and even *noise*, which might decrease the reliability of this advanced control system. For these reasons, they propose to use the *principal component analysis* (PCA) methodology to reduce the dimensionality of data with many potentially interrelated variables. In doing so, they reduce the huge amount of metrics to a smaller set of so-called *principal components* that contain much of the information necessary for controlling the project. It is known that PCA is designed to reduce the dimensionality of a problem in a structured way with a minimal loss of valuable information. In the case of controlling projects, this means that the periodic observations of the current project performance can be projected onto a new set of coordinate axes, thereby removing the previously mentioned problem of redundancy and noise. This means that the performance metrics (SPI, CPI, SV, CV, etc.) will be transformed into new schedule metrics, unknown to any EVM system. The authors present two new schedule control metrics, known as the *Hotelling's T2 statistic* and the *squared prediction error* (SPE) (Hotelling, 1951), which now serve as new modified schedule metrics that can be used for schedule control using tolerance limits in a similar way as explained before. Consequently, this extended approach is called a *multivariate* approach to denote that multiple EVM performance metrics are used as inputs for schedule control in contrast to the SPC-TL methodology, which is considered to be a *univariate* approach (since only a single performance metric is plotted on the X and R control charts). Computational experiments have shown that the multivariate approach has a superior ability to detect project problems compared to the univariate approach, but this improvement comes at the price of extra advanced statistical analyses needed to calculate the new schedule metrics.

Extension 2: Multivariate Regression Thanks to the improved performance of the multivariate techniques for project control, Vanhoucke and Colin (2016) have extended this approach even further using multivariate regression methods. The authors compare four extensions of the original PCA model described earlier. The first two extensions embed the *principal component analysis in a regression* *model* (PCR) or extend this PCR approach with an additional *matrix decomposition* (PCR-PC). Two other extensions consider a *kernel variant* for PCR (kPCR) and a *partial least squared regression* (PLSR). It goes without saying that these extensions significantly increase the complexity of the project management methodologies and may no longer be accessible methods for professional users. However, the research was aimed at measuring the potential of these advanced statistical methods to predict the expected outcome of a project and to get the most out of an EVM system for better project control. A comprehensive overview of these multivariate regression methods and the parameters used for these methods is beyond the scope of this book, and the readers are referred to the original manuscripts for more details.

8.2 Best of Both Worlds

I hope that the readers have found the courage to work their way through the preceding paragraphs and have taken the opportunity to thoroughly review and compare the various statistical methods. The purpose of the previous section was mainly to show that such advanced methods are the subject of many research projects, but that their day-to-day use is not very likely. The difficulty of such methods was measured in the previous chapter by means of the effort of control. As we have seen, the concept of *control efficiency* is a simple concept to find a balance between the effort of control and the quality of actions, and it is a way to find the right balance between ease of use of control methods and the accuracy of their warning signals for taking actions. As mentioned, project control methodologies need to strike a balance between *not too easy* and *not too complex*, which is of course also applicable outside the field of project control. Finding the right balance applies in many facets of life, as Albert Einstein argued:

> Everything should be made as simple as possible, but not simpler.

I particularly like the *"but not simpler"* in this quote, since otherwise, proposing easy methods would be pretty straightforward. However, when simplifying advanced methods, the challenge lies in doing it in such a way that they do not lose much of their performance.

One of the first serious research attempts to turn academic project control research into a new *easy* method without reducing the control efficiency started in 2014 with a Business Engineering student working on a Master's thesis on *statistical project control*. Annelies Martens was applying the SPC-TL methods to practical projects, and during her graduation as a Master in Business Engineering she received a prize from the Project Management Institute (PMI Belgium) for her Master's thesis. The ultimate goal of her thesis was to test the new statistical control tools that I discussed in Chap. 5 (and briefly summarised in the previous section). Her research revealed that not many project managers were ready to use static Monte Carlo simulation runs on their projects to calculate tolerance limits. So she decided to devote her research time to developing an alternative project control method that should be just as good as the statistical methods, but much simpler to use. And so she started a PhD in our research department and she received—4 years later—a second prize from PMI Belgium for the thesis entitled *"Buffer management methods for project control"*. The basic idea of her research project was very simple (it always is, in hindsight) as she went looking for the right balance between simplicity and complexity, and she came up with the proposal to work out so-called *analytical project control* methods. This new method should find a middle ground between the easy *standard control methods* and the complex *statistical project control* methods by simplifying the processing of the progress data. Instead of relying on the *static simulations* to generate data, she wanted to replace them with *simple analytic calculations*, as illustrated in Fig. 8.1. The underlying idea of using analytical tolerance limits is that they only require the construction of the project's baseline schedule (reference point) and the calculation of only a few basic EVM metrics (SPI, SPI(t), or CPI) to monitor project progress and that is it. No additional data (historical data or simulated data) nor advanced statistical tools (such as Monte Carlo simulations or regression models) are required, making these methods much easier to use (i.e., less control effort). Consequently, the analytical models are very similar to the rules of thumb of the standard control method but differ in the way the control limits are constructed. Rather than just setting thresholds using arbitrary values, some simple analytical calculations will be used to determine the correct values for the control charts.

The basis of the analytical project control method lies in the simple principle of using a *buffer* to protect the project from unacceptable delays. A project buffer adds an extra safety time margin to the planned duration (PD) of the baseline schedule and acts as a simple project control tool to visually see whether delays are still acceptable or not. Figure 8.3 shows such a project buffer divided into three zones. Any delay of the project in the *green* zone is acceptable and indicates only minor deviations from the basic schedule. The *orange* zone indicates that delays are increasing and warns the project manager to be careful and not to relax too much. Finally, the *red* zone indicates that further delays could jeopardise the project and potentially consume the total project buffer and thus lead to an unacceptable delay if corrective actions are not taken. In that regard, a project buffer is nothing but a visual representation of a control chart, and when the project status enters the red zone, it is very much like exceeding the statistical tolerance limits as a call to action. Project buffers are widely used in the academic literature (and commercial software tools) but are often

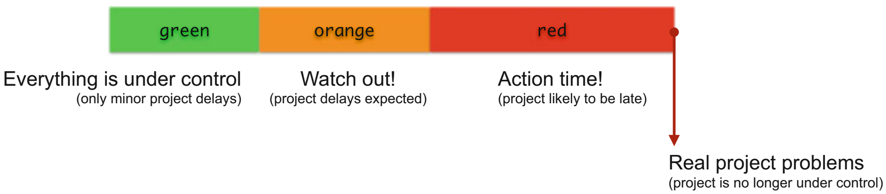


Fig. 8.3 The principle of a project buffer: detecting problems and taking actions

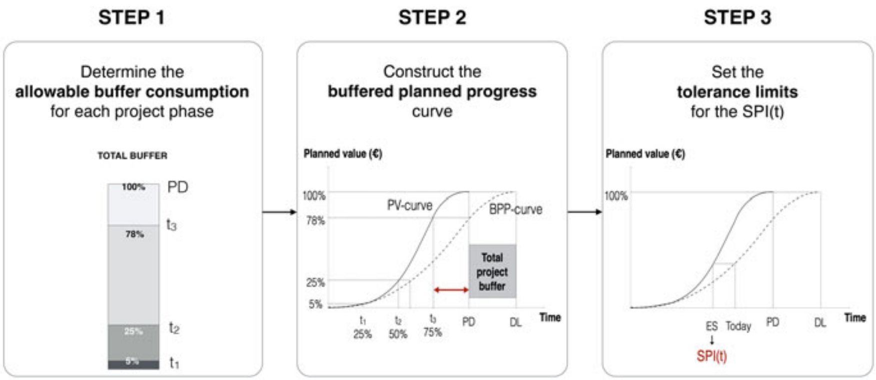


Fig. 8.4 The three steps of analytical project control

little more than dashboards to visually monitor buffer penetration (Goldratt, 1997; Hu et al., 2015). However, the analytical project control methods go one step further and define the different zones in the buffer based on project-specific characteristics using analytical calculations (instead of Monte Carlo simulations). The general approach consists of three distinct phases, which are visually summarised in Fig. 8.4 and further discussed along the following lines.

Step 1: Determine Allowable Buffer Consumption for Each Project Phase
The first step of the proposed approach consists of determining the *allowable buffer consumption* for each stage of the project, such that the expected buffer consumption at project completion is less than or equal to 100%. The allowable buffer consumption defines the maximum amount of the project buffer that is allowed to be consumed at each project stage. Traditionally, the allowable buffer consumption is set linearly with the project progress, i.e., at $x\%$ of the project completion, and the allowable buffer consumption is set at $x\%$ of the total buffer size, as shown in Fig. 8.5. The buffer is shown vertically and displays that the percentage completion of the project (horizontal axis of the graph) grows linearly with the expected buffer consumption. However, this approach does not consider that the amount of work during the project life can vary along the different completion stages and does not take any project-specific characteristics into account.

Therefore, Annelies has presented an *allowable buffer assignment* procedure to allocate a non-linear buffer consumption over the different project phases using EVM cost metrics (published in Martens and Vanhoucke (2017a)). More specifically, the buffer is allocated proportionally with the *planned value* of each phase in the project, which is expressed in a monetary unit as the cumulative increase in the total cost of all activities of the baseline schedule. When this allowable buffer consumption is known at each phase of the project, it can be used as a threshold denoting that overconsumption at any time in the project will likely result in a project duration overrun. Hence, the concept of *allowable buffer consumption* is nothing more than a tolerance limit but is now based on

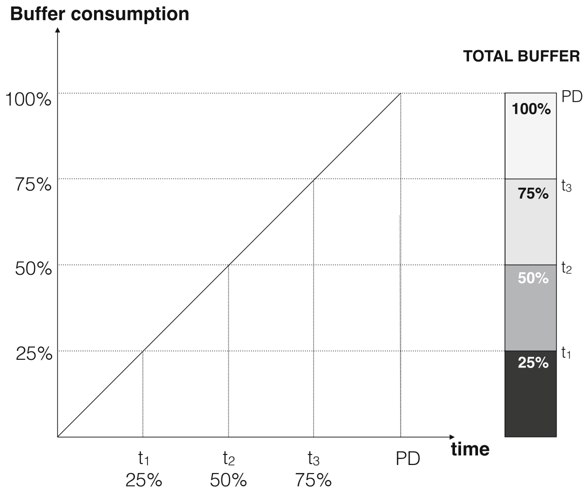


Fig. 8.5 A linear allowable buffer consumption

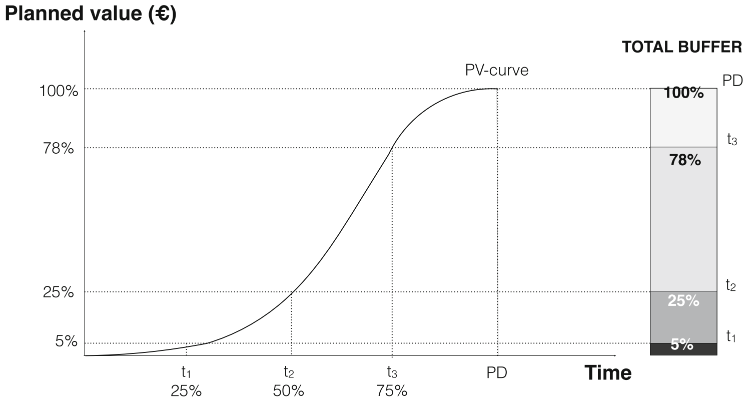


Fig. 8.6 A non-linear allowable buffer consumption

simple analytical calculations (i.e., the planned value curve of the baseline schedule) instead of Monte Carlo simulations.

The graph of Fig. 8.6 depicts the allowable non-linear buffer consumption at each time t during the project duration. As shown in this figure, at time t_1, 25% of the planned project duration (PD) has passed, but only 5% of the total budget at completion (BAC)[2] is planned to be earned at that time. As a result, only 5% of the total project buffer should be assigned to t_1, which is much lower than the 25%

[2] Recall that the construction of the project baseline schedule results in two key metrics, known as the planned duration (PD) and the budget at completion (BAC or total planned cost) as briefly discussed in Sect. 4.1.

that normally would be assigned in a linear buffer method. This simple calculation can be made at any time in the project plan to ultimately allocate the entire buffer at the end of the project. More specifically, for each time t (from $t = 0, \ldots,$ PD), the allowable buffer consumption ab_t can be calculated as shown in the following equation:

$$ab_t = pb \times \%PV_t \tag{8.1}$$

In this equation, pb is used to refer to the total size of the buffer and $\%PV_t$ is equal to $\frac{PV_t}{BAC}$. The total buffer size pb must be defined as the difference between the planned duration (PD) of the project's baseline schedule and the maximum allowable project duration, possibly defined by the deadline (DL) promised to the customer. The ab_t formula does nothing but distributes the total buffer pb proportionally over the different periods as the proportion of the planned value at time t (PV_t) over the maximum value of the planned value curve (which is equal to BAC).

Step 2: Construct the Buffered Planned Progress

With the obtained values for the allowable buffer consumption for each time t of the project's base schedule, the new approach must now determine when each phase of the project must be completed at the latest. Therefore, in the second step, the *buffered planned progress* (BPP) is determined for each phase of the project, which represents the planned progress of a project taking into account the allowed buffer consumption in each phase. Therefore, the allowed buffer consumption must be added to the project phase, as shown in Fig. 8.7. Since the ab_t reflects the project's acceptable delay at time t during the project duration, the BPP for each time t can be determined analytically by adding ab_t to time t, as follows:

$$BPP_t = t + ab_t \tag{8.2}$$

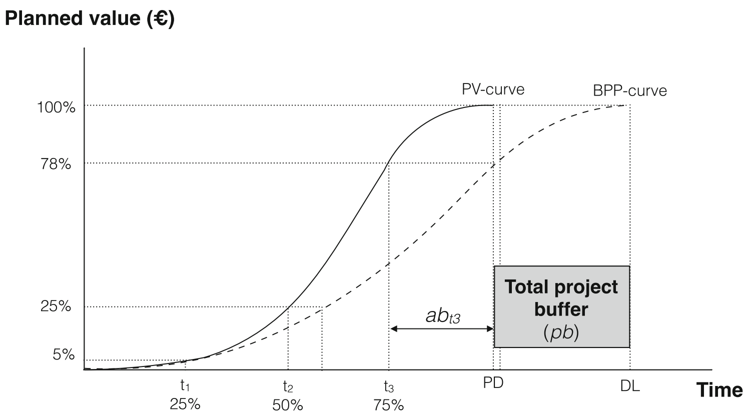


Fig. 8.7 The buffered planned progress curve

Step 3: Set the Tolerance Limits for the SPI(t)
The BPP curve now represents for each stage of the project the allowable project delay, which can be used as control limits that cannot be exceeded. More specifically, it is very easy to calculate what the minimum value of the schedule performance metrics (e.g., SPI(t)) must be to guarantee that the BPP curve is not exceeded at each time t. These minimum values for the schedule performance metrics can now act as control limits and serve, just like all previously discussed control methods, as warning signals for actions. During each project progress monitoring period, the SPI(t) values are monitored and compared to these minimum required threshold values for these schedule performance metrics. When the observed SPI(t) values fall below the minimum value at any given time t, this is an indication that the buffer consumption is too high, possibly indicating a high probability of large project delays. In that case, it is—as always—time for actions.

Practical Implementation
The three-step procedure to determine the non-linear buffer consumption as control limits for project control can of course be implemented in different ways. The previously discussed method uses EVM cost data (represented by the PV curve) to determine the allowable buffer consumption, but this curve can be replaced by any other non-linear curve describing the course of the project plan, as explained hereunder:

Cost Data As argued earlier, the analytical project control method differs from traditional buffering methods since the *buffered planned progress* is calculated using EVM cost metrics (the planned value curve) and does not simply rely on linear or arbitrary buffer consumption rules as is often the case. Computational experiments have shown that these analytical control methods using non-linear buffers show a slightly improved performance for project control compared to the standard project control method (rules of thumb) for serial projects. However, the analytical buffer method improves the efficiency of project control a lot for projects with a more parallel structured network. Since it is known that the classic EVM methods traditionally perform poor on these parallel projects, the analytical project control method can therefore be considered as a good alternative for schedule control when the project is close to a completely parallel network.

Resource Data Given these improvements for parallel projects, we could not resist the temptation to consider more project-specific characteristics than just cost data. The most logical (but also challenging) step was to integrate the use of scarce project resources into the control method, so that the BPP curve can be constructed from *resource data* instead of purely from cost data. While many algorithms exist to construct baseline schedules with scarce renewable resources,[3] it is surprising that the scarce availability of resources has not been taken into account when studying EVM methodologies for project control. A new analytical project control method incorporates the *shiftability reservation rate* (SRR) into the allowable buffer consumption curve defined as the division of the absolute shiftability with the maximal shiftability. The *absolute shiftability* is based on right-shifting the activities in the project towards the expected deadline, without violating the project network (precedences between activities) and the limited availability of resources (resource constraints). The *maximal shiftability* is quite similar but ignores the precedence relations between activities and only takes the limited availability of resources into account when shifting activities. Since the SRR is the division of these two right-shifts, the metric is expressed as a percentage between 0 and 1. The new method sizes the buffer in a similar way as the cost data method but defines the allowable buffer consumption based on the SRR values of the project.

Risk Data It is in the nature of academic research to never give up, and any improvement is a sign to look for more. After the observation that adding more project-specific features helps improving the accuracy of the analytical control methods, we decided to extend the buffering approach once again, now incorporating a third source of data. It is quite obvious to set the allowable buffer consumption at each phase of the project according to the expected risk of this project phase. More specifically, when there is a much greater expected risk at certain project phases, it seems logical to allow more buffer consumption there and to slightly reduce the permitted consumption during safer project periods. The risk curve is constructed as the cumulative increase of the *schedule sensitivity index* (SSI) values for each activity obtained from a traditional *schedule risk analysis*, as briefly discussed in Sect. 3.4 of Chap. 3. In periods where many activities with a high value for the SSI are planned, the curve will therefore rise much steeper, so that the allowable buffer consumption is set to larger values. All other steps are identical to the previous methods, and this third approach results in only a slightly different version to set control limits using analytical calculations.

In our study published in Martens and Vanhoucke (2018), we have compared the three analytical project control methods on 93 real-life projects of Batselier and Vanhoucke (2015a). The computational experiments have shown that all analytical project control methods perform better than the rules of thumb used by the standard project control method and almost perform as well as the complex statistical project control methods. The method with resource data was considered as the most promising method to increase the control efficiency of the project manager and is therefore recommended for further use. Table 8.2 provides an overview of the three different implementations of the analytical project control method using three different non-linear curves with different sources of project data (cost, resource, and

[3] As you may recall, this scheduling problem is known as the *resource-constrained project scheduling problem* as discussed earlier in Chap. 6.

Table 8.2 A comparison of analytical control methods

Type of limits	#	Project level and Tracking level (DP, PO, E, R) and (SD, SR, SE, SR)	Reference
Linear time limits	93	For regular cost curves, the linear time limits perform reasonably well	Colin and Vanhoucke (2015a)
Cost limits	93	For irregular cost curves, the cost limits outperform the linear time limits significantly	Martens and Vanhoucke (2017a)
Resource limits	21	The resource limits perform much better than the linear time and cost limits	Martens and Vanhoucke (2017b)
Risk limits	32	The risk limits perform slightly better than the linear time and cost limits	Martens and Vanhoucke (2018)

risk data). The table also displays results for the traditional buffering methods using a linear buffer consumption. For each method, the number of tested projects is given (#) and a reference to the original study is displayed in the last column. The middle columns display two times four quality metrics that assess the performance of each control method, which can be divided into two classes (project level and tracking level metrics). No specific values for the quality metrics are reported, but a general conclusion is drawn for each method. The interpretation and calculation of the two times four quality metrics are the subject of the next section.

8.3 The Signal (*Not the Noise*)

The performance of any project control system with tolerance limits can and should be assessed by the reliability of its warning signals, as well as by its ability to use these signals to detect and solve the problems of the project in progress. A reliable project control system should warn the project manager only when real project problems occur and should leave the project manager relaxed when no special problem occurs. This ability to generate true warning signals requires accurate forecasts of the impact of problems, and should be measured by *statistical quality metrics*, as will be discussed in this section. In his book *"The signal and the noise: Why so many predictions fail — but some don't"*, Nate Silver (2012) describes the accuracy of a forecasting system as follows:

> The signal is the truth.
> The noise is what distracts us from the truth.

The search for *true signals* for project control is not a new theme in this book and has been discussed in many different ways in previous chapters. Chapter 4 introduced the *accuracy* of project duration predictions using three different forecasting methods. These methods provided time predictions for projects in progress and thus did not use control charts with warning signals. Nevertheless, it was implicitly assumed that a higher accuracy would lead to a better control system, and so the accuracy was actually an indirect metric to evaluate the quality of predictions. In Chap. 7, the concept of *control efficiency* was introduced as a trade-off between *effort of control* and *quality of actions*. Although this concept does not really use very precisely defined statistical measures, it comes a lot closer to how warning signals should be evaluated. Note that the *effort of control* is measured by the number of times the project manager drills down in the work breakdown structure to look for project problems (generated by the warning signals). Because it is known that these warning signals are not always correct (*false* signals), not every drill-down leads to finding a problem. Sometimes there is simply a signal without a real problem occurring, resulting in a waste of effort. The *quality of actions* is measured by the impact of the corrective action taken by the project manager when a problem was found. Thus, if the warning signal did not result in any detection of a real problem, the effort was high (waste of time) and the impact of actions low or non-existent (because there was no problem), so the control efficiency will be very low. Consequently, this *control efficiency* is just another way of measuring the performance of a project control system, but it is more encompassing than the *accuracy* metric of Chap. 7. However, it still does not use any *statistical* measurement as will be discussed in this section. The use of statistical performance measures to evaluate the warning signals was only introduced for the statistical project control systems of Chap. 5. In this chapter, the *probability of overreaction* (type I error) and *detection performance* (type II error) were presented and combined in the *area under the curve* metric. These statistical measures will be reviewed and refined in the current chapter by applying them to two levels of the project progress, as discussed next.

The statistical signal metrics proposed in this chapter will focus on the *quality* of the signals (i.e., the reliability to generate true warning signals) and on the *intensity* of the signals (i.e., the number of signals generated during the progress of the project) and will no longer include the corrective actions as was the case for the *control efficiency* concept. The two types of metrics—quality and intensity—correspond to two different levels of project progress and are both based on known metrics in Bayesian statistics. The first set of metrics measures the quality of the signals generated by the project control system at the *project level*. More precisely, the signal quality is expressed by the ability of the generated warning signals to give a correct signal or not (as a binary evaluation). The previously discussed *detection performance* and *probability of overreactions* belong to this class of signal quality metrics, but two additional metrics, known as *efficiency* and *reliability* for project control, are also included in Table 8.3. These quality metrics measure the performance of the control system at the project level, which means that they are not evaluating each individual tracking period, but only the signal quality as a whole after the project is finished. A second set of signal metrics evaluates the performance of a project control system at the *tracking period level*, i.e., at each period at which the performance of an ongoing project is measured. These metrics

Table 8.3 Overview of quality metrics for project control

Statistical signal metrics		Range	Desirable state
Signal quality *(Project level)*			
Detection performance	Probability of warning signals for late projects	[0, 1]	High
Probability of overreactions	Probability of encountering warning signals for timely projects	[0, 1]	Low
Efficiency	Probability that the project deadline is exceeded when a warning signal is generated	[0, 1]	High
Reliability	Probability that the project is finished timely when no warning signals are generated	[0, 1]	High
Signal intensity *(Tracking period level)*			
Signal density	Average number of signals generated for late projects	$[0, K]$	High
Signal redundancy	Average number of signals generated for timely projects	$[0, K]$	Low
Signal efficiency	Proportion of correct warning signals	[0, 1]	High
Signal reliability	Proportion of correct absence of signals	[0, 1]	High

measure the *signal intensity* by counting the number of signals generated over the entire project progress. Instead of just measuring whether a signal is right or wrong (*signal quality*), it is also important to know *when* and *how often* these true or false signals are generated as the project progresses. This new set of metrics consists of metrics such as the *signal density*, the *signal redundancy*, the *signal efficiency*, and the *signal reliability*.

The summary in Table 8.3 shows the eight signal metrics and the range for each metric. All *signal quality* metrics are expressed as a percentage (between 0 and 1), but two of the four *signal intensity* metrics are expressed as a value between 0 and K with K being the number of tracking periods during project progress. The desired status of each metric indicates the ideal value (high or low) for determining whether the project management system is of good quality. Each metric is explained in detail in the following sections, and due to its technical nature, a summary of the abbreviations can be found in Table 8.4. I would like to warn the readers who suffer from a math phobia and are afraid of statistics and numbers and advise them to drink a strong cup of coffee (black, no sugar) before proceeding with this section. If you do not have coffee nearby, consider taking a break or go immediately to the next chapter.

Set 1: Quality of signals (*measuring the performance at the project level*)

The quality metrics of a control system with tolerance limits should consist of two aspects at the project level. First of all, the tolerance limits must be able to detect problems that ultimately have a high probability to lead to a project duration exceeding the deadline (true signals). Second, these limits should not indicate that a project is expected to exceed the deadline, while the project will eventually be completed on time (false signals). In a statistical context, measuring

Table 8.4 Symbols and abbreviations used for the signal metrics

Abbreviations for the signal metrics	
$\delta_{\mathcal{N}}$	Project deadline
RD	Real duration of the project
S	A generated signal
n	Number of simulation runs
n_L	Number of late projects in the simulation
n_O	Number of timely completed projects in the simulation
K	Number of tracking periods

Fig. 8.8 Hypothesis testing framework

	H_0: on time	H_1: late	
signal	false positive (type I error)	true positive	**positive predictive value** (efficiency)
no signal	true negative	false negative (type II error)	**negative predictive value** (reliability)
	false positive rate (probability of overreactions)	**true positive rate** (detection performance)	

the performance of tolerance limits at the project level can be represented as a statistical hypothesis test with the null hypothesis H_0 specified as *"the project finishes on time"* (which is defined as finishing before or at the deadline) and an alternative hypothesis H_a specified as *"the project is late"*. The null hypothesis is rejected when a performed test has a positive outcome, while a negative outcome implies that the null hypothesis cannot be rejected. In a project control context, the generation of a warning signal corresponds to a positive test outcome, while the lack of warning signals indicates a negative test outcome. In Fig. 8.8, the hypothesis testing framework used in the remainder of this section is outlined. The framework consists of four results (true positives, false positives, true negatives, and false negatives) and four statistical measures of performance (true positive rate, false positive rate, positive predictive value, and negative predictive value), which correspond to the four signal quality metrics of Table 8.3 (detection performance, probability of overreactions, efficiency, and reliability).

First, *true positives* are positive outcomes that correctly reject the null hypothesis. *False positives* are positive outcomes that incorrectly reject the null hypothesis and are often referred to as type I errors. Similarly, *true negatives* occur when the null hypothesis is correctly not rejected. Finally, *false negatives* incorrectly fail to reject the null hypothesis and are considered type II errors. Consequently, the quality of the warning signals generated by the tolerance limits can be classified into two groups. A distinction is made between *correct warning signals* and *false warning signals*. The correct warning signals are generated during the project life cycle of a project that eventually exceeds its deadline and can hence be considered true positives. On the contrary, warning signals generated for projects that eventually finish on time are referred to as false warning signals and can be considered false positives.

Moreover, the *true positive rate* and *false positive rate* measure the proportion of positives (i.e., late projects) and negatives (i.e., projects on time) that are identified as positive (i.e., warning signals are generated). Hence, these rates represent the conditional probabilities of receiving warning signals given that a project is late or on time, respectively. Furthermore, the *positive predictive value* and *negative predictive value* are the proportions of positive results (i.e., warning signals are generated) or negative results (i.e., no warning signals generated) that are true positives (i.e., for late projects) or true negatives (i.e., timely projects). The predictive values are thus the conditional probabilities of finishing late or on time given the presence or absence of warning signals. Moreover, these conditional probabilities are the inverse of the true positive and false positive rate. So much for the statistical theory which, in my humble opinion, is not very complex but still confuses me all the time. Let me therefore translate this theory into the four signal quality metrics of Table 8.3.

As mentioned before, the signal quality metrics examine whether or not warning signals are generated, without considering the precise number of generated signals. The *detection performance* indicates the probability that warning signals are generated for late projects, which is equivalent to the true positive rate. It is calculated as the ratio of the sum of the late fictitious project executions that generated a warning signal to the number of late executions in the set of simulated fictitious executions, as shown by Eq. (8.3):

$$\mathrm{P}[S|RD > \delta_{\mathcal{N}}] = \frac{\sum_{i=1}^{n} \mathbb{1}_i(RD > \delta_{\mathcal{N}})\mathbb{1}_i(S_i)}{n_L} \tag{8.3}$$

with $n_L = \sum_{i=1}^{n} \mathbb{1}_i(RD > \delta_{\mathcal{N}})$ and S_i the logical disjunction over all review periods of run i:

$$S_i = \bigvee_{k=1}^{K} S_{ik} \tag{8.4}$$

with $S_{ik} = 1$ if a signal is generated at review period k of run i. As a result, Eq. (8.4) entails that S_i is 1 if one or more review periods of run i generate a warning signal.

Furthermore, the *probability of overreactions* reflects the probability that warning signals are generated for projects that are completed before the deadline (i.e., the false positive rate). It is defined as the ratio of the sum of the timely fictitious project executions that generated a warning signal to the number of timely executions in the set of simulated fictitious executions (Eq. (8.5)).

$$\mathrm{P}[S|RD \leq \delta_{\mathcal{N}}] = \frac{\sum_{i=1}^{n} \mathbb{1}_i(RD \leq \delta_{\mathcal{N}})\mathbb{1}_i(S_i)}{n_O} \tag{8.5}$$

with $n_O = \sum_{i=1}^{n} \mathbb{1}_i(RD \leq \delta_{\mathcal{N}})$.

In a real-life context, the final outcome of a project is unknown to the project manager during execution. However, the presence or absence of warning signals may provide valuable information to the project manager, based on which it can decided whether to take corrective actions or not. Therefore, it is of major importance that the provided information is as accurate as possible. More specifically, when warning signals are generated, the probability that the project will be late should be as high as possible. Correspondingly, when no signals are received, the probability that the project will finish on time should be as high as possible. Consequently, this implies that both the positive and the negative predictive value should be as high as possible.

The positive predictive value can be used as a metric to evaluate the performance of tolerance limits, and is referred to as the *efficiency* (Eq. (8.6)). Using Bayes' theorem, the efficiency can be calculated from the detection performance ($\mathrm{P}[S|RD > \delta_{\mathcal{N}}]$), probability of overreactions ($\mathrm{P}[S|RD \leq \delta_{\mathcal{N}}]$) and the prior probabilities of finishing on time ($\mathrm{P}[RD \leq S_{\mathcal{N}}]$) or late ($\mathrm{P}[RD > \delta_{\mathcal{N}}]$):

$$\begin{aligned} &\mathrm{P}[RD > \delta_{\mathcal{N}}|S] \\ &\quad = \frac{\mathrm{P}[S|RD > \delta_{\mathcal{N}}] \times \mathrm{P}[RD > \delta_{\mathcal{N}}]}{\mathrm{P}[S]} \\ &\quad = \frac{\mathrm{P}[S|RD > \delta_{\mathcal{N}}] \times \mathrm{P}[RD > \delta_{\mathcal{N}}]}{\mathrm{P}[S|RD > \delta_{\mathcal{N}}] \times \mathrm{P}[RD > \delta_{\mathcal{N}}] + \mathrm{P}[S|RD \leq \delta_{\mathcal{N}}] \times \mathrm{P}[RD \leq S_{\mathcal{N}}]} \end{aligned} \tag{8.6}$$

The negative predictive value can also be used to accurately assess the performance of tolerance limits by introducing the *reliability* metric (Eq. (8.7)). Similar to the efficiency, the reliability can be calculated from the detection performance, probability of overreactions, and the prior probabilities of finishing on time using Bayes' theorem:

$$\begin{aligned} &\mathrm{P}[RD \leq \delta_{\mathcal{N}}|S^C] \\ &= \frac{\mathrm{P}[S^C|RD \leq \delta_{\mathcal{N}}] \times \mathrm{P}[RD \leq \delta_{\mathcal{N}}]}{\mathrm{P}[S^C]} \\ &= \frac{(1 - \mathrm{P}[S|RD \leq \delta_{\mathcal{N}}]) \times \mathrm{P}[RD \leq \delta_{\mathcal{N}}]}{(1 - \mathrm{P}[S|RD \leq \delta_{\mathcal{N}}]) \times \mathrm{P}[RD \leq \delta_{\mathcal{N}}] + (1 - \mathrm{P}[S|RD > \delta_{\mathcal{N}}]) \times \mathrm{P}[RD > S_{\mathcal{N}}]} \end{aligned} \tag{8.7}$$

with S^C the complement of S, namely the absence of a signal.

Set 2: Intensity of signals (*measuring the performance at each tracking period*)

At the tracking period level, the performance of tolerance limits is assessed in terms of the number of correct or false warning signals generated during the project life cycle. For projects that are late, tolerance limits should have generated as much correct warning signals as possible during project execution since each signal corresponds to an opportunity for taking corrective actions. Contrarily, timely projects should have generated as few warning signals as possible since each of these signals involved unnecessarily spent effort. Since the signal quality metrics at the project level are not able to assess the number of generated signals, four signal intensity metrics are proposed at the tracking period level. Similar to the project level metrics, two metrics consider the correct and false signals separately, namely the *signal density* and the *signal redundancy*. Two other metrics consider both the correct and the false signals, namely the *signal efficiency* and the *signal reliability*.

The *signal density* reflects the average amount of signals generated during the project lifecycle of late projects and is defined by Eq. (8.8) as follows:

$$\frac{\#\textit{signals for late projects}}{\#\textit{late projects}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_i (RD > \delta_{\mathcal{N}}) S_{ik}}{n_L} \tag{8.8}$$

Furthermore, the *signal redundancy* indicates the average amount of warning signals generated during the project lifecycle of timely finished projects and is represented in Eq. (8.9).

$$\frac{\#\textit{signals for timely projects}}{\#\textit{timely projects}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_i (RD \leq \delta_{\mathcal{N}}) S_{ik}}{n_O} \tag{8.9}$$

The *signal efficiency* combines correct and false warning signals and represents the relative number of correct warning signals generated by the tolerance limits (Eq. (8.10)). The number of correct warning signals is compared to the total number of generated warning signals as follows:

$$\frac{\#\textit{signals for late projects}}{\#\textit{signals}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_i (RD > \delta_{\mathcal{N}}) S_{ik}}{\sum_{i=1}^{n} \sum_{k=1}^{K} S_{ik}} \tag{8.10}$$

Finally, the *signal reliability* represents the relative number of times that warning signals were not generated, rightly so, by the tolerance limits. This metric compares the number of times that warning signals were correctly absent to the total number of tracking periods for which no warning signal was generated (Eq. (8.11)).

$$\frac{\#\textit{absent signals for timely projects}}{\#\textit{absent signals}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_i (RD \leq \delta_{\mathcal{N}}) S_{ik}^C}{\sum_{i=1}^{n} \sum_{k=1}^{K} S_{ik}^C} \tag{8.11}$$

with S_{ik}^C the complement of S_{ik}.

Conclusion

The preceding technical discussion of signal metrics illustrates that the quality of control methods can be validated in different ways. Each perspective uses the *control efficiency* concept implicitly but approaches it differently. Whichever perspective is chosen, they all examine the quality and accuracy of control methods in a similar way and they all focus on measuring the trade-off between (1) the *effort of control* (the ease of use, or the number of control points in the system), (2) the *accuracy of signals* (correct or false signals, i.e., type I and type II errors), and (3) the *quality of actions* taken to resolve problems in a timely manner. None of the previously used concepts in the previous chapters fully integrated these three perspectives, and it is therefore necessary to study these signal metrics more deeply and measure them all simultaneously to evaluate a project control system from all sides. I must admit that I am also sometimes confused with the large number of metrics (8!) and they sometimes overwhelm me and prevent me from drawing clear conclusions. However, since there is currently no standardised definition available to evaluate project control systems, I think these eight metrics should all be used during the evaluation. It would be an interesting research path to integrate these metrics into one unified measure so that it would be much easier to evaluate a project control system. Consider it a call for further research.

8.4 Hope and Dream

This chapter proposed a new methodology to control projects in progress and new quality metrics to assess the performance of project control methods. The analytical control methods are intended to combine the simplicity of the standard control methods (which rely only on simple rules of thumb with simple EVM performance statistics) with the accuracy and quality of the statistical control methods (which make use of statistical tolerance limits but require more data and use advanced statistical methods). It is believed that these analytical methods are within the reach of any project manager who is able and willing to improve the current EVM methodology without much additional effort. These methods can thus perfectly bridge the gap between the simple rules of thumb and the advanced statistical techniques. Despite the belief that these analytical project control methods are easy to use or at least much easier than the advanced statistical project control methods, not much has changed in my experience and most professional project managers continue to stick to their easy rules of thumb (the standard methods) when controlling projects. Nevertheless, I keep trying to convince the project management community of the power of newer and better methods, and that is why I organise at least three times a year a 2-day workshop at Vlerick Business School (Belgium) in which Belgian project managers learn the benefits of using these (analytical and even statistical) methods for their projects. The student ratings are excellent and the program is sold out every time, despite the relatively high registration fee. I often get requests after the training for further guidance and help in implementing these techniques, which indicates that the interest is there. Although I think that these *easy* analytical methods are still a bit too difficult for most project managers, it may also be a matter of time (and money) before the younger generation will adopt them to monitor their projects. Who is to say. . .?

However, I do see positive changes and notice that a number of techniques that do require some data and statistical analyses are becoming more and more accessible to a wider project manager audience. I therefore continue to see it as one of my important tasks to convince professional project managers to implement data-driven methods for managing their projects. I know this is a challenging task and progress may not be as fast as I would like, but I will stay positive and keep going. One of the most encouraging signs that change is on the way lies in the growing use of a data-intensive technique for forecasting project durations and costs known as *reference class forecasting*, which will be discussed in detail in the next chapter.

References

Aliverdi, R., Moslemi N. L., & Salehipour, A. (2013). Monitoring project duration and cost in a construction project by applying statistical quality control charts. *International Journal of Project Management, 31*(3), 411–423.

Batselier, J., & Vanhoucke, M. (2015a). Construction and evaluation framework for a real-life project database. *International Journal of Project Management, 33*, 697–710.

Bauch, G. T., & Chung, C. A. (2001). A statistical project control tool for engineering managers. *Project Management Journal, 32*, 37–44.

Colin, J., Martens, A., Vanhoucke, M., & Wauters, M. (2015). A multivariate approach to top down project control using earned value management. *Decision Support Systems, 79*, 65–76.

Colin, J., & Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega the International Journal of Management Science, 49*, 107–122.

Colin, J., & Vanhoucke, M. (2015a). A comparison of the performance of various project control methods using earned value management systems. *Expert Systems with Applications, 42*, 3159–3175.

Colin, J., & Vanhoucke, M. (2015b). Developing a framework for statistical process control approaches in project management. *International Journal of Project Management, 33*(6), 1289–1300.

Fleming, Q., & Koppelman, J. (2010). *Earned value project management* (4th ed.). Newton Square, Pennsylvania: Project Management Institute.

Goldratt, E. (1997). *Critical chain*. Great Barrington, MA: North River Press.

Hotelling, H. (1951). A generalized t test and measure of multivariate dispersion. *Proceedings of the second Berkeley symposium on mathematical statistics and probability* (pp. 23–41).

Hu, X., Cui, N., Demeulemeester, E., & Bie, L. (2015). Incorporation of activity sensitivity measures into buffer management to manage project schedule risk. *European Journal of Operational Research, 249*, 717–727.

Leu, S. S., & Lin, Y. C. (2008). Project performance evaluation based on statistical process control techniques. *Journal of Construction Engineering and Management, 134*, 813–819.

Lipke, W. (2004). Connecting earned value to the schedule. *The Measurable News, 1*(Winter), 6–16.

Lipke, W., & Vaughn, J. (2000). Statistical process control meets earned value. *CrossTalk: The Journal of Defense Software Engineering, 16–20*(June), 28–29.

Martens, A., & Vanhoucke, M. (2017a). A buffer control method for top-down project control. *European Journal of Operational Research, 262*(1), 274–286.

Martens, A., & Vanhoucke, M. (2017b). The integration of constrained resources into top-down project control. *Computers and Industrial Engineering, 110*, 277–288.

Martens, A., & Vanhoucke, M. (2018). An empirical validation of the performance of project control tolerance limits. *Automation in Construction, 89*, 71–85.

Silver, N. (2012). *The signal and the noise: Why so many predictions fail—but some don't*. Penguin.

Vanhoucke, M. (2019). Tolerance limits for project control: An overview of different approaches. *Computers and Industrial Engineering, 127*, 467–479.

Vanhoucke, M., & Colin, J. (2016). On the use of multivariate regression methods for longest path calculations from earned value management observations. *Omega the International Journal of Management Science, 61*, 127–140.

Wang, Q., Jiang, N., Gou, L., Che, M., & Zhang, R. (2006). Practical experiences of cost/schedule measure through earned value management and statistical process control. *Lecture Notes in Computer Science, 3966*, 348–354.

Chapter 9
Reference Class Forecasting

In the previous chapter, several project control methods were discussed that generate warning signals to take corrective actions in a timely manner so that projects can be completed without problems. These warning signals act as alarms so that the project manager can search for potential problems. The nature of the problem obviously varies from case to case, and it is up to the project manager to estimate the impact of the problem on the project (time and cost) objectives and, if necessary, to compare the current performance against the expected future performance. The future project performance must therefore be accurately estimated, and this estimate uses the current state of progress (in the middle of the project progress) to predict the future. In Chap. 4, I have already suggested a number of ways to predict the expected total duration from EVM data, and you saw that these predictions (obviously) became more accurate as the project progressed. However, none of the proposed methods were able to accurately estimate the expected duration of the project during the early project phases because there was simply too little progress data available.

This chapter presents an alternative method for estimating the expected time and costs of a project in its early phases. More specifically, this method tries to predict the total duration and total costs of the project based on project-specific characteristics by using historically finished projects that display the same characteristics. Making use of similar projects from the past is quite normal since every project manager makes estimates about time and costs based on experience from previous projects. The danger of these estimates is, of course, that the project manager is not very objective and is often tempted to make unreasonable assumptions or even tends to change these estimates in some desired direction. This has been known for a long time, and as early as the seventeenth century, the Marquis of Halifax already knew that predictions based on human experience are prone to error, as he stated:

> The best qualification of a prophet is to have a good memory.

People do not have very good memories (it gets worse for me with age) and some experiments even claim that the best estimate is often not much better than a random guess. The technique discussed in this chapter attempts to avoid this

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_9

arbitrariness by providing better estimates in a structured manner based on historical data. The technique is known as *Reference Class Forecasting* (RCF) and was first introduced by Daniel Kahneman and Amos Tversky, the founding fathers of behavioural economics.

In this chapter, I will give an overview of three studies that we have performed for time and cost estimations based on this RCF technique. I must admit that I cannot call myself an expert in this field and that I was even a bit averse to research in this field at first. It is of course not because I found this technique inferior, but I found the methodology clear and simple and I thought that as a researcher I could add little to it. However, this changed after reading a book and a series of papers while looking for some good and exciting science literature.[1]

The book can be found everywhere, and it is undoubtedly one of the best-selling books in the world because I have never been to an airport where it was not for sale. The book is titled *"Thinking fast and slow"* and was written by Daniel Kahneman (2011) after his friend and colleague Amos had already died. There are not many books that have had such an impact on my career like this one, and to this day, I recommend the book to my students in any *Project Management* or *Decision Making for Business* course module. The book shows in an incredibly original way how irrational people are, especially when making decisions under uncertainty. I therefore consider the book a valuable resource not only for my academic research but also in my day-to-day life. The book does not explicitly refer to the domain of project management, but it was immediately crystal clear that the presented theories, including *reference class forecasting*, could have many applications in my favourite field of research. One of the first persons who gained this insight was undoubtedly Bent Flyvbjerg. He has carried out an incredible series of studies on this project forecasting theme and has shown that the RCF technique is an extremely valuable technique in the field of project management. I would be happy to summarise his research, but this would lead me way too far. However, I advise every reader to look for his papers and get started with his numerous recommendations. I will make my own limited contribution in this chapter by providing an overview of my three studies in this field (Sects. 9.2–9.4). There is no doubt that everything that I did in these studies was heavily inspired by what I have read in Bent's papers. Therefore, I will first explain the general principle of RCF in the next section (*outside view*) before moving on to the three OR&S research studies.

[1] I must admit that I rarely read novels, because in the scientific literature, especially in the popular science literature, there can be so much excitement that it not only thoroughly improves my knowledge but also sometimes completely blows my imagination.

9.1 Outside View

Before starting a project, project managers must estimate how long the project will take and how much it will cost. Traditionally, project managers focus on specific characteristics of the project under consideration to make these estimates, relying on their experience when trying to predict uncertain events that would affect the future course of the project. Such an approach is often referred to as an "*inside view*" forecasting approach because it is clearly based on human judgement taking into account specific characteristics of the project. More precisely, these forecasts are typically based on characteristics such as the project size, the estimated durations and costs for the activities, the network structure and/or available resources, and even the likelihood of project delays or cost overruns based on an estimated project risk profile. However, quite a number of studies have shown that human judgement is biased, as it is generally too optimistic because of overconfidence and insufficient regard to actual previous experiences (*"optimism bias"*). Moreover, project managers could also deliberately and strategically underestimate costs and durations to give the impression that they would surpass the competition ("*strategic misinterpretation*"). Therefore, Kahneman and Tversky (1979a,b) propose to take a so-called *outside view* forecasting approach that suggests not taking into account the specificities of the project to be estimated but make predictions based on a class of similar projects carried out in the past and for which the real durations and costs are therefore known.

This idea may seem very logical, but the most important condition to be able to make an accurate prediction lies in the way in which projects are compared. Since the past projects must be similar to the new project, a so-called *reference class* of historical projects—containing similar characteristics to the current project—must be identified. Once this class is determined, the discrepancy between their initial estimated time and cost (from their baseline schedules) and the final actual time and cost (after their execution) can be calculated, and this discrepancy is called the *forecast error*. The general idea is that this error will probably also be made for the new project as it shows so many similarities with the previously executed projects in the reference class. Therefore, the RCF technique will slightly adjust the prediction of the new project and take this error into account. More specifically, once a forecast for the new project has been established using a traditional forecasting method, the distribution of the forecast errors will be used to adjust the initial budget, and such adjustment is called an *uplift*.

The three phases of the RCF method (*reference class*, *forecast error*, and *uplift*) will be illustrated with a fictitious example to demonstrate the simplicity and elegance of the concept. The example considers a fictitious project with a budgeted cost of €100, and this budget must be validated (i.e., based on the budgeted costs, the actual expected costs must be predicted) with the RCF technique using only one illustrative property of similarity called "*experience of the company*". The company experience is expressed in number of years of existence, and three classes are distinguished, namely, mature companies ($\geq$40 years), young companies

Table 9.1 Actual and budgeted cost for historical projects in three reference classes

Reference class "≥40 P years"				Reference class "]10,40["				Reference class "≤10 years"			
ID	Budget	Actual	F_x	ID	Budget	Actual	F_x	ID	Budget	Actual	F_x
1	98	100	0.0204	11	98	105	0.0714	21	98	110	0.1224
2	102	96	−0.0588	12	102	103	0.0098	22	102	112	0.0980
3	100	96	−0.0400	13	100	105	0.0500	23	100	112	0.1200
4	99	95	−0.0404	14	99	106	0.0707	24	99	114	0.1515
5	101	102	0.0099	15	101	100	−0.0099	25	101	109	0.0792
6	101	97	−0.0396	16	101	100	−0.0099	26	101	111	0.0990
7	99	98	−0.0101	17	99	105	0.0606	27	99	113	0.1414
8	103	99	−0.0388	18	103	104	0.0097	28	103	112	0.0874
9	100	99	−0.0100	19	100	105	0.0500	29	100	111	0.1100
10	101	105	0.0396	20	101	104	0.0297	30	101	110	0.0891

(≤10 years), and a class in-between (]10,40[). A fictitious historical database of 30 projects is shown in Table 9.1 to illustrate the three phases of predicting reference classes as discussed below.

Phase 1: Identify a Relevant Class of Historical Projects (*Reference Class*)
In this first step, the reference classes must be constructed that consist of projects with the same characteristics as the current project. The structure of these classes strongly depends on the choice of properties that will be used to categorise the projects. A *property* is an important project attribute, quality, or characteristic that—according to the project manager—is a good indicator of the similarity between projects. A reference class consists of projects that have the same value for one or more properties. In the case of categorical data, the assignment of projects to reference classes is relatively simple as projects belonging to the same category are assigned to the same reference class. However, when properties are measured using numerical data, intervals of the property values are used to assign projects to a reference class. In general, a reference class must be wide enough to ensure a significant number of projects in the class, but it must also be narrow enough to ensure that the projects in the class are sufficiently similar. Choosing the right properties is an art in itself and the identification of such properties is discussed later in Sect. 9.4. In the artificial example, only one property is used ("*number of years of company existence*"), which greatly simplifies the problem and is therefore not very realistic. The project database of Table 9.1 shows the budgeted costs, actual costs, and the forecast errors for 30 historical projects, of which 10 projects come from mature companies, 10 projects from young companies, and 10 projects from companies that exist between 10 and 40 years. A closer look at the values reveals that the budgeted costs of projects in the reference class "≥40" are usually an overestimation of the actual project costs, while the opposite is true in the reference classes medium-experienced and young companies. Especially in this last reference class ("≤10"), the projects are significantly and systematically underfunded. These three reference classes form the basis for predicting the total cost of a new project.

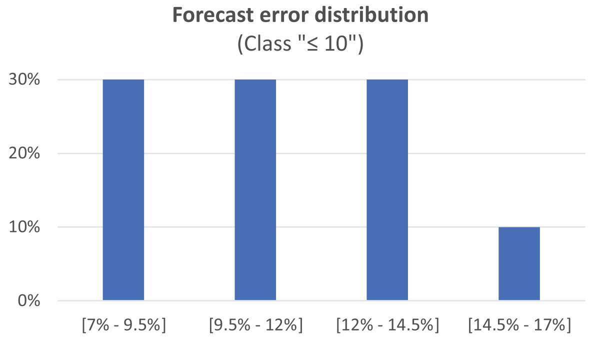


Fig. 9.1 Distribution of forecast error ("≤ 10" reference class)

Suppose the new project is being done in partnership with a company that has been around for less than 10 years, and the budgeted cost of €100 needs to be validated using the historical project database. Since the new project belongs to the third reference class of Table 9.1, the data from these 10 projects with the same value for this property (i.e., projects 21–30) should be used as reference class. This reference class is used to compose the error in the second step of the procedure.

Phase 2: Establish a Distribution for the Reference Class (*Forecast Error*)
Once a desired reference class is chosen, a probability distribution of the forecast errors of the historical projects in this reference class must be determined. Table 9.1 shows that the budgeted costs for these projects in the reference class are between €98 and €103, close to the budget of the new project, while the actual costs were significantly higher (between €109 and €114). The forecast error is measured as $F_x = \frac{\text{Actual}-\text{Budget}}{\text{Budget}}$ and shows that the errors vary from 7.92% (for project 25) to 15.15% (for project 24). The distribution of these forecast errors is displayed in Fig. 9.1, which shows the frequency of the forecast errors between 7% and 17% in bins of 2.5 percentage points. This probability distribution will be used to determine the required uplift for the project in the third step.

Phase 3: Compare the Project with the Error Distribution (*Uplift*)
In order to adjust the initial budget of the new project to a more realistic cost estimate, the distribution of the reference class must be transformed to a *cumulative probability distribution*, which shows the cumulative frequency in function of the forecast error. This cumulative frequency of forecast errors is displayed in the left graph of Fig. 9.2 and shows that, for example, 5 out of 10 projects have a forecast error larger than 10%. Based on this curve, the *inverse cumulative distribution* must be constructed to understand the relation between the required uplift and the acceptable chance of cost overruns. This is shown in the right graph of Fig. 9.2, which displays the probability of cost overruns for different uplifts (i.e., for all values of the forecast errors). For example, it is known that an uplift of at least 7.92% is necessary since this is the minimum error value for the 10 projects of the current reference class (cf. Table 9.1). Consequently, in order to obtain a realistic project budget for the new project, the original budget (€100) should be uplifted

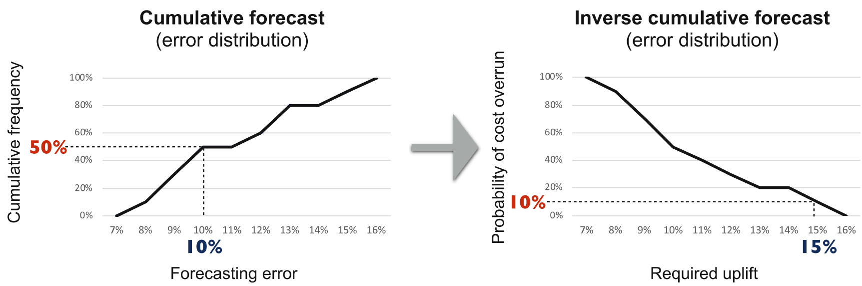


Fig. 9.2 Cumulate error frequency and required uplift

to at least €107.92. Suppose that the budget is set to €107 (7% uplift), then the likelihood that the project will have a budget overrun is expected to be 100%. If the budget is increased to €115 (i.e., an uplift of 15%), this probability of cost overruns drops to only 10% as can be seen in the graph. Consequently, this right graph allows the decision makers to assess their willingness to accept risk for the new project and determine the corresponding uplift of the estimated costs of the original budget.

I hope that this simple example has convinced the readers that the RCF technique is a relatively simple but very useful technique for forecasting project costs and durations. However, I should note that the method makes no attempt to predict very specific events that could affect the specific project. Instead, it focuses solely on predicting the expected time and costs of an unseen project based on previous projects where the actual time and costs are known. In the next paragraphs, I will summarise three studies using this technique on real data conducted together with some colleagues from my research team.

9.2 Construction Project (*Study 1*)

Our very first investigation into the use of the RCF technique for project duration and cost forecasting began with the collection of empirical project data (discussed in Chap. 13). Initially, we wanted to use the three forecasting techniques from Chap. 4 for empirical projects (instead of the artificial projects that we always used until then), but we soon realised that comparing these methods was not going to provide many additional insights. In contrast, we did feel that we would be better off using some completely different prediction techniques to properly understand how predictions work on real projects. After all, the three forecasts of Chap. 4 are all based on Earned Value Management (EVM) metrics and so we decided to compare them with two other methods, namely the simplest forecasting method of all (namely, baseline schedule forecasting) and the more advanced forecasting methods that use Monte Carlo simulation runs. In addition, we had also decided to use the RCF method as the ultimate goal of the study was to find out whether the reference class forecasts would perform better than the more “classical” methods.

Our ambition in this first study was not so much to conduct a full study on a broad set of empirical projects, but instead, we wanted to make the existing forecasting methods of our previous research studies a little more known to a bigger audience. Therefore, we decided to submit the study to the *Project Management Journal*,[2] a popular and high-quality journal that typically attracts a wider audience than just academics. Since we limited the predictions in this study to only one project, the results should of course be interpreted with caution. The project studied was C2013-17 from the empirical database, and both project duration and final cost were predicted using the four prediction methods mentioned earlier. Project C2013-17 is a real construction project and involves the execution of finishing works in an office building consisting of the interior carpentry and installation of drywall, relocatable partitions (including acoustic), raised floors, suspended ceilings, and furniture. The work was carried out by a medium-sized finishing company with extensive experience in this field. Nevertheless, the considered project includes some smaller works that are rather unusual for the company, such as the installation of carpets and special glass walls. The project consists of a list of 23 activities, each with planned costs and durations to establish a baseline schedule for the project as well as known risk profiles for each individual activity, estimated by the project manager. The four prediction methods used in the study are summarised along the following lines:

Method 1: Baseline Schedule Estimates (BSEs) The budget at completion (BAC) and planned duration (PD) represent the final cost and duration of the project, respectively, estimated by the project manager based on expectations of the future course of the project (i.e., the *inside view*). These two key metrics are the result of baseline planning and are thus defined before the start of the project. These estimates are collectively referred to as baseline estimates in this study and can thus be regarded as *easy* predictions for the project.

Method 2: Earned Value Management (EVM) Of course, the baseline schedule estimates need to be adjusted and improved as more data become available. The baseline estimates from the baseline schedule are used as input for the EVM methodology during project execution. These periodic progress measurements can be used to predict the final expected duration and cost of the project, as was demonstrated in Chap. 4. Thus, unlike the BSE, the EVM methodology provides a set of forecasts during the project progress, and the average value for all the time and cost forecasts will be used as a basis of comparison for the other forecasting methods.

Method 3: Monte Carlo Simulation (MCS) The periodic forecasts of the EVM method are based on progress data coming in gradually over the course of the project. At the beginning of the project, there is no data available and no forecasts can be made. However, this is not the case with the MCS method, which, like the

[2] The study was published in Batselier and Vanhoucke (2016) and is a collaboration with Jordy Batselier, a PhD student at OR&S between 2012 and 2016 whom I will introduce in Chap. 10.

BSE method, tries to generate forecasts before the project has started. Monte Carlo simulation is therefore an approach to obtain more substantiated estimates of project cost and duration before the project is started, using risk distribution profiles for individual activities. The use of Monte Carlo simulation has already been explained in several places in this book (e.g., in Chap. 5 as the so-called *static simulation runs*) and it is used as a *schedule risk analysis* to obtain sensitivity metrics for each project activity. These same static simulation runs can now be used to obtain a *probability distribution* for the total time and cost of the project, which can then be used as probabilistic estimates. In this study, we applied easy-to-use triangular distribution profiles, which can be symmetric, skewed to the left or skewed to the right. It is important to mention that the assignment of these distribution profiles to the different activities of the considered project was carried out by the project manager based on experience with previous projects having similar activities. Because this process uses historical data, it is tempting to think that the MCS method is an *outside view* forecasting technique (like RCF). However, unlike RCF, Monte Carlo simulations still require distributional information for each activity, which will often require the project manager to make (arbitrary or unsupported) assumptions (e.g., for unusual activities), which in turn is a typical feature of the *inside view* similar to the BSE method. Therefore, the predictions of the MCS method are best described as a *"semi-outside view"* of project forecasting.

Method 4: Reference Class Forecasting (RCF) To apply the *outside view* of the RCF technique, it is necessary to identify a reference class of projects that are similar to the project under consideration. This reference class must be wide enough to be meaningful, but narrow enough to be truly comparable to the project under consideration. As the C2013-17 project of this study is a construction project, 24 other construction projects from the empirical database were selected as a basis for comparison to perform the three aforementioned stages of reference class prediction:

Phase 1. Reference class: To identify the relevant classes, we considered four different compositions of reference classes, ranging from broad sector-oriented to company-specific categories. Because the construction industry is very broad, we therefore first divided our set of 24 empirical construction projects into three classes, referred to as *general construction*, *building construction*, and *commercial building construction* projects in order of increasing specificity and similarity to the considered project under study. Finally, a fourth class was defined as *office finishing work*, which is the most specific reference class with the highest degree of similarity with the original project, since it only includes finishing constructions carried out by the same company as the one who carried out the project under consideration.
Phase 2. Forecast error: The second phase requires the preparation of a probability distribution for the selected reference class to determine the forecast error. Such a distribution is necessary for determining the required increase (*uplift*) of the original time/cost estimates (i.e., the budget increase relative to the initial BAC estimate and the time increase relative to the initial PD estimate) that corresponds to a certain acceptable probability of cost or time overrun. However, in this study, we did not

explicitly look at the probability distributions for the selected reference classes as is usually done in this second phase. This is because our intention was to compare RCF to traditional forecasting approaches (BSE, EVM, and MCS) that mostly provide point estimates of the most likely project cost and duration. Therefore, we were only interested in getting the most likely outcome for the project under consideration (i.e., a point estimate) rather than the full probability distribution function.

Phase 3. Uplift: To get a forecast for the final duration and cost of the project, the initial estimates of the baseline schedule (PD and BAC) must be increased (*uplifted*) by the most likely outcome of the previous phase. Since we are working with point estimates in our study, this corresponds to an increase needed for the 50th percentile of the probability function in the RCF procedure. These estimates are then compared with the estimates of the other methods (BSE, EVM, and MCS) to see which method generates the best predictions.

The quality of the predictions of the four methods was assessed using the two performance measures discussed in Chap. 4, namely the *accuracy* and *stability* of the predictive values. The results can be summarised along the following lines:

Cost Forecasting We saw that the Monte Carlo simulation (MCS) technique provides a more accurate cost prediction than the BAC estimate of the baseline schedule method (BSE), which is not very surprising, but the improvements were very modest. The RCF technique using the most specific reference class of projects (*office finishing*) was the most accurate cost forecasting method and even slightly outperformed EVM cost forecasting methods (with nearly similar results). This is quite remarkable, as the EVM methodology allows for continuous predictions that can be updated during project progress (based on current progress data). RCF, on the other hand, generates only one fixed forecast at the start of the project, so it remains unchanged throughout the project. The fact that the RCF technique makes this prediction quite accurate right away without the need to adjust it is remarkable and plays to the advantage of this technique to perform cost predictions. Since RCF provides constant predictions, the approach logically exhibits perfect prediction stability, which is obviously not the case with EVM.

Time Forecasting When we used the baseline estimate PD as the reference value, we saw that the Monte Carlo simulation (MCS) technique now provided a much bigger improvement in accuracy than for the cost forecasts. This observation confirms the results in previous studies of this book that the project network structure—with critical and non-critical activities—has a major impact on the accuracy of a schedule risk analysis. For the RCF methodology, we saw that an increasing similarity between the original project and the four reference classes resulted in an increasing prediction accuracy. Moreover, when applying RCF with the most specific reference class of in-company projects (*office finishing work*), the EVM method was surpassed by the RCF method, which corresponds perfectly with the observations for cost forecasting.

Despite the fact that this study was quite a simple one without presenting new methodologies and new algorithms, the results show that the RCF method indeed performs best, both for cost and time predictions. These observations therefore support the practical relevance of the technique and have convinced us to continue with this technique in our research group. It should be noted, however, that this study had a number of weaknesses and the results should therefore be put into perspective. Firstly, the forecasts were only performed on one case study project, and therefore the results cannot be generalised. A more significant weakness is that comparing these four forecasting methods was somewhat unfair. As mentioned, the BSE and RCF methods provide a single forecast that is available before the start of the project and is never adjusted. However, the EVM technique provides multiple forecasts, i.e., one value for each tracking period of the project from start to finish. For these forecasts, average values were therefore used to compare them with the point estimates from the previous methods. Finally, the MCS method normally provides a distribution of forecasts prior to the project start (such as BSE and RCF), but these forecasts can be easily updated as the project progresses, resulting in multiple forecast values similar to EVM. The fact that these four methods work so fundamentally different should be kept in mind when comparing the results. It was therefore an obvious step to integrate these different prediction methods into a hybrid approach, which is the subject of the next study.

9.3 Hybrid Approach (*Study 2*)

The starting point of the second study was to improve the EVM technique to make more reliable periodic time and cost predictions. The research in the previous section had shown that the stability of EVM forecasts is sometimes relatively low because they generate different prediction values for each follow-up period of the project. The RCF method obviously does not suffer from this drawback because it only generates a single prediction value (before the start of the project). It therefore seemed intuitively clear that a combination of these two methods could lead to possible improvements by adapting the periodic EVM methods to achieve both high accuracy and stability.

The integration of these two methods consists of removing very extreme prediction values for the periodic forecasts of the EVM method by using the well-known *exponential smoothing* methodology. Although the exponential smoothing technique is mainly used in financial and economic environments, it can in fact be applied to any individual set of repeated measures (i.e., to any time series). Since the tracking data collected during project progress forms a time series, exponential smoothing can also be applied to predict the project duration and project costs. Indeed, traditional EVM forecasts assign equal importance (or weight) to all previous observations, while the exponential smoothing approach allows the weight of older observations to be gradually reduced. This hybrid approach should thus combine the best components of these different methods to improve the quality of predictions, in terms of both accuracy and stability. It will be shown that the parameters for exponential smoothing are best determined using the RCF technique, which is why Study 2 is called a hybrid technique. To properly explain the hybrid form of EVM forecasting, I must first refer back to Chap. 4 where EVM methods were discussed and used as forecasting methods in three different ways (planned value method, earned duration method, and earned schedule method). However, I did not provide technical details on how these time/cost forecasts were made in this chapter, and I will therefore elaborate on that first in the following paragraphs. I will restrict myself to the *earned schedule method* (further abbreviated as ESM)—which is the time-based extension of the traditional earned value method—because it provides the most accurate project duration forecasts. I will use this ESM technique to forecast the project duration and extend it to cost forecasts. Later in this chapter, I will incorporate a smoothing parameter to smooth the time and cost predictions and integrate it with the RCF technique.

EVM Forecasting

The use of EVM methods to predict time and cost was already cited in Chap. 4, but the prediction formulas used were not explicitly mentioned. The general *time forecasting* formula for an EVM-based prediction can be formulated as follows:

$$EAC(t) = AT + \frac{PD - ES}{PF} \tag{9.1}$$

with AT, ES, and PD being three metrics that have been used earlier in this book. More specifically, AT is the actual time of the project (i.e., the number of days the project is in progress up to today), and PD is the planned duration of the project in the baseline schedule and ES is the earned schedule metric at time AT, which is equal to the earned value (EV) metric but expressed in time instead of a monetary unit.

The PF is new and known as the *performance factor*, which expresses how the future progress of the project will evolve given the current performance of today (at AT). In the previously mentioned study of Vandevoorde and Vanhoucke (2006), it is argued that the performance factor can have three different values, and the choice of this value depends on the nature of the problems (or opportunities) that happened during the project's progress. Consider, for example, a project with a planned duration PD of 5 weeks for which 2 weeks have passed (AT = 2) and an ES value of 1.5 weeks (i.e., 0.5 weeks delay), and then the three PF versions can be explained as follows:

- PF = 1: Past performance is not a good predictor of future performance. Problems/opportunities in the past will not affect the future, and the remaining work will be done according to plan. For the example project, the time prediction EAC(t) $= 2 + (5 - 1.5) = 5.5$ weeks meaning that the half-week delay has been lost and thus cannot be recovered, but no further delays are to be expected.
- PF = SPI(t): Past performance is a good predictor of future performance (which is often more realistic). Problems/opportunities in the past will affect the future performance, and the remaining work will be corrected for the observed efficiencies or inefficiencies. Since the schedule performance index SPI(t) of the example project is equal to ES/AT = 75%, it updates the EAC(t) to $2+(5-1.5)/0.75 = 6.6$

weeks. The half-week delay at AT = 2 is likely to lead to further delays, and the total duration time will therefore be significantly higher than initially planned.

- PF = CPI × SPI(t): Not only the past time performance but also the past cost performance is a good indicator of future performance (i.e., cost and schedule management are inextricably linked). The PF = SPI(t) × CPI is often called the *critical ratio index* (SCI).

The generic formula for *cost forecasting* is very similar to Eq. (9.1) for time forecasting but replaces any time metric with a cost index, as follows:

$$EAC(€) = AC + \frac{BAC - EV}{PF} \tag{9.2}$$

with AC being the actual cost at the current time AT, BAC being the budget at completion of the baseline schedule, and EV being the earned value metric at time AT, which is identical to the ES metric but expressed in a monetary unit. The performance factor (PF) can be set at three different levels to express the expectations of the future project progress similar to the three versions discussed earlier. More specifically, PF can be equal to 1 (no further cost problems are to be expected), CPI (the current cost performance is a good indicator for the future cost performance), and CPI × SPI(t) (both time and cost determine the future project performance).

Performance Factor

Selecting the right performance factor obviously depends on the project manager's expectations about the further course of the project. In some previously mentioned studies that eventually were published in my book "*Measuring Time*", I saw that the performance factor is best set to the SPI(t) value for time predictions. These results were obtained by using dynamic simulation runs to mimic project progress on a series of artificial projects and indicate that the past time performance is a good indicator of the expected future performance, ultimately leading to the highest forecasting accuracy. The advice to use the SPI(t) for time forecasting was already included in one of my first EVM publications with Stephan (Vanhoucke & Vandevoorde, 2007) and was thereafter often repeated and confirmed by additional experiments. This lasted until we suddenly conducted a study on empirical projects and were no longer able to show that the SPI(t) should be used as a performance factor. Indeed, in the empirical study of Batselier and Vanhoucke (2015b), we saw that the unweighted ESM technique (with PF = 1), further referred to as the ESM-1 method, rather than the ESM-SPI(t) method, clearly produced the most accurate time predictions. However, it could be argued that this method is intuitively unrealistic because it does not take into account current schedule performance, while the weighted ESM-SPI(t) method does take into account past performance. However, the SPI(t) reflects the *cumulative* time performance of all previous periods up to AT, which assumes that the performance of *every* past tracking period has an equal impact on future expectations. This implies that the SPI(t) cannot accurately account for the following two possible influences, which were not included in the simulation studies on our artificial project datasets:

- The occurrence of natural performance improvement during the course of the project due to increasing experience levels of the resources (e.g., workers)
- The effect of corrective management actions that were taken recently with the aim of improving future performance

Because the SPI(t) gives equal weight to each period, the ESM-SPI(t) will always include the performance of the earliest project phases as well. To overcome these drawbacks, it seems appropriate to give more weight to the performance of the most recent project phases as they best reflect the effect of experience-based performance improvements and/or the effect of current management efforts. Unlike the ESM-SPI(t) method, the ESM-1 method more or less takes into account the effect of increasing levels of experience and corrective actions taken by assuming that future performance will proceed exactly as planned (i.e., according to the baseline schedule). Of course, an intuitive problem arises with this assumption through the use of the term "*exactly*". First, there is no guarantee that experience will necessarily lead to a performance improvement (so that future productivity would increase by itself) or that corrective actions will lead to a project where the future goes according to plan. Moreover, if such occasions did arise, it is highly unlikely that they would lead to exact future compliance with the original plan, as the ESM-1 method does presume. Nevertheless, when resources begin to work more efficiently due to increased experience levels, the resource costs reduce as the tasks being performed take less time to complete and so the ESM-1 method might be closer to the expected performance than the ESM-SPI(t) method. The main conclusion that we could draw based on the above discussion and the observation that the empirical results differ from the simulated results was that we identified the need for a modified time forecasting method that lies somewhere between ESM-1 and ESM-SPI(t) (and similarly, for cost forecasting between PF = 1 and PF = CPI). This new method should be able to give more weight to more recent tracking periods, taking into account the potential impact of increasing experience levels of resources and/or management corrective actions. Furthermore, we added the additional requirement that the adjusted method should be able to express changes in management attention through an adjustable parameter. Taking all these conditions into account, the technique of *exponential smoothing* quickly emerged as the ideal basis for developing the desired new time prediction method.

Smoothing Parameter

Exponential smoothing is a forecasting technique for time series data that solves the problem of equally weighted observations and uses exponential functions to assign exponentially decreasing weights over time with a smoothing parameter β. Therefore, the performance factor PF is replaced by a more sophisticated expression using the smoothing parameter, which redefines the PF for time forecasting as

$$PF = \frac{T_{t,ES}}{T_{t,AT}} = \frac{\beta(ES_t - ES_{t-1}) + (1-\beta)T_{t-1,ES}}{\beta(AT_t - AT_{t-1}) + (1-\beta)T_{t-1,AT}} \tag{9.3}$$

The AT$_t$ and ES$_t$ metrics now contain the subscript t to refer to the value of these metrics at a certain tracking moment t. Adding a subscript t to refer to a certain tracking period can be confusing, which is why I mostly leave it out of the metrics. However, here it is necessary to clearly define the β parameter. The current tracking moment is referred to as moment t and corresponds to an actual project duration of AT_t, and the previous tracking moment is equal to $t-1$ for which the actual project duration was equal to AT_{t-1}. For example, assume a project that started at week 0 and reviews the project progress every 2 weeks. Assume that the current time is equal to week 6. This is the third tracking moment ($t = 3$) held at week 6 ($AT_3 = 6$), while the previous tracking moment ($t - 1 = 2$) was held at week $AT_2 = 4$.

For cost forecasting, a similar PF function is defined in which ES is replaced by EV and AT is replaced by AC as follows:

$$PF = \frac{T_{t,ES}}{T_{t,AC}} = \frac{\beta(EV_t - EV_{t-1}) + (1-\beta)T_{t-1,EV}}{\beta(AC_t - AC_{t-1}) + (1-\beta)T_{t-1,AC}} \tag{9.4}$$

The newly derived performance factors of Eq. (9.3) (*time*) and Eq. (9.4) (*cost*) are influenced by the smoothing parameter β, which can be explained by analysing two extreme cases (explained for *time* only). First, if $\beta = 1$ (maximum responsiveness to the current schedule performance), then $PF = SPI(t)_{inst}$.[3] In this case, the effect of a corrective management action performed during the current tracking interval would be integrally extrapolated to the remaining portion of the project. For example, consider a situation where management has assigned extra resources to a particular project during the last tracking interval. Assume that this has led to a considerable increase in schedule performance for this last interval, compared to the performance earlier in the project. In this case, a choice of $\beta = 1$ ($PF = SPI(t)_{inst}$) would imply that the recently achieved augmented schedule performance will be maintained for the rest of the project's life (i.e., this would reflect a situation where the extra resources remain in service until the very end of the project and maintain the current performance level). On the other hand, if $\beta = 0$ (no responsiveness to the current schedule performance), then $PF = T_{t-1,ES}/T_{t-1,AT} = T_{0,ES}/T_{0,AT} = 1$, producing the well-known $ESM-1$ method, which assumes that future progress will be exactly according to plan (i.e., according

[3] $SPI(t)_{inst}$ is the instantaneous $SPI(t)$, reflecting the schedule performance over the last tracking interval. More specifically, $SPI(t)_{inst}$ is calculated by dividing the increase in ES during the last tracking interval by the corresponding increment of AT, or $(ES_t - ES_{t-1})/(AT_t - AT_{t-1})$. Notice the difference between $SPI(t)_{inst}$ and the standard cumulative $SPI(t)$, which represents the schedule performance over the *entire* project up to the current tracking period.

to the baseline schedule). Obviously, we are not limited to only using one of these two extreme βs. There is an entire spectrum of β values, ranging from 0 to 1, possible for selection. The general rule is that the closer β is to 1, the more weight is assigned to the more recent tracking periods. The parameter β of the proposed method thus provides the required possibility of adjusting the level of forecast responsiveness to the more recent schedule performance of the project.

Computer Experiments

To verify whether this new method, which we have called the XSM (an acronym for *eXponential Smoothing-based Method*) in Batselier and Vanhoucke (2017), can indeed generate better predictions, we selected 23 projects from the empirical project database of Batselier and Vanhoucke (2015a). All of these projects contain fully authentic baseline schedule and tracking data that were received directly from the actual project owners.[4] 21 of the projects can be situated within the broad construction sector, while the other two are IT projects. Furthermore, project durations range from only 2 months to more than 3 years and project budgets from less than € 200,000 to over € 60,000,000. When predicting the time and cost of a certain project in progress, the smoothing parameter β can be set at various values as follows:

- β_{opt}: The optimal value of β for a certain project in the database. This optimal value can only be obtained when all progress data of the project are known for each tracking period, and the β parameter is set in such a way that the accuracy is maximised. This is unrealistic since it assumes that the complete progress of the project is known in advance. However, the β_{opt} parameter is used as a benchmark to compare the accuracy of the predictions with the three other β parameters discussed hereafter.
- $\beta_{opt,oa}$: The optimal value of β over all projects in the database (except the one for which the forecast is made). This approach assumes that historical projects are used to calculate the smoothing parameter. These completed projects can therefore be used to find the optimal value of the β parameter, which will then be used as the smoothing parameter for the new project in progress.
- $\beta_{opt,rc}$: The optimal value for β over all projects within a same reference class, i.e., with similar characteristics w.r.t. sector, budget, duration, etc. This approach is similar to the previous approach since it assumes that past project data are available. However, rather than using *all* projects to determine the best β value, only projects lying in the same reference class are now used to calculate the best value of the smoothing parameter.
- β_{dyn}: The variable smoothing parameter value that is calculated for every tracking period based on the performance of the past tracking periods of the project in progress. Unlike the previous methods, this β_{dyn} can thus be different

[4] For more information about the concepts of *project authenticity* and *tracking authenticity* for empirical project data, the readers are referred to Chap. 10.

for every tracking period and does not look at historical projects as for the $\beta_{opt,rc}$ (*reference classes*) and $\beta_{opt,oa}$ (*all projects*) methods.

In order to integrate the reference class forecasting (RCF) technique into the exponential smoothing-based EVM method (XSM), the $\beta_{opt,rc}$ should yield more accurate forecasts than the $\beta_{opt,oa}$. The results on the 23 projects indeed confirmed this expectation, as the performance of the predictions could be improved through the consideration of reference classes. Indeed, when applying $\beta_{opt,rc}$ for a reference class of similar projects from the database, the accuracy increased to 13.9% for time and 22.2% for cost. We saw that the new method performed better for cost than for time forecasting, although the improvements can be deemed considerable in both contexts. Despite these improvements when using RCF in time/cost forecasting, obtaining a pool of projects sufficiently similar to a certain project still remains a challenging task. I argued before that the highest degree of similarity is required to yield the highest forecasting accuracies. In the study of Sect. 9.2, the reference class consisted of similar projects from the same company, and this approach was also followed in this second study for the calculation of $\beta_{opt,rc}$. However, finding other criteria to define similarity between projects should increase the accuracy of the RCF methodology even further, and the search to more advanced similarity drivers is the topic of the third study discussed in the next section.

9.4 Similarity Properties (*Study 3*)

Since any reference class forecasting method to predict time and cost of projects is based on an *outside view* (i.e., characteristics of similar past projects) rather than an *inside view* (i.e., estimates of project managers), the usefulness of the methodology depends to a great extent on the ability to establish suitable reference classes. Hence, the definition and identification of *similar* projects is of the greatest importance. In the two previously discussed research studies, the reference classes are mainly constructed based on the size and the type of industry of the projects. This means that projects belong to the same reference class when they can be classified in the same sector, division, or company. This way of classifying projects can be explained by some pioneering studies that used the RCF technique for large-scale (i.e., based on project size) infrastructure and transportation (i.e., based on type of industry) projects. Table 9.2 displays an overview of the classification properties used in different research studies on RCF. Such a rudimentary classification of projects based on generic properties might be too simplistic and might result in a major problem. It is true that the historical data from the reference class should be highly similar to the new project, but every project faces its own problems based on individual circumstances (resources, technologies, management methods, etc.) in its own project environment, and the generic properties to define project classes might fall short detecting these project-specific features. The main challenge in RCF is how to define project similarity and how to create the appropriate reference class.

Table 9.2 Properties of similarity used in different research studies

Study	Parameters	# Projects	This chapter
Flyvbjerg et al. (2002)	Type of project, geographical location	258	–
Flyvbjerg et al. (2004)	Type of transportation project (group and subgroup)	252	–
Cantarelli et al. (2012a)	Geographical location	78	–
Cantarelli et al. (2012b)	Project phases	78	–
Cantarelli et al. (2012c)	Project type, project phase, project size	78	–
Leleur et al. (2015)	Type of project, size, economic situation	262	–
Cantarelli and Flyvbjerg (2015)	Project ownership	183	–
Batselier and Vanhoucke (2016)	Sector, subsector, division, company	24	Yes
Batselier and Vanhoucke (2017)	Company	23	Yes
Walczak and Majchrzak (2018)	Company-specific	222	–
Servranckx et al. (2021)	Type of deliverable, project complexity, experience of company, project definition, governmental law, impact employees	52	Yes

The search for similarities in projects that are—by definition—unique reminds me of a quote from the American cultural anthropologist Margaret Mead:

> Always remember that you are absolutely unique.
> Just like everyone else.

In a third study published in Servranckx et al. (2021), we argued that there is a need to combine the RCF method with expert judgement in order to have an *outside view* based on more detailed and specific properties. The research was conducted with two of my closest colleagues and friends, Tom Servranckx and Tarik Aouam, in collaboration with many project managers from the field. I have already introduced you to Annelies Martens in Chap. 8 in my book, but I must also introduce you to Tom Servranckx.[5] Tom was a PhD student between 2015 and 2019 and then decided to remain in my team as a postdoctoral researcher. Without him, the OR&S team would not be what it is today, and I am very grateful to have him around me as a colleague and a friend. Together we work with an endless passion on our many challenges that we face in our research. The paper also has a third author, Tarik Aouam, who is a fellow professor in our department and with whom I have an intense friendship

[5] I should really pay more attention to Tom's work, as I did in Chap. 8 to Annelies' work, but his research topic (project scheduling with alternative technologies) is somewhat beyond the scope of this book. I am sure I will write another book 1 day where I will describe his results in more detail, and I will refer to Tom again in Chap. 15.

in addition to an occasional collaboration. Apart from these two co-authors, many project managers and field experts joined this third study and were asked to come up with more relevant properties of similarity than the existing properties we already saw in the project data. Instead, we asked them for identifying their own properties that typify the project data based on their experience in the field. For this reason, our approach to RCF can be distinguished from the existing approaches in literature in two ways. First, the projects in a reference class do not have to belong to the same company or industry as they can be considered similar even when they are collected from different companies and industries. Secondly, the reference classes will be constructed using multiple similarity properties. A limitation of RCF, often mentioned in literature, is the challenge of acquiring historical project data about finished projects that are sufficiently similar. This limitation is mainly caused by the fact that it is impossible or impractical to collect sufficient historical data within a single company. Even when sufficient historical data could be collected, increasing the number of similarity properties reduces the size of each individual reference class to a degree that (potentially) there are no longer enough projects to ensure a good comparison. As a result, only one similarity property is used in many applications of RCF in order to keep the technique workable in a practical setting. In our research, however, these limitations are countered by allowing historical project data of different companies/industries to be used for the construction of the reference classes. This allows us to consider multiple similarly properties, and thus by definition reducing the size of each reference class, while still having sufficiently large reference classes. In order to discover the drivers of similarity between projects to improve the accuracy of RCF, a three-step approach has been applied, as described along the following lines.

Phase 1: Interviews Initially, we conducted a survey to identify properties that indicate project similarity. In the survey, we asked project managers to score different properties on their ability to measure the similarity between projects. A high score implies that the participants consider this property important for identifying similar projects, and hence it is a better property of similarity. All participants had a proven track record of relevant experience in project management and were approached by making use of the Alumni networks of Ghent University (Belgium) and the Polytechnic University of Milan (Italy). Initially, we started with a pool of over 65,000 people, afterwards narrowed down to 400 potential participants by deriving a filtered sample based on the years of experience, the function within the corporation and the specific job classification. In the end, 76 project managers had accepted the invitation to complete the survey. A total of 44 Italian project managers, with an average of 9.5 years ($\sigma = 5.31$) of experience in the field, and 32 Belgian project managers, with an average working experience of 13.6 years ($\sigma = 10.15$), participated in the study. Almost half of the participants were employed in construction, consulting, energy, or IT, and however, other industries such as aerospace, confection, and gaming were also present in the study. The broad spectrum of participants, in terms of both horizontal and vertical diversities within the organisation, ensured significant variation in the data and information obtained from the project managers. In the initial survey, the participants were asked to score 9 categories consisting of a total of 27 properties. These categories correspond with generic project features that can be used by project managers to identify similar projects. In order to increase the level of detail, we determined specific properties in these categories based on an extensive literature review. Since the identification of properties of similarity is a key aspect of this third study, the participants were also stimulated to add properties that were important in their opinion, but currently missing in the survey. This updated survey was subsequently redistributed to all participants in order to ensure that all participants could score all properties. This process of data collection was repeated over the course of 1 month and a conclusive survey with 60 properties was completed by 76 project managers. Since it would be impossible or impractical from a managerial point of view to consider 60 properties during RCF, we have limited them to the 10% best scoring properties and thus only the six most relevant properties were considered in the study. The six properties are mentioned in the last row of Table 9.2 and are briefly explained along the following lines. A full list of all 60 properties is given in Appendix C.

- Type of deliverable: The project can be a product development, a service, or a combination of both.
- Project complexity: The overall complexity of a project, independently of how often it is executed.
- Experience of company: The experience that the executing company has in performing a certain kind of project.
- Project definition: The project can be a straight redo, an expansion of an earlier executed project, or a totally new kind of project.
- Governmental law: The government can impose certain rules that can affect the project duration, cost, or other factors in the project.
- Impact on employees: The project can be mentally exhausting when there is a lot of pressure on employees.

Each property can take different values using a nominal or ordinal measurement scale and the number of projects for each class are shown in Table 9.3. Further details are discussed in Phase 2.

Phase 2: Project Analysis During the survey, project managers were requested to provide project data. In total, 52 projects in various industries and companies were collected, for which the estimated and actual cost were known. The estimated cost is defined as the budgeted or predicted cost determined at the time of formal decision to build, while the actual cost is defined as the real, accounted cost determined at the time of project completion. As a result, we could compute the forecast error for each specific project. An analysis of the forecast errors of the 52 projects showed that costs were underestimated in 63% of the projects and the actual costs were on average 16% (σ = 42.5) higher than the estimated costs. If the negative and positive forecast errors were considered separately, we noticed that the absolute size of cost underestimation (30.5%) was significantly bigger than the absolute size of cost overestimation (9.2%). We compared these findings with results observed in

Table 9.3 Absolute and relative number of projects per property value

Property	Scale	Property values		
(A) Type of deliverable	**Nominal**	**Product**	**Service**	**Combination**
Absolute (#)		7	9	36
Relative (%)		13	17	69
(B) Project complexity	**Ordinal**	**High**	**Average**	**Low**
Absolute (#)		17	25	10
Relative (%)		33	48	19
(C) Experience of company	**Ordinal**	$\leqslant$ **10**	$>$ **10**, $<$ **40**	$\geqslant$ **40**
Absolute (#)		17	18	17
Relative (%)		33	35	33
(D) Project definition	**Nominal**	**New**	**Modification**	**Redo**
Absolute (#)		19	24	9
Relative (%)		37	46	17
(E) Governmental law	**Ordinal**	**High**	**Average**	**Low**
Absolute (#)		14	23	15
Relative (%)		27	44	29
(F) Impact on the employees	**Ordinal**	**High**	**Average**	**Low**
Absolute (#)		14	19	19
Relative (%)		27	37	37

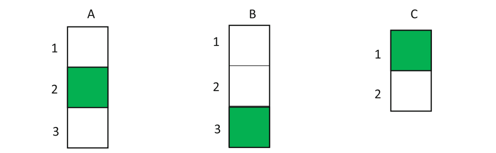
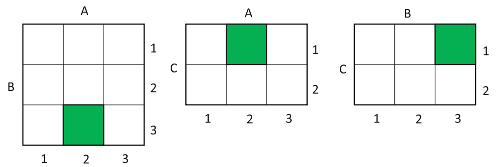
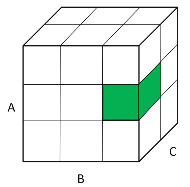
earlier research by Flyvbjerg et al. (2002) and noticed that a similar average cost overrun (27.6%) and standard deviation (38.7) was observed, which validates the findings in this research study. The values for the standard deviations in our overall project dataset initially seemed relatively large to us, but their values were expected to decrease when the projects are subdivided into groups based on the different properties of similarity. As a matter of fact, such a decrease in standard deviation is part of the idea behind RCF as it indicates that more similar projects are grouped together, which is the topic of the third and final phases.

Phase 3: Construct and Analyse Reference Classes We used the set of 52 projects to analyse the accuracy of RCF in two phases. First, an appropriate framework of reference classes to apply RCF was constructed by making use of the two following concepts:

- A *reference class* can be defined as a collection of projects that possess the same values for one or multiple properties.
- A *combination* is defined as the grouping of reference classes that are based on the same properties.

An example to clarify the previous concepts is visualised in Table 9.4. Suppose that there are three available properties A, B, and C that can, respectively, take 3, 3, and 2 values. When the three properties are considered, we can conclude that 7 combinations ($\#CO = 7$) are possible, taking 1, 2, or all 3 properties into account, resulting in 47 possible reference classes ($\#RC = 47$) corresponding to all cells visualised in the table. The table clearly shows that a reference class is specified by

Table 9.4 Formation of reference classes

#Properties	**Visualisation**	**#CO**	**#RC**
1 property		3	8
2 properties		3	8
3 properties		1	18
TOTAL		**7**	**47**

the value it assumes for a distinct property, while the grouping of multiple reference classes represents a combination. Three combinations exist using only one property (either A, B, or C) or using two properties (AB, AC, or BC) and one combination exists using all three properties.

In order to determine how many reference classes should be considered and how they must be combined, we used the six previously defined properties to obtain the best possible accuracy. Using a stepwise procedure where different combinations were tested based on a k-fold cross-validation approach, the projects in the dataset were randomly divided into different groups. Each group sequentially acts as the *test set*, while the remaining groups act as the *training set*. The training set is used to determine the accuracy of each reference class and then the test set is used to validate this accuracy by assigning these projects to the most relevant (i.e., most accurate) reference class and predict their costs. The *accuracy of each individual reference class* should therefore be determined first for the projects of the training set. It was defined as the improvement obtained in terms of the prediction error in case RCF was applied to a specific project compared to the initial prediction error (i.e., without using the RCF method). It was measured by two key metrics:

- The *intra-accuracy* is the relative improvement in the time and cost forecasts of a project when the average forecast error of *similar* projects in the *same reference class* is used as an uplift for the time and cost forecasts of the specific project.

- The *inter-accuracy* is the relative improvement in the time and cost forecasts of a project when the average forecast error of projects in *other reference classes* in the *same combination* is used as an uplift for the time and cost forecasts of the specific project.

The greater the intra-accuracy, the more significant the improvements in forecasting when applying the specific reference class. In contrast to the intra-accuracy, the inter-accuracy is expected to be very low as the correction of the initial cost estimate of the project is now adjusted using the uplift of an incorrect reference class. Since the intra-accuracy is presumed to be maximised for well-performing reference classes and the opposite applies to the inter-accuracy, the overall accuracy of a single reference class can be expressed by subtracting the inter-accuracy from the intra-accuracy.

The next step was to group reference classes into combinations. Based on the six selected properties of similarity, it was possible to establish 63 different combinations of reference classes (i.e., six combinations with one property each, fifteen combinations with combinations of two properties, ..., one combination with all properties). The results indicated that the accuracy depends both on the number of reference classes and on the combination of properties. An average improvement in accuracy of 2.41 percentage points was obtained, and however, certain combinations of properties were able to provide improvements of up to 5.47 percentage points due to positive interaction effects between the properties. We observed that the accuracy increased as more properties were added, and however, it slightly decreased when all six properties were used. More specifically, as more properties were added, the positive interaction effects between the properties increased as well. The corresponding reference classes were also narrowed down, sometimes containing not enough projects to draw valuable conclusions. We observed that combinations with five properties resulted in the highest accuracy. These experiments show that a careful selection of a relatively small number of properties (in our study, five properties were selected) may already lead to a better accuracy. The selection of a small number of properties is important since it reduces the effort that needs to be invested in the data collection. However, the experiments also revealed that this property selection should be carried out with utmost care since the performance of RCF might go down when the method is based on poor-performing properties. Therefore, we can conclude that a correct and accurate formulation of project similarity is one of the most important steps in using the RCF technique to obtain accurate time and cost forecasts for projects.

9.5 Thank You, Bent

The three studies clearly illustrated that the RCF technique contains promising ideas that can be embedded in the "classical" forecasting methods traditionally used in project management. As I mentioned earlier, I had never really thought about this

RCF technique prior to these studies because it seemed such an obvious and easy technique to me. However, I must admit that after these three studies, my interest was only further sparked, and I am therefore very much considering studying this technique further and integrating it more into my current research. What intrigues me the most is the creation of similarity properties and more specifically the definition of additional criteria to split a database of projects into similarity classes. In Chap. 14, I will elaborate on this topic and will try to start splitting projects into subprojects based on empirical data, so that each subproject will have a similar risk profile and thus potentially a similar project progress. In this so-called *calibration* approach, the definition of similarity between (parts of) projects can be further refined and possibly extends the RCF technique to simulation techniques that have already proven to be very powerful in predicting time and costs. I must therefore admit that I am extremely proud of these three studies because they do approach my research in a very different way. I am therefore flattered that one of the pioneers in the field of reference class prediction for project forecasting, Bent Flyvbjerg,[6] scrutinised our research results, and wrote the following words of praise in one of his excellent papers:

> Evidence should decide truth claims. Today, a dozen independent evaluations exist with evidence that supports the accuracy of reference class forecasting over other estimation methods, for large and small projects alike. Here is the conclusion from one such evaluation, covering construction projects:
>
> > "The conducted evaluation is entirely based on real-life project data and shows that RCF indeed performs best, for both cost and time forecasting, and therefore supports the practical relevance of the technique" (from Batselier and Vanhoucke (2016)).

References

Batselier, J., & Vanhoucke, M. (2015a). Construction and evaluation framework for a real-life project database. *International Journal of Project Management, 33*, 697–710.

Batselier, J., & Vanhoucke, M. (2015b). Empirical evaluation of earned value management forecasting accuracy for time and cost. *Journal of Construction Engineering and Management, 141*(11), 1–13.

Batselier, J., & Vanhoucke, M. (2016). Practical application and empirical evaluation of reference class forecasting for project management. *Project Management Journal, 47*(5), 36–51.

Batselier, J., & Vanhoucke, M. (2017). Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting. *International Journal of Project Management, 35*, 28–43.

[6] A few weeks before I had sent my final version of this book to my publisher, I noticed that Bent Flyvbjerg's new book had just come out. I immediately bought and read it, and I have no hesitation in recommending it to anyone interested in project management (Flyvbjerg & Gardner, 2023). Needless to say how pleased I was that our three studies on reference class forecasting are included in the reference list.

Cantarelli, C., & Flyvbjerg, B. (2015). Decision making and major transport infrastructure projects: The role of project ownership. In R. Hickman, M. Givoni, D. Bonilla, & D. Banister (Eds.), *Handbook on Transport and Development* (pp. 380–393). Edward Elgar Publishing.

Cantarelli, C., Flyvbjerg, B., & Buhl, S. L. (2012a). Geographical variation in project cost performance: The Netherlands versus worldwide. *Journal of Transport Geography, 24*, 324–331.

Cantarelli, C., Molin, E., van Wee, B., & Flyvbjerg, B. (2012b). Characteristics of cost overruns for Dutch transport infrastructure projects and the importance of the decision to build and project phases. *Transport Policy, 22*, 49–56.

Cantarelli, C., van Wee, B., Molin, E., & Flyvbjerg, B. (2012c). Different cost performance: Different determinants? the case of cost overruns in Dutch transport infrastructure projects. *Transport Policy, 22*, 88–95.

Flyvbjerg, B., & Gardner, D. (2023). *How big things get done: The surprising factors that determine the fate of every project, from home renovations to space exploration and everything in between*. McClelland and Stewart.

Flyvbjerg, B., Glenting, C., & Rønnest, A. (2004). Procedures for dealing with optimism bias in transport planning. *London: The British Department for Transport, Guidance Document*.

Flyvbjerg, B., Skamris, M., & Buhl, S. (2002). Underestimating costs in public works projects: Error or lie? *Journal of the American Planning Association, 68*, 279–295.

Kahneman, D. (2011). *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

Kahneman, D., & Tversky, A. (1979a). Intuitive prediction: Biases and corrective procedures. In S. Makridakis & S. Wheelwright (Eds.), *Studies in the management sciences: Forecasting* (p. 12). Amsterdam: North Holland.

Kahneman, D., & Tversky, A. (1979b). Prospect theory: An analysis of decisions under risk. *Econometrica, 47*, 313–327.

Leleur, S., Salling, K. B., Pilkauskiene, I., & Nicolaisen, M. S. (2015). Combining reference class forecasting with overconfidence theory for better risk assessment of transport infrastructure investments. *European Journal of Transport and Infrastructure Research, 15*(3), 362–375.

Servranckx, T., Vanhoucke, M., & Aouam, T. (2021). Practical application of reference class forecasting for cost and time estimations: Identifying the properties of similarity. *European Journal of Operational Research, 295*, 1161–1179.

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*, 289–302.

Vanhoucke, M., & Vandevoorde, S. (2007). A simulation and evaluation of earned value metrics to forecast the project duration. *Journal of the Operational Research Society, 58*, 1361–1374.

Walczak, R., & Majchrzak, T. (2018). Implementation of the reference class forecasting method for projects implemented in a chemical industry company. *Acta Oeconomica Pragensia, 26*(1), 25–33.

Part IV
About Project Data

We had a lot of project data,
but we just weren't paying much attention to their quality.

In the previous chapters of this book, I told my personal research story (and the story of my team) by distinguishing between the three main missions of academic research in project management (Part II) and the needs of the professional project managers responsible for managing projects (Part III). I wrote in several chapters about the different ways in which these two worlds—academia and practice—can be connected to improve the current knowledge about managing projects under uncertainty. I argued that project data availability is essential for developing and testing new methodologies, translating complex techniques into practical guidelines, and comparing and benchmarking current methods using computational experiments. Indeed, you may have noticed that the proper use of project data played an important role in my research story.

Part IV of this book explores this project data in more detail and aims to provide a comprehensive overview of the availability of project data, both relevant to academia and practice. I will discuss how my research team has worked for years to *generate artificial data* and *collect empirical data*, and how the search for useful project data has been more difficult than it often seems. A process for generating data (for artificial data), a process for collecting data (for empirical data), and a classification method (for both artificial and empirical project data) will be discussed, followed by a summary on how these two types of data can be connected in such a way that they become useful for academic experiments. Although it was not a conscious decision, project data generation and classification have become major themes of my research agenda and have become a quest with many difficulties, oppositions, failures, but also occasionally with some small successes that I owe to some key members of my team. Long before the words *big data* became a hype, I began generating an overwhelming amount of project data for my own academic research, not because I necessarily wanted them, but simply because I needed them for my computational experiments. The artificial generation of fictional project data began as early as the time of my PhD, but it took years before I realised that I also needed real empirical project data. Once I came to this realisation, I quickly learned that collecting empirical data is even more difficult than generating artificial data. This book part tells the story of this struggle and search for usable project data, with the goal of improving the quality of my research and trying to bring the two worlds—academia and practice—closer together. As always, it has become a challenging quest that I have undertaken together with my team.

The structure of Part IV of this book is as follows. Chapter 10 provides a basic introduction to the need for project data for academia and practice and discusses the two different types of project data. The first type of project data, consisting of *artificial projects*, is discussed in Chap. 11. This chapter summarises the data generation process that I developed during the time that I was working on my own PhD. It elaborates on this process up to and including the recent studies that I am conducting to this day with many members of my research team. I will provide an overview of the various studies to generate artificial project data and to help the research community develop more and better algorithms to schedule projects with limited resources. Chapter 12 elaborates on the use of artificial project data but now focuses on the generation of dynamic project progress data using three different generation models. Chapter 13 will discuss the second type of project data, consisting of *empirical project data* to be collected from companies with the help of the project managers in charge of the project. It will be shown that such a collection process is more difficult than it seems, and a new data collection framework will be proposed. While the previous chapter discusses the artificial project data mainly from a *static* point of view (i.e., mainly for baseline scheduling), Chap. 13 will discuss how to obtain *dynamic* project data (i.e., extended to risk and control). This dynamic data consists of periodic summary reports that measure the performance of the project during its progress and includes a variety of deviations from the original baseline schedule. The chapter discusses three methods for generating project progress data that were used in the simulation studies of the previous chapters of this book. Chapter 14 introduces the readers to *calibration methods* that aim to make use of empirical progress data to construct probability distributions for the duration of activities. These methods aim to bridge the gap between academic research and practical relevance and take into account two types of human behaviour. Three different calibration methods will be presented, and it will be shown that they allow researchers to translate empirical project data into probability distributions with an impressive 97% accuracy. Finally, Chap. 15 provides a brief overview of other project databases generated by my research team, all of which are freely available on my website for further academic research purposes.

You will discover that we now have a lot of project data and we are paying a lot of attention to their quality to ensure that they can be used for academic research.

Chapter 10
Project Data

Since the goal of the fourth part of this book is to provide a comprehensive overview of project data available in the project management literature, I think it is wise to begin with a brief introduction and a clear definition of what is meant by *project data*. To manage and control projects, project managers need data about their project to construct a baseline schedule, but they also need to collect progress data to measure the project performance during progress. Academics also need project data to test and validate their newly developed methods, and easy access to project data is crucial to any academic study. So both academics and practitioners need access to project data, but they often get this access each in their own way, without much exchange between the two worlds. The first type consists of *artificial project data* generated by researchers to conduct their research and computational experiments. These data often come from the researcher's own imagination and often have no link to the reality of real projects. The second type consists of *empirical project data*, collected by the project manager to manage the project. These data are often unstructured and occasionally miss some data points, but they do reflect how projects are managed and controlled in the real world. The duality between these two worlds during project data collection is the subject of the first section of this chapter, and a more detailed comparison between these two types of data will be discussed in Sect. 10.2.

10.1 Where Are We Now?

Before entering the exciting world of project data in the following chapters, let me briefly discuss where I believe the greatest challenge lies in using project data for research in data-driven project management. Figure 10.1 provides a concise summary of the current state ("*where we are right now*") of the use of project data for academic research and their link with practice. I created this picture for

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_10

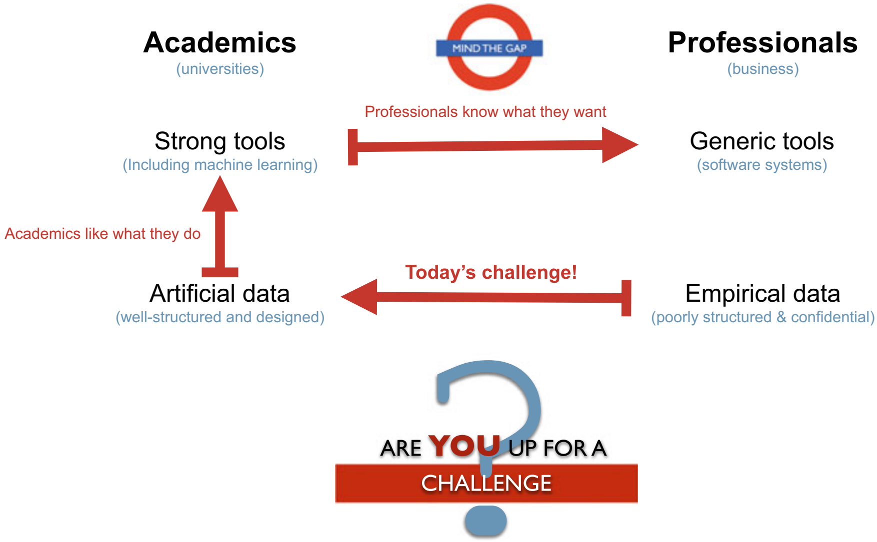


Fig. 10.1 Today's challenge in project management research

the keynote presentation that I gave at the *Creative Construction Conference* in Budapest (Hungary) in 2016 under the title *"Academics like what they do, and professionals know what they want"*, and since then, I have received quite a lot of response, from both academics and professionals, as to why the search for project data seems to be such a challenge after all. The figure is simple and does not represent much in itself, but somehow it caught the attention of a few interested parties.

The truth is that I never thought the discussion about the proper use of project data between academics and practitioners would attract so much attention, and it was during the Q&A session of this presentation that I decided to write a book about this interesting topic. I suspect that most of the attention was drawn by the fact that I insisted on the observation that academics do their best to collect project data from practice, but that these data are often not very well-structured and cannot be used for our research because they contain far too many shortcomings. *"The bridge between academia and practice"*, I argued, *"is very wide, and both sides are often unwilling or unable to understand each other's worlds"*. Such words, of course, provoke some controversy and do not really help to narrow the gap between academia and practice. I think this is why they immediately caught the attention of everyone in the audience.

I initially spoke about my own world, introducing the public to the academic research (as depicted in the left part of the picture). I talked about our research projects, and how academic researchers are developing new methodologies and new tools for data-driven project management (as I discussed in Part II of this book). I started with some studies on project control with Earned Value Management and slowly expanded to more advanced studies with statistical project control and even machine learning algorithms. I am always amazed at how much interest these studies arouse in the project management field, and I told the public that we need access to a lot of project data (*more is better*) to conduct our computational experiments in such research studies. Everything seemed very logical to everybody, until I talked about using *artificial* project data, and the advantage these data have because they can be easily generated with project-specific parameters that fit perfectly within the scope of the study. *"We are not always interested"*, I ended this part of my speech, *"whether the data can actually be used in practice. As long as we can develop strong tools (algorithms, simulations, etc.), we (as researchers) are happy with these artificial data"*.

Then I moved to the professional world (as shown in the right part of the picture) and told the audience that I am aware that project managers are not always interested in our new academic methodologies and research results, and prefer to stick to the easy tools useful for their daily project management activities (as I discussed in Part III of this book). I expressed my appreciation for the professionals because they often manage to use the best parts of academic research in their everyday practice, and I strongly believe that most commercial software tools for project management originated from the algorithms developed at many universities. There is still a lot of room for improvement of course, but I believe that the translation from theory to practice is a process that is already running at full speed (as represented in the arrow pointing to the right on the figure).

However, professionals have something that academics do not have, but want so badly: real project data! Professionals are responsible for managing real projects and therefore have access to a much richer set of project data, often with characteristics unknown to academia. As I mentioned, despite their great advantage, the empirical project data are often chaotic and not well-structured and therefore not really useful for academic research. It is not that academics have a lack of interest in using real data, but rather that they realise that the data are often simply inaccessible or unusable for their own experiments. I argued that because of all these problems, academics often do not even try to use empirical project data and therefore prefer to simplify reality by generating their own artificial project data under a well-controlled design. It should not be like this, I argued, and I presented the audience with the greatest challenge in using project data for academia (as represented in the arrow pointing to the left of the figure). If we can translate the richness of the unstructured empirical data into useful project data for research purposes, the bridge between theory and practice will become so narrow that academic research not only finds easier access in practice, but practice can be an important source of inspiration for further academic studies. In the following chapters of this book, I will therefore elaborate on the challenging and difficult task of using empirical data for academic research as well as on how to generate artificial data that better reflect real-life characteristics. Before I dive into the challenging story of project data in the following chapters, the next section will explain the important difference between artificial and empirical project data.

10.2 Two Types of Project Data

The widespread use of *artificial data* for research purposes lies not only in the fact that it is quite easy to generate these data, but especially in the ability to generate these data with predefined parameters that can be very relevant to the research study. The main goal of academic research is to develop new methodologies and test their performance on a wide range of project cases in order to look for causes of good or poor performance. Rather than presenting a methodology that can solve a problem for a *specific* project, the contribution of research often lies in demonstrating why the new methodology performs very well in some cases, but cannot compete with alternative methodologies in others.[1] Therefore, researchers test their new methodologies on a large number of fictitious projects and present *general* results that can be applied to a wide range of projects, rather than offering very project-specific solutions. This search for drivers determining the performance of the new methodologies is crucial for academic research to provide insight into the characteristics of the newly presented ideas. Their research findings can encourage other researchers to search for new developments by adding improvements to existing methodologies in cases where they fail. Consequently, researchers want access to project data that fit perfectly within the specific needs of the research question being studied and often do not feel the need to use real company-specific project data. Over the past 30 years, an overwhelming amount of artificial project data has been generated and made accessible to the project management and planning community. Chapter 11 will tell the story of my team's contribution, as well as that of some other researchers, to the generation of artificial project data for academic research. In addition, Chap. 15 will provide a complete overview of all available data for various challenging project scheduling problems.

Apart from the ease of generating artificial data in a very controlled way, there are nevertheless numerous reasons why an academic researcher should also rely on *empirical data*. The main reason is that the results of studies with artificial data are not always applicable to real-life projects and the conclusions from the studies often differ from the observations from practice. However, demonstrating the relevance of a new method in a real-life setting is key for academics to convince professional project managers to use the new methods from academia. A professional project manager often does not have much interest in the results published in journals, but rather wants to test the new methods with company-specific data in order to tailor them to the unique and specific settings of the company culture and the needs of the specific projects. Instead of providing insight into the causes of good or poor performance of the new methods, the project manager's focus is on adapting and modifying these academic methods so that they can be used optimally in a particular specific context. Without empirical project data, this translation process from theory (academic research with artificial data) to practice (professional

[1] For example, in Chap. 4, it was shown that the *earned schedule method* works very well for serial projects but fails miserably for parallel projects.

experience with empirical data) remains without useful results, sometimes reducing academic research to a theoretical exercise with little or no practical relevance. Therefore, I found it necessary to use empirical project data as well, and Chap. 13 provides an overview of the freely available database of empirical projects that I have collected with my research group over the past decade.

It is tempting to favour the use of empirical data over the use of artificial data with arguments focusing on the realism of empirical data and the limitations inherent in generating artificial data. This argument is often made by professionals who rightly argue that research should support the real needs of project managers, not the other way around. Most professionals prefer research results obtained from real data, but academic researchers often disagree and are often convinced of the usefulness of artificial data while forgetting that there is a reality behind their ivory tower. While they obviously agree that empirical projects have more realistic characteristics, it should also be recognised that using artificial data has some important advantages over empirical data. In my (non-academic) article, *"On the use of empirical or artificial project data"* published in *The Measurable News* (Vanhoucke, 2016a), I tried to convince professionals that real data and artificial data can both add value for academics and professionals alike. Figure 10.2 shows the main advantages and disadvantages of artificial and empirical data, and a brief discussion is given along the following lines.

I have already argued that the main advantage of using artificial data is the ability to generate the data according to the specific needs of the research. The use of artificial data is crucial for researchers to provide insight into the project drivers that determine the quality and accuracy of project schedules, risk sensitivity metrics, and control methodologies. It is my personal belief that the use of controlled experiments should be a key part of academic research, and these are best conducted on artificial project data generated under a well-controlled design. In this way, researchers have full control over all project parameters (controlling the network structure, time/cost structure, and resource scarceness) to obtain and present general results applicable

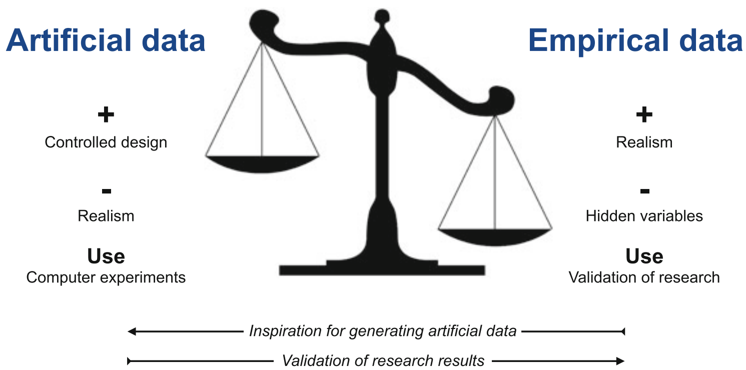


Fig. 10.2 Pros and cons of artificial and empirical project data

to a wide variety of projects. By using simulated computer experiments on artificial data, new relationships can be found between the generated project drivers and the simulated outputs that would not have been found when using empirical data, increasing the insight gained into the behaviour of planning and control techniques.

However, I have experienced that it is often extremely difficult and a waste of time to convince (some) professionals of the relevance and benefits of artificial project data. The obvious advantage of using empirical data is that they represent the real world of project management better than artificial data. It is therefore often concluded that empirical data are always preferable to the use of artificial data. However, I believe that the use of empirical project data is not without danger for academic research. It should be remembered that the ultimate goal of research on integrated project management and control is to improve the decision making process during project progress. These studies focus on evaluating current methods and presenting new techniques for project control that can be used as triggers for corrective actions to get projects at risk back on track. These warning signal triggers should be used carefully and should allow the project manager to take actions *only* when they are really necessary (as discussed in Chaps. 5 and 8). Control methodologies should only provide warning signals when the project is highly likely to get out of control, not warning signals for every small change in the project with a small impact on the project objectives. Therefore, the main aim of these studies is to contribute (directly or indirectly) to this challenging objective by proposing methodologies to better monitor ongoing projects *and* improve the corrective action decisions. However, the major and inherent weakness of the empirical data lies in the fact that these project data already *include* many of these corrective actions, making it difficult to know whether the project was progressing with or without actions. If no explicit distinction can be made between project data and management actions, computer experiments cannot cleanly interpret the accuracy and quality of warning signals from control methods, and it is difficult to make unambiguous judgements about the quality of control systems. Therefore, I believe that simulation experiments on empirical data do not always provide the necessary insights into the relationship between project drivers and the performance of new project control methods.

This is not to say that I think researchers should rely solely on artificial project data and be blind to real empirical projects. Including an empirical project dataset in research increases the likelihood of producing relevant results and reduces the risk of obtaining artificial results with little or no practical relevance. Therefore, I personally believe that academic research results should first be obtained from experiments on artificial datasets and then validated on empirical projects in a second phase to assess their realism and their potential use in practice. Ideally, such an approach leads to practical guidelines and insights relevant to real projects, but also to general insights from experiments on artificial data that can be easily used in new follow-up studies. Consequently, I believe that the main advantage of empirical data for academic research lie in the *practical validation* of artificial research results, rather than in the *creation of new insights* obtained only from empirical data.[2] When the two types of data are both used in project management research, academic learning and professional relevance are brought together with the aim of closing the gap between theory and practice to support better decisions for projects. The proposed data approach seems to me to be the ideal way to satisfy the two worlds of project management. I therefore wish the readers an enjoyable and engaging journey into the world of artificial and empirical project data in the following chapters.

Reference

Vanhoucke, M. (2016a). On the use of empirical or artificial project data. *The Measurable News, 2*, 25–29.

[2] I am fully aware that many researchers do not share my view, and I am therefore not blind to their arguments as to why my approach is not the only possible way to conduct sound academic research. I just want to say that this approach (artificial results first, then empirical validation) works best for me and my team.

Chapter 11
Artificial Projects

I have always experienced academic research as a very challenging search for new ideas, often not knowing the goal of the search in advance, but exploring all kinds of new avenues until I reach new insights. Research has never been an act of observing real business needs or existing phenomena, but rather a discovery of patterns in artificial data and seeing where it takes me. Of course, research is ideally inspired by real observations or conversations with professionals, but any idea that comes from my own imagination is also considered and often leads to an inspiring search. For this reason, *generating artificial project data*, rather than observing real data, has always been crucial in most of my research, and I have often gone very far in doing so: When I could not detect desired patterns in the artificial data, I mostly blamed it on the data (and rarely, if ever, on my own failure), and most of the time I decided I simply had not generated enough data to make a real breakthrough. In that case, I generate more project data and keep searching for something I do not yet understand. For me, research is and always will be exploring uncharted territory, and if I knew in advance that what I was doing made sense, I would no longer find research challenging at all. The American statistician and artist Edward Tufte once expressed the quest for clear results as follows:[1]

> To clarify, add data.

Academics are obviously in a luxury position. They do not always have to take practical needs and requirements into account and can therefore simply generate more and more fictitious projects. To explore new research avenues, academics have generated an overwhelming amount of project data in the past, and it seems they just cannot get enough of it. In the last few decades, a huge number of project databases have been published in the literature, and it is becoming difficult to get

[1] I often refer to Edward Tufte in my classes and like to mention that, like Albert Einstein, he was born on the 14th of March. Then I tell my students—a fun fact—that I am born on the 14th of March too, by which I mean nothing of course:-).

M. Vanhoucke, *The Illusion of Control*, Management for Professionals ,
https://doi.org/10.1007/978-3-031-31785-9_11

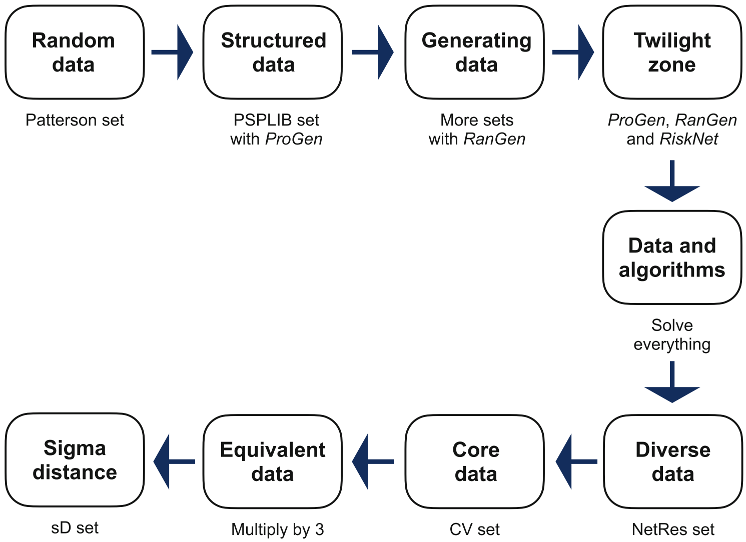


Fig. 11.1 Artificial datasets in nine chapter sections

a clear overview. I am certainly partly responsible for this, so in this chapter I want to tell the story of project data generation, so that the readers not only get an overview, but also understands the bigger picture. As I mentioned earlier, the luxury of an academic research job lies in the ease with which artificial project data can be obtained. However, generating artificial data is not as easy as some may think, and the generation and classification of artificial project data are a research project in itself. Since artificial data generation has always been a central theme in my research career, I like to tell the data story through the next nine sections of this chapter summarised in Fig. 11.1.

The origin of artificial project data generation lies in the collection of *random data* and the additional generation of *structured data* by well-known researchers, and this process will be discussed in the first two sections of this chapter. I only came into the picture much later when, during my PhD research, I felt the need to generate my own project data that fitted my research better than the existing datasets. This search for different and better project data starts in Sect. 11.3 and is continued in all sections thereafter. It all started with the development of the project data generator *RanGen* together with my PhD supervisor (Erik Demeulemeester) and co-supervisor (Willy Herroelen). This data generator literally changed my academic life, because from then on I have been busy generating more and better project data. A year after my PhD, I came into contact with José Coelho from the Universidade Aberta (Lisbon, Portugal) at a workshop in Valencia (Spain), and when I found out that he too was working on a data generator (*RiskNet*), I proposed to him to extend my *RanGen* data generator together (which eventually became the *RanGen2* data generator). Little did we both know then that this accidental collaboration would lead to years of work on project data and a close friendship that satisfies me to this day. Our initial research took us into the *twilight zone* between the various existing data generators, and we decided to compare them based on a number of criteria (Sect. 11.4). Afterwards, we continued our work by running exact branch-and-bound algorithms to optimally solve the *resource-constrained project scheduling problem* on all kinds of projects (Sect. 11.5). Since the results of that study were somewhat disappointing, we continued the search for new data that led to a new set of diverse projects (NetRes set in Sect. 11.6) and a set of very hard-to-schedule projects (CV set in Sect. 11.7). The idea of generating very difficult projects came to us because we wanted to challenge academia to look for the core of the complexity of project scheduling problems. This research was followed by splitting each existing dataset into three similar sets of *equivalent projects*, which suddenly tripled the already large number of available artificial projects (Sect. 11.8). Our final study involved generating a new artificial project dataset that could be used for testing meta-heuristic project scheduling algorithms (rather than exact algorithms), which resulted in the sD set discussed in Sect. 11.9.

This collaboration with José has given me enormous pleasure, and so I am very proud to summarise our challenging quest in this chapter. As I mentioned earlier in this book, our collaboration has also given me the chance to live in Lisbon for two years (2015 and 2016) and has given José the opportunity to start as a visiting professor at Ghent University (Belgium). During this research, I completely lost my heart to this beautiful capital of Portugal, and in 2022, I even decided to settle there for a few months a year and bought my own modest apartment. Who knows what new research ideas we will develop that may be the subject of a new book sometime in the future. But now let us talk in the next eight paragraphs mainly about the research that has already been done on artificial project data.

11.1 Random Data

The majority of the project scheduling research has focused on the well-known and challenging *resource-constrained project scheduling problem*, abbreviated as the RCPSP, that I introduced earlier in Chap. 3 (*solution methods*) and Chap. 6 (*machine learning*) of this book. It is a very challenging project scheduling problem because it is difficult to find an optimal solution to this problem, and academics express this difficulty in mathematical language and say that this problem is *NP hard*. An optimal solution to this problem consists of a project schedule in which each activity has a start and end time, such that the total duration of the project (i.e., the project makespan) is minimised. These start times of activities should be allocated taking into account both the relations between activities (i.e., network logic) and the limited availability of resources (i.e., resource constraints). Since its introduction into the academic community, researchers have spent much time developing algorithms to solve this problem and along with better solution procedures came better and more diverse artificial datasets to test these algorithms. At the beginning of this research, not much project data were available, so researchers each created their own small project to test their algorithms. After a while, there were a lot of research papers showing a few projects until a researcher from the Indiana University Bloomington (US), James Patterson, had the brilliant idea of collecting these different projects into one set. This so-called *Patterson dataset* is thus a collection of fictitious project examples collected from various papers in the literature, consisting of 110 projects with limited resources. This set has no specific structure and was not designed with input parameters such as the number of activities or scarcity of resources, but it has nevertheless been the primary source for testing new procedures for a very long time. Most early algorithms to solve the RCPSP were tested on the Patterson instances, and researchers now often refer to the *Patterson format*[2] as the text format used for project files. We, researchers, owe a lot to James' efforts at the time as he is the *father* of artificial data generation! The Patterson format was, and still is, used by many academics, and even today many of the new artificial project data are generated in this widely known format. After a few years, the importance of the Patterson set declined, and researchers used the data less and less, until it was widely accepted that the 110 project instances could be treated as *easy* projects that could all be solved to optimality with today's sophisticated algorithms. The time had come to replace this dataset with new and larger projects, which was exactly the idea of a young researcher from Germany, as discussed in the next section.

11.2 Structured Data

Rainer Kolisch of the TUM School of Management (Munich, Germany) was one of the first researchers to recognise the need for a larger dataset for research into project scheduling, and he was well-aware that a structured generation of such a set was better than simply collecting random data. His important contribution fundamentally changed the way many researchers test and compare algorithms, and his research on project data has been the start of much additional data research, as shown in Fig. 11.2. The figure shows a timeline of key milestones in research on data generation for the resource-constrained project scheduling problem. Of course, like many sections in this book, this figure does not contain a complete overview of all research but reflects my own contribution in (and thus biased view of) this fascinating field of research. Nevertheless, I have tried to add the main references in the literature on artificial data generation to give you an almost complete picture of this field. This figure is only the first part of two related figures, and an update is given in Fig. 15.1 of Chap. 15.

The figure starts in the year 1995, the year in which Rainer proposed his new data generator *ProGen* to generate his well-known PSPLIB dataset (published two years later) that completely replaced the *Patterson* dataset in just a few years and

[2] For further details, go to Appendix D.

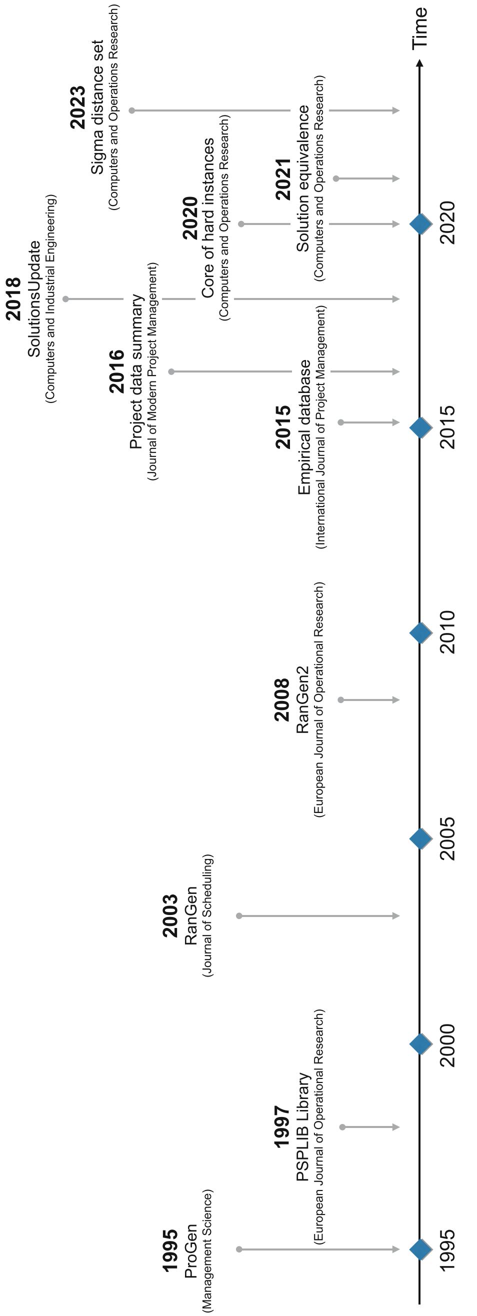


Fig. 11.2 Overview of research on project data (part 1)

is to this day the most commonly used dataset in project scheduling. As a young PhD student, he worked on fast heuristic algorithms that used simple priority rules to solve the RCPSP with the aim of constructing feasible, but not optimal, project schedules without over-allocation of resources. Such priority rule-based algorithms are *fast* and *easy* and generate a feasible schedule in fractions of seconds. They cannot compete with the much more sophisticated algorithms developed in recent decades to create project schedules, but many of today's software tools (such as the well-known MS Project but also my own tool ProTrack) still rely on these algorithms to create project schedules for large projects in no time.[3] Since priority rules are only relevant for scheduling very large projects, Rainer must have realised that the Patterson projects are far too small. I suspect he needed bigger projects to test his algorithms and so decided to generate a new artificial project dataset under a controlled design. In 1995, he published an article in the leading journal *Management Science*[4] (Kolisch et al., 1995) in which he *characterised* and *classified* artificial project data based on a new network generator *ProGen*. Not much later, he used his generator to generate the new PSPLIB dataset (*Project Scheduling Problem LIBrary*) that I discussed in the beginning of this section. This set is up to today the most widely used artificial dataset in the project scheduling literature. The following sections explain his approach to generate and present new artificial project data using the following three keywords:

CLASSIFICATION → GENERATION → DATABASE

Data Classification In contrast to the collection of random artificial project data (like the Patterson projects) or empirical project data from the business field (as discussed in Chap. 13), the artificial data generation must be carefully designed to ultimately distinguish between different project types. Therefore, the generation of artificial data should not be random but should follow a so-called *full factorial design* methodology to control as many relevant input parameters as possible. Such an approach allows the researcher to generate the *full range of complexity* in the project data, which basically means that any project network that could possibly exist in reality must also be in the dataset. To achieve this goal, this generation process requires an a priori definition of project parameters to characterise the project and ultimately the hardness of the scheduling problem (i.e., the construction of a resource-feasible project schedule). In other words, the primary requirement of an artificial project dataset is that it is diverse enough such that researchers, who rely on the data to validate their project scheduling algorithms, are able to distinguish between *easy* and *difficult* project instances. As such, they can provide insight into how and why some projects are different (more complex) than other (easy) projects, and how to focus future research on the unsolved challenging

[3] Recall that I used priority rules in the machine learning research project of Sect. 6.1.

[4] It is well-known that when you publish a paper in *Management Science*, your academic life will never be the same. It is like getting an Olympic medal for sports people: It changes everything.

problems (by leaving the easy projects outside the scope of future research). For the research on algorithms to solve the *resource-constrained project scheduling problem*, a number of project indicators were proposed that can be divided into two clusters, as follows:

- *Network topology indicators*: The topological structure of a project is determined by the specific set of activities and the precedence relations between them. The way the activities are linked by these precedence relations determines the logic and structure of the project network, which can vary from a fully serial network to a fully parallel network. To generate projects with different values for the serial/parallel structure, different network topology indicators have been proposed in the literature, such as the *coefficient of network complexity* (CNC), the *order strength* (OS), or the *serial/parallel indicator* (SP). Note that the SP indicator has been used earlier in this book to show that project control works better for serial projects (high values for SP) than for parallel projects (low SP values) (Chap. 4) and for Bayesian network forecasting models of Chap. 6.
- *Resource scarceness indicators*: Since each activity of a project must be performed by a set of renewable resources (people, machines, etc.) with limited availability, the projects must also be classified into clusters with similar activity resource requirements and availability of the renewable resources. This classification is done using resource scarcity parameters for quantifying and controlling the relationship between these resource requirements and availability. Several resource indicators have been proposed in the literature, such as the *resource factor* (RF), the *resource scarceness* (RC), the *resource strength* (RS), and the *resource use* (RU).

I have deliberately chosen not to provide a discussion or technical details and formulas for these network and resource indicators, but I will give some additional details in the following chapters where necessary. The impatient readers can already find a brief summary in Appendix E where the most commonly used indicators are explained in detail.

Data Generation The indicators for measuring the network topology and resource scarcity of projects are especially important to generate new artificial projects where the values of these indicators (which usually vary between 0 and 1) are set to different values. Some researchers have proposed project network generators to create artificial data using some of these network and resource parameters, but the number of articles presenting these generators is rather sparse. These generators are called *network* generators since each project consists of a sequence of activities with precedence relations between the activities to model the network topology. Most generators also add resource data to the project networks, but generating such data is much simpler than generating the network. Therefore, these generators are not called *network* and *resource* generators, although most generate both types of data. Most generators rely on the well-known *activity-on-the-node* network format in which each activity is a node and each precedence relation between a pair of activities is represented by an arc between these nodes. Ideally, a generator should be *strongly random*, meaning that every possible network with a certain number of nodes and arcs should have an equal chance of being generated from the space of all feasible networks. Unfortunately, most of the generators suggested in the literature do not have the ability to be strongly random and therefore just generate as many networks as possible in a limited time without guaranteeing that every possible network can be generated.[5] The project network generator *ProGen* was one of the first generators for activity-on-the-node networks, developed by Rainer Kolisch and his co-authors. In this generator, the network structure is controlled by setting the previously mentioned *coefficient of network complexity* to predefined values. This CNC indicator simply measures the number of arcs divided by the number of nodes, and it is now known that this indicator is not a very good indicator for controlling the network complexity. In addition, a number of other parameters, such as the minimum and maximum number of successors and predecessors for each activity, were predefined in order to control the artificial process of generating project networks. Finally, the generator has also incorporated the resource scarcity indicators discussed earlier to include resource data in each project network using the *resource factor* and the *resource strength*.

Project Database Two years after the introduction of the *ProGen* network generator, Rainer and his co-authors decided to generate a new set of projects under a well-controlled design with different values for their network and resource indicators. It resulted in the well-known PSPLIB set that I discussed earlier, which quickly replaced the old Patterson set and became the standard dataset for testing algorithms for resource-constrained project scheduling problems. The artificial project dataset consists of four subsets (with projects of 30, 60, 90, and 120 activities) with different values for the CNC (equal to 1.5, 1.8, and 2.1) and the RF (equal to 0.25, 0.50, 0.75, and 1). There is no doubt that this dataset paved the way for many researchers to develop and test new algorithms in an easy and standardised way, making comparisons with other algorithms much easier. The success of the PSPLIB dataset clearly demonstrates the relevance of artificial data to academic research, and I am convinced that research over the decades would not have made such great progress without the PSPLIB dataset. I mentioned earlier that the importance of artificial project data is often not well understood by professionals, but for academics it is their main source of data for their research. Comparing algorithms and new research results in a standardised and fair manner that helps elevate current state-of-the-art knowledge and contributes to a better understanding of a vast amount of procedures and algorithms to schedule projects under different situations. It shows that—despite the importance and necessity of having access to empirical project data—nothing can compete with the vast availability of artificial project data generated under a controlled design. More than 25 years after its introduction, PSPLIB is still the standard set for most project scheduling researchers. Along with

[5] As far as I know, only one network generator is strongly random, but it generates networks in the *activity-on-the-arc* format (Demeulemeester et al., 1993), which is less commonly used in the project scheduling community.

this new set, Rainer and his co-authors also proposed some criteria to establish a fair evaluation between different project scheduling algorithms, such as using a 5000 schedule stop criterion when population-based meta-heuristics are used. In addition, he also created a website with the best-known solutions for the PSPLIB instances, encouraging researchers to download the benchmark sets to evaluate their algorithms and upload their results to the library. To date, not all currently found solutions could be confirmed as the optimal ones, despite the rapid increase in computing speed over the years. It shows the impressive importance of Rainer's work, and we should be as grateful to him as we should be to James' work when he proposed the Patterson dataset.

11.3 Generating Data

Despite the importance and undeniable popularity of the PSPLIB set for academic research, it is in the nature of every researcher to question existing knowledge and look for possible improvements. I also had that need in 1996 when I started my PhD, and together with my advisors Erik Demeulemeester and Willy Herroelen, I jumped into the challenging research domain of artificial data generation with the aim of proposing better alternatives to the PSPLIB dataset and the *ProGen* network generator. The result was the new *ran*dom network *gen*erator *RanGen* (published in 2003 as shown in Fig. 11.2) and an improved version *RanGen2* (published in 2008 with José Coelho) that I used over the past two decades to generate much more artificial data.

The first ideas to develop our own network generator *RanGen* came from some interesting research studies that revealed some weaknesses in current network and/or resource indicators. More specifically, some studies questioned the value of some of the most commonly used indicators as they were criticised for failing to properly distinguish between easy and difficult project instances. These studies therefore presented improved indicators to address the shortcomings in the existing indicators. For example, it has been independently shown by several researchers that the CNC network indicator cannot discriminate between easy and hard instances, and therefore, it is not appropriate to use this indicator as a good measure for describing the impact of the network topology on the hardness of a project scheduling problem. Similarly, it has been shown in the literature that the *resource strength* (RS) sometimes fails to distinguish between easy and hard project instances, while the alternative *resource constrainedness* (RC) continues to do so. Such studies brought us to the question of why and how one indicator should be called "*better*" than another indicator, and the answer could be found by explaining the concept of *phase transitions*. A phase transition is usually used to refer to a process in which something changes from one state (e.g., liquid) to another (e.g., solid), and the typical example is water turning into ice at zero degrees Celsius. This concept has been used in the project scheduling literature by Herroelen and De Reyck (1999) and refers to a sudden shift for a specific value of an indicator that transforms the problem from easy to difficult. In their research, the authors draw attention to the importance of indicators with sufficient distinctiveness to determine the complexity of project scheduling problems. More specifically, researchers need to know the values for the indicators for which a project instance moves from easy to difficult (or vice versa) similar to the temperature value of zero degrees Celsius for turning water into ice.

Based on some interesting results from these studies, we decided to build our own network generator (*RanGen*) capable of generating thousands of project networks with different order strength (OS) values to maximise network diversity. In addition, resource data are added to the networks for given values of the resource constrainedness (RC), because both indicators (OS and RC) are known to be much more reliable than other existing network or resource indicators. A first version of this generator was published in 2003 in the *Journal of Scheduling* (Demeulemeester et al., 1993) in which we were able to show that the generator can generate a much wider spectrum of possible networks than the existing network generators at that time. Five years later, we extended the generator to a second version (*RanGen2*), which will be discussed in later sections. This network generator has led to many pleasant benefits within my OR&S team. First of all, it has given us the ability to generate a lot of project data for a wide range of project scheduling problems for which I will provide an overview in Chap. 15. Moreover, somewhat to my surprise, I experienced that the use of network generators itself received attention from the non-academic field. More specifically, it attracted the attention of a Brazilian publisher who I met while I was on a teaching tour.

Indeed, in 2016, I was in Brazil to teach project management to several companies in São Paulo, Rio de Janeiro, and Curitiba led by Osmar Zózimo De Souza. After the lectures, we took some time to drink some beers and do some sight-seeing, and he introduced me to a new concept that he called "*a day in the life of Zózimo*".[6] During such a day, I basically did everything Zózimo usually does on an ordinary Brazilian day, such as drinking a cup of coffee on his terrace, eating an ice cream when it gets too hot, and working in the office as a publisher of various magazines. I heard that he was the owner of MundoPM and an editor at the *Project Design Management Magazine* and the *Journal of Modern Project Management*, and he asked me if I was interested in publishing an article on artificial project data in the latter magazine. Since I had already generated a lot of data with the *RanGen* and *RanGen2* generators, I immediately accepted the offer and started working with José Coelho on the article, which eventually resulted in the article published in Vanhoucke et al. (2016) and a TV interview in Portugal. The article contains a summary of 10 datasets available in the literature with over 20,000 projects, but it quickly became obsolete. After all, after 2016, the number of generated projects continued to grow, leading to many more datasets with a variety of projects for different research purposes (cf. Chap. 15) and the end does not seem

[6] We had such a good time together that I discussed my friendship with Zózimo in my book "*The Art of Project Management*".

to be in sight yet. In the remaining part of this chapter, I will provide an overview of this continuous growth of artificial project data, but first I want to take the readers into the study of the next section where we examined and compared the different network generators available at that time (*RanGen* and *RanGen2*, but also other existing generators).

11.4 Twilight Zone

I personally believe that the biggest advantage of the *RanGen* network generator is that it can generate a lot of *different* networks in a fairly short time. This is because the generator contains a fast and efficient procedure to check whether each newly generated network is different from a previously generated network, which ensures that the set of generated networks contains only unique networks. Despite this uniqueness check, it still does not guarantee that *any possible network* can be generated from the space of all networks. Such a guarantee can only be given theoretically when the network generation process is *strongly random* where every possible network is generated with equal probability (as discussed earlier). While this concept also cannot guarantee that every possible network will be generated (simply because the space of all possible networks is too large), it does give an idea of how big that space really is. Although I initially thought that I could make *RanGen* strongly random, I have to admit that it never worked. So I switched to a fast generation of networks to generate as many networks as possible in a short time to have a rough estimate of how big that space could be. And yet that feeling continued to gnaw, as I still did not know whether it could generate enough different networks. That was when I got to know José. I knew that José also wanted to know more about network generation because as a PhD student he had worked on his own network generator *RiskNet* together with his advisor Luis Valadares Tavares. He had developed six topological network indicators, named I_1 to I_6,[7] to describe the detailed structure of a project network (Tavares et al., 1999). At that time, I was only familiar with the CNC (used in *ProGen*) and OS (used in *RanGen*) to classify the network topology for projects, and I had never heard of these six network topology indicators. When I met José for the very first time in Valencia, we decided to make a new version of *RanGen* including these six indicators. Our goal was to better understand how network generators could cover the entire space of all possible networks, even though neither of us had any idea how we would approach such a study.

Initially, we used three different network generators to obtain a very large number of networks, and we decided to make a comparison between these generators based on all these networks. The *ProGen* data generator used the CNC as an indicator

[7] The choice of names was a bit unfortunate and we later made the names more understandable, as can be seen, e.g., in Table 11.1 for 4 of these 6 indicators.

to measure the network topology, while the *RanGen* generator used the OS. José's *RiskNet* generator made further use of the six indicators presented before (I_1 to I_6), yielding a total of *eight* different indicators to describe the topology of a network. And so it happened that we started thinking of plotting each network in eight dimensions, where it has a value for each one of the eight indicators. A small change in the value of one of these indicators already puts us in another place of this 8-dimensional space, giving us a new project network that is different than the first one. Thinking in multiple dimensions is not easy, and it has played tricks on us at times, but once you realise that multiple dimensions exist, you cannot go back to the early days of one dimensionality. It is just like Oliver Wendell Holmes, an American lawyer who served as a judge of the United States Supreme Court, said:

> A mind that is stretched by a new experience can never go back to its old dimensions.

There was no way back. That is why we decided to incorporate these eight indicators into our updated version of the *RanGen* generator to get an improved version (*RanGen2*, Vanhoucke et al., 2008) that generates networks for given values of the I_2 indicator (instead of the OS for *RanGen*) and then automatically calculates all seven other network indicators for each network. The move from OS to I_2 was a very pragmatic choice, as we thought the latter could describe the serial/parallel structure of a network in a much simpler way than the OS. Apparently, it was not a bad choice because much later the I_2 indicator was renamed the SP indicator, which has already been used in several places in this book and has shown to be able to measure the performance of control methods in a very accurate way (as discussed in Part II of this book). With the four network generators at our disposal (*RanGen*, *RanGen2*, *ProGen*, and *RiskNet*), we generated as many 30-activity networks as possible to find out how many *different* networks each generator could create in order to get an idea of the full space of all possible networks.

In order to define that one network is different from another one, we only used the 6 indicators of José and ignored the CNC and OS (and so worked in a 6-dimensional space instead of an 8-dimensional space). Furthermore, since the I_1 indicator measures the number of activities in the project, we fixed it to a value of 30 (which reduced the space to a 5-dimensional space). We defined a *"new"* network as a network with a different I_2 to I_6 combination than any previously found network, so we created a matrix of 30×101^4 cells where we initially set each cell to zero. From the moment the network generator finds a network with known I_2 to I_6 values, the corresponding cell in this matrix is set to 1. The size of the multidimensional matrix of 30×101^4 was of course not arbitrarily chosen. Given this I_1 value of 30, the I_2 indicator can have only 30 possible values, ranging between 0 and 1, indicating the proximity of the project network to a fully parallel or serial network. The other I_3 to I_6 indicators can have any fractional value between 0 and 1, and we created 101 cells for each indicator (i.e., we worked in 0.01 increments). It is important to realise that the total number of cells in this multidimensional matrix is equal to $30 \times 101^4 = 3{,}121{,}812{,}030$, but not every cell in this 5-dimensional matrix has an existing network. We could nevertheless use this number to see how many networks each network generator could generate. All we wanted to do was setting up a comparative study between the four network generators to see how the generated networks differ from each other in this multidimensional space. The computer started working and generated a huge amount of networks, and to prevent it from generating forever, we used the following two stop criteria for each network generator:

- Maximum allowable CPU time of 1 network of 100 seconds: If no network could be found within this time, the generator was restarted with new input values to search for other networks.
- Maximum of 1000 consecutive generations without finding a new network: If a network generator continued to generate networks with the same value for all I_2 to I_6 indicators, we suspected that it would not be very likely that other *new* networks would be found.

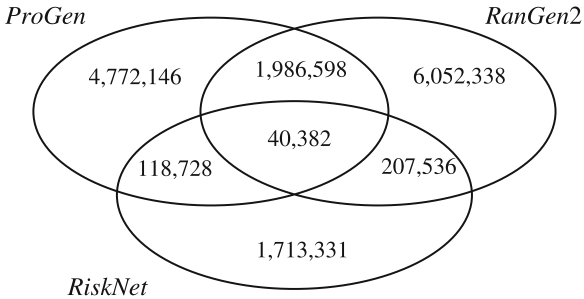
After a long series of experimental runs on different computers[8] (where one computer had heating problems, the other suddenly crashed, and my brand new laptop almost caught fire), we were finally able to generate a total of 19,105,294 different networks. A total of 19,105,294 networks sounds like a lot, but it only covers 0.61% of the 3,121,812,030 cells in our multidimensional matrix, and this percentage would probably drop even more drastically if we took more than 5 indicators into account. Indeed, creating such a multidimensional matrix to define similarities of project networks is like working in hyperspace, and strange things are known to happen in hyperspace beyond our intuition. In his book *"The Master Algorithm"* (Domingos, 2018), Pedro Domingos stated that hyperspace is like the *Twilight Zone* where the intuitions that we have of living in three dimensions no longer apply, and where weird things start happening. I really like the author's reference to an orange, as he wrote:

> Consider an orange: a tasty ball of pulp surrounded by a thin shell of skin. Let's say 90 percent of the radius of an orange is occupied by pulp, and the remaining 10 percent by skin. That means 73 percent of the volume of the orange is pulp (0.9^3). Now consider a hyperorange: still with 90 percent of the radium occupied by pulp, but in a hundred dimensions, say. The pulp has shrunk to only about three thousandths of a percent of the hyperorange's volume (0.9^{100}). The hyperorange is all skin, and you'll never be done peeling it!

The number of projects generated was compared for each network generator and shown in Fig. 11.3 (omitting the *RanGen* generator because the *RanGen2* generator gave better results). The results can be summarised as follows: *RiskNet* is not able to generate many networks (compared to the other two generators), but nearly 78% of the networks found by *RiskNet* were not found by any other network generator, which makes this generator special. However, this fraction only accounts for 8.97% of the total number of networks found. Our *RanGen2* performs best both in the number of networks and in the number of networks not found by other generators.

[8] At that time, we had no access yet to the supercomputer infrastructure at Ghent University that I introduced in a footnote in Chap. 5.

Fig. 11.3 Total number of generated networks



ProGen is also a significant contributor to the total number of networks generated, and about 50% of the networks found by both generators were completely new (nearly 32% of all networks were generated only by *RanGen2* and nearly 25% of all networks were generated only by *ProGen*).

The number of generated projects were compared for each network generator and shown in Fig. 11.3. The results can be briefly summarised as follows: *RiskNet* is able to generate a large set of networks that were not found by any other generator. Indeed, almost 78% of the networks found by *RiskNet* were not found by any other network generator. However, this fraction counts only for 8.97% of the total amount of networks found. *RanGen2* performs best, followed by *ProGe*n, and both have a high contribution to the total amount of networks generated. Approximately 50% of the networks found by both generators were completely new. Almost 32% of all the networks have only been generated by *RanGen2*, and almost 25% of all the networks have only been generated by *ProGen*.

This study was published in Vanhoucke et al. (2008), and this experiment shows that generating artificial networks to cover the full range of complexity is not an easy task. None of the network generators is capable of generating all the networks generated by the others, and it therefore seems impossible to capture the full space of all possible networks by a single network generator. Nevertheless, I believe that our focus on generating as much diversity as possible in the project networks is a good thing as it creates a broader set of project data and therefore increases the likelihood that every possible network is in the generated set. This is of utmost importance for researchers who want to develop and test new algorithms on project data and present general results. The more diverse the dataset, the more likely that some algorithms will not be able to solve some of the projects generated, which may incentivise other researchers to develop new alternative methods for solving project scheduling problems. Having access to project data and a tool to share obtained results with other researchers is therefore essential for academic research. I invite the readers to take a look at the next section where solutions are generated for a large set of projects using a branch-and-bound algorithm, and a new tool is proposed to share these solutions with other researchers.

11.5 Data and Algorithms

After a number of successful research projects with José, we felt that it was time to return to our roots. We had spent a lot of time generating artificial project data in the last few years and had developed all kinds of algorithms for various *extensions* of the *resource-constrained project scheduling problem*. Moreover, in the past years, I had shifted the focus with my OR&S team in Ghent (Belgium) somewhat from project scheduling to project control (discussed in the previous chapters) and was less and less concerned with developing algorithms for project scheduling. And yet, José and I never lost interest in the basic version of the resource-constrained project scheduling problem (without all kinds of extensions) and kept following the literature closely. We noticed that most research studies now mainly proposed meta-heuristic algorithms that were fast and efficient but were unable to find optimal solutions (project schedules with the minimum project duration). Despite the usefulness of such algorithms, we found that these studies often do not provide a deeper understanding of the complexity of the problem, which was the primary reason why we had generated so much artificial data. Therefore, we reviewed the literature on exact methods, which mainly consisted of branch-and-bound algorithms that can optimally solve the RCPSP and decided to program them all in C++. The idea was to solve all PSPLIB projects (and the other projects from other datasets) to optimality, which did not look like a very difficult task as both José and myself had worked on that challenging problem during our PhD period many years earlier. Most branch-and-bound algorithms originated from that era (around 2000 or earlier), so it should be quite easy to optimally solve all the projects in 2015, more than a decade and a half after these procedures had been developed. After all, we now had access to much faster computers and had built up a lot of experience to efficiently program the existing algorithms. I therefore quickly decided to move to Lisbon for a few years to code all existing branch-and-bound procedures available in the literature and then, in one run, optimally solve all project instances of the PSPLIB set, and that was it. Our enthusiasm was endless, and the motivation for this research project originated from a number of observations of the research results published over the decades:

- All 110 Patterson instances as well as the PSPLIB instances with 30 activities (the so-called J30 instances) were once very difficult but could now be optimally solved with these existing branch-and-bound algorithms and are therefore now considered easy instances.
- Since most branch-and-bound procedures were relatively old and thus tested on slow computers, a new test would generate much better solutions. We were therefore convinced that the larger projects from the PSPLIB set (J60, J90, and J120 instances) could now also be solved to optimality.
- Each branch-and-bound procedure from the literature had tried individually to optimally solve as many projects as possible, but there had never been an attempt to combine the different components of each algorithm. Each algorithm had strengths and weaknesses, and the integration into one large hybrid algorithm

would therefore magnify the individual strengths, which undoubtedly would generate much more optimal solutions.

After an intense study to compare and better understand the existing branch-and-bound algorithms, we started coding. José is a master at writing efficient programs and in less than a year had put all the procedures into one big algorithm that we called the *composite lower bound branch-and-bound procedure* (published in Coelho & Vanhoucke, 2018[9]). This hybrid procedure not only contains the best-functioning components from the various separate procedures, but also contains many lower bounds and a credit system that steers the search for optimal solutions as well as possible. This procedure was eventually used to solve all PSPLIB instances, and after a series of computer tests, sometimes allocating more than one day of computing time per project, we finally looked at the results in the hope that all projects would have been solved to optimality.

And along with the results, came the disappointment.

We were surprised to see that the newly developed composite algorithm could not solve many more projects to optimality, as we initially thought before the start of our research. The J30 instances were indeed easily solved, but a significant number of the J60 and J90 projects, and most of the J120 projects remained *open* instances (i.e., these instances could be solved, but we had no guarantee that the solution is optimal). Even when we repeated our computer experiments, this time with a total duration of several months on the supercomputing infrastructure of Ghent University, we did not get optimal solutions. And so, to our regret, we had to conclude that small projects with 30 activities (J30 instances) are easy, but larger projects (with 60 and more activities) are still, after all these years, very difficult to solve, despite the much faster computer speed. *Size apparently does matter!*

Obviously, we were not going to give up just like that, and we decided to generate 500 new projects with 30 activities using our *RanGen2* generator to see why these *"small"* projects are so easily solvable. After another series of tests, we found (this time somewhat less to our surprise) that some of these 500 projects could not be solved to optimality, even though they were the same size as the *easy* J30 instances. The reason, of course, was that we generated the new projects in such a way that they were much more diverse than the J30 instances. After all, we had set the network and resource indicators to many different values to cover a wider spectrum of all possible projects with 30 activities than the J30 instances. Size matters less than we initially thought. *It is diversity that matters!*

This experiment illustrates once again that the generation of artificial project data must be done with care and that a researcher must realise that the controlled design of artificial data can be both a strength and a weakness. Setting the network and resource indicator to specific values to generate data can be a dangerous task as algorithms evolve in such a way that difficult instances can suddenly become easy, leading to erroneous conclusions. I think this is what happened when the J30

[9] This composite lower bound procedure is the procedure that was used in the machine learning study of Sect. 6.1 where 48 different configurations were discussed.

instances were generated in the 90s. With our knowledge of network and resource indicators and the current state-of-the-art algorithms, a set of project instances was generated that, as mentioned before, completely changed the research in project scheduling. However, when it suddenly appeared that the J30 instances could be optimally solved, this quickly led to the false belief that *30-activity projects can easily be optimally scheduled with the current state-of-the-art procedures*. The new dataset of 500 projects proved otherwise and clearly indicated that even such small projects cannot be solved to optimality, even with the super-fast computers that we have at our disposal today. It appears that the J30 projects are simply too restrictive (i.e., not diverse enough in the network structure or resource scarceness). Perhaps nobody knew anymore how these projects were generated back then, and everyone just started testing the algorithms on larger projects (J60 and more) thinking that the project size is the only criterion for deciding whether a project is difficult or easy to solve. Our disappointment that we could not solve all J60, J90, and J120 project instances became a challenge to extend the J30 instances to more diverse and more difficult projects. If we wanted to achieve a better marriage between data and algorithms, we needed to look for the right project data and not blindly use larger projects to test new algorithms. As Michael Schrage, author and research fellow at the MIT Sloan School's Center for Digital Business, explained, using data and algorithms correctly remains a challenge:

> Instead of asking, "How can we get far more value from far more data?" successful big data overseers seek to answer, "What value matters most, and what marriage of data and algorithms gets us there?"

Still with endless enthusiasm and an unsatisfied hunger for better insights, José and I went looking for *diverse* projects (Sect. 11.6) and *difficult* projects (Sect. 11.7) without increasing the size of these projects above 30 activities. And so I had to stay a little longer in Lisbon.

11.6 Diverse Data

Our first search for diverse project data was a fairly easy process and resulted in a new dataset (NetRes) and a new tool (*SolutionsUpdate*) to share solutions more easily with researchers. Although this research was published in the journal *Computers and Industrial Engineering* (Vanhoucke & Coelho, 2018), I have to admit that it has not become the paper of which I am most proud of. Not that I am dissatisfied with the result, but rather I see this paper as a preparation for the next study (from Chap. 11.7) that both José and myself are extremely proud of. The results of this preparatory study are briefly discussed in the following paragraphs.

A New Diverse Dataset (*Network and Resource Diversity*) Since both network topology and resource scarcity are known to impact the solution quality of algorithms for solving the RCPSP, researchers should have access to project data with very different values for these two parameters. We therefore decided to create two

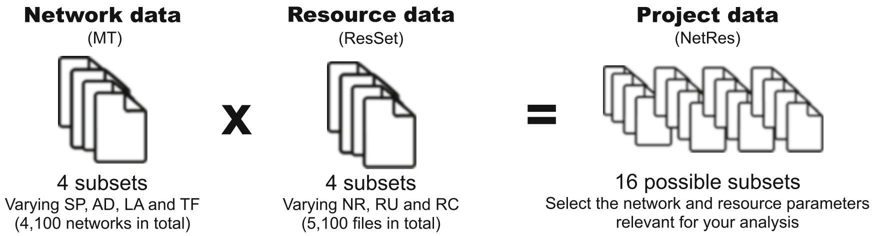


Fig. 11.4 The construction of the NetRes dataset

Table 11.1 Four network topology indicators

Network metric	Minimum (=0)	Maximum (=1)
Serial/parallel (SP)	Activity network has a completely parallel structure	Activity network has a completely serial structure
Activity distribution (AD)	All network levels have an equal number of activities	One network level has maximum activities, while other levels have one activity
Length of arcs (LA)	There are many levels between activities with precedence relations	There is only one level between immediate successors of all activities
Topological float (TF)	Without increasing the SP level, none of the activities can be shifted	Without increasing the SP level, all of the activities can be shifted

separate databases, one containing 4100 networks with a very diverse network topology (clustered into 4 subsets), and another containing only resource data with 5100 different files (also clustered into 4 subsets). These two subsets can then be combined into a new project database which we have named NetRes, an abbreviation to refer to the combination of *net*works with *res*ources. This new dataset could potentially result in +20 million projects (using all possible combinations of these two sets) as shown in Fig. 11.4, but it is up to the researcher to select the appropriate combinations that are relevant to the research study. The two separate databases are generated in the following way:

- Network topology: The network structure is controlled using five of the six I_x network parameters discussed earlier at various places in this book. The first indicator (I_1) measures the number of activities of the project that is set to 30 ($I_1 = 30$), while the values for 4 of the 5 remaining indicators I_2, I_3, I_4, and I_6 (we did not use the I_5 indicator) were set at different values. As I mentioned earlier, we no longer referred to these network topology indicators as I_x as we did in Vanhoucke et al. (2008), but gave them an easier and more understandable name. More specifically, the network structure is controlled using the *serial/parallel* (SP) indicator (I_2), the *activity distribution* (AD) indicator (I_3), the *topological float* (TF) indicator (I_4), and the *length of arcs* (LA) indicator (I_6). A summary of these network topology metrics is given in Table 11.1. The generated networks were not generated specifically for this new study but were generated for the research of my book *Measuring Time*, which is why this network set is called the MT set. The projects do not contain any resource data, and the set consists of 4100 projects with a high degree of network topology diversity (details are given in Appendix E).

- Resource scarcity: The resource data were generated in separate resource files and can only be used in conjunction with the project networks of the MT set. The resource availability for each resource and the resource requirements of the activities were generated using the two well-known resource indicators *resource constrainedness* (RC) and *resource use* (RU). Since each resource file contains resource data for exactly 30 activities, they can be perfectly combined with the MT network instances, and the *number of resources* ranges from a minimum of 2 to a maximum of 10. The resource database is divided into 4 subsets, each with a variable number of data instances, further referred to as the ResSet dataset.

Of the 16 possible NetRes combinations of Fig. 11.4, we ultimately proposed 7 subsets that can be used by researchers. These sets contain a total of 3810 projects, which is only a fraction of the 4100 × 5100 possible projects with resources. Researchers can quite easily put together other combinations, and we have also given advice on how to properly select new subsets that can be used for different research purposes. The ultimate choice rests with the researcher, but the NetRes dataset certainly offers enough flexibility to have access to the desired combinations of network and resource indicators to compile a new diverse dataset. Further details are beyond the scope of this book, and I refer the readers to a brief summary for both sets in Appendix F.

A New Data Tool (*Analysing and Sharing Solutions*) Now that we were in full swing generating new projects, we thought it would be a good idea to simplify sharing this project data with researchers. We therefore proposed a new tool to easily download this project data, but also to upload obtained solutions for these projects so that researchers can easily obtain the current state-of-the-art results. The new tool called *SolutionsUpdate* has been developed to stimulate data and results sharing. The tool is intended to facilitate the testing and comparison of algorithms for the project scheduling problem with limited resources, and all project data that will be discussed in this chapter are available in this tool. The tool should also make it easier for researchers to download current solutions and upload new improved solutions and is accompanied by a website http://solutionsupdate.ugent.be. The specific details of how to use the tool, including command line syntax and formats for downloading and uploading the various files, are, of course, beyond the scope of this book, and the readers are referred to the published version of our article (Vanhoucke & Coelho, 2018). For those who prefer not to learn a new tool, we have also put the data available on our data website at www.projectmanagement.ugent.be/research/data.

11.7 Core Data

The advantage of a large set with many diverse project instances such as the NetRes dataset is that it can be easily used to train advanced algorithms using the machine learning techniques from Chap. 6. After all, these algorithms require a fairly broad spectrum of different projects to optimally execute the training process, and it is not always very important whether these projects are difficult to solve or not. José and I quickly got to work with the NetRes dataset to verify whether this diverse set actually also contained *difficult* project instances, which was ultimately our main interest. Obviously, we could not test all instances (more than 20 million!), but after several thousands of runs we quickly realised that many instances were quite easy to solve, which made much sense. After all, if you generate a diverse set of projects, chances are high that for many of those projects, an optimal schedule with resources can be constructed very easily. For example, the projects with SP values close to 1 mainly contain serial activities, and therefore, the search for an optimal schedule for the RCPSP is quite simple because the activities just have to be scheduled one after the other. So we had to run a more controlled experiment to find difficult project instances, which led us to the research project that we are both most proud of.

I wrote earlier that the J30 instances were all very easy to solve, but that we also generated some other projects with the same amount of activities that could not be solved to optimality at all. We did not really understand which instances of the same size were easy or hard, so we had to find a better way for generating these difficult instances. It would, of course, have been very easy to simply conclude to generate projects with many parallel activities, as the construction of a schedule for these projects likely leads to many conflicts in the use of resources (because they cannot all be planned together with the limited availability of resources). However, this would lead to a new set of difficult projects with very little diversity (consisting of only very parallel projects). So the search for hard project instances promised not to be easy, something José and I love very much.

In our search for difficult projects, a trade-off had to be made between *project size* and *dataset diversity*, as researchers rely on different types of algorithms to construct resource-feasible project schedules. The use of *exact algorithms* such as branch-and-bound procedures is primarily aimed at optimally solving instances, and it is unlikely that these (rather advanced) algorithms will ever be implemented in software tools to schedule real projects. Therefore, researchers do not use such algorithms to compete with the state-of-the-art algorithms for solving real-life instances, but rather to understand the search space exploration and find ways to improve the search direction to (unknown) optimal solutions. The focus of our research was mainly on these kinds of algorithms (using our *composite lower bound branch-and-bound procedure*), so we decided to search for hard projects that contain as few activities as possible, maximum 30 (like the J30 set) but preferably even less. This is not to say that larger projects, such as the J60, J90, and J120 instances, are no longer interesting for academic research. The meta-heuristic algorithms (such as genetic algorithms or scatter search procedures) are specifically aimed at finding near-optimal solutions in a relatively fast way without guaranteeing optimality, and therefore, larger projects are needed to test their performance. Because these algorithms often provide little insight into why a project instance is easy or difficult to solve, we did not focus on that type of procedures in our study.[10]

If you think deeply about our idea of looking for difficult projects (as we have done many before), then it is actually a surprise that 20 years after their development, most branch-and-bound procedures are still unable to optimally solve project scheduling problems, despite much faster computers. It may mean that researchers still have no good understanding of which projects are difficult and thus develop the wrong algorithms, or it may simply be a logical consequence of the fact that most researchers have switched to meta-heuristic methods, and have stopped trying to solve the scheduling problem to optimality. That is why we thought that a new set of small but very difficult projects could improve the academic research by stimulating researchers to propose a completely new approach for solving these projects, potentially leading to new and better scheduling algorithms and new ideas for further research.

The search for hard project instances is summarised in a new study that we called *"going to the core of hard instances"*, published in Coelho and Vanhoucke (2020). The paper presents not only the new dataset, which we eventually called the CV set,[11] but also the process to generate this dataset. This process consisted of a very large experiment that we performed on the supercomputing infrastructure of Ghent University that took just under 40 years of computing time if it had to use a single core computer with a 2 GHz processor. The experiment started with 13,980 projects from different existing databases that were gradually changed to smaller projects. Every time a project instance was changed, we checked whether the project was difficult to solve or not. If the change resulted in an easy project, it was removed from the database. If the project were difficult to solve, we made further changes in the hope that we could increase the difficulty. The project hardness was checked by trying to solve the project with our *composite lower bound branch-and-bound procedure* (hereafter abbreviated as CBB), but we also used a *mixed-integer programming model* (MIP) (upon request of the reviewers of our paper) to make sure our results are not biased towards branch-and-bound procedures only. This hard instance search was performed in three phases, which are graphically depicted in Fig. 11.5 and briefly summarised in the following paragraphs.

Phase 1. Reduce Project Size (*Core Procedure*) The 13,980 project instances used at the start of the procedure come from existing datasets (named PSPLIB, RG30, 1kNetRes, and DC2, cf. Chap. 15) with various projects of different sizes. The CBB procedure tries to optimally solve all these projects. Whenever an optimal solution

[10] Note that in a later study presented in Chap. 11.9, we shifted our focus to meta-heuristic algorithms and looked for project instances that are both large and difficult for such methods.

[11] I consider myself to have an endless imagination, which is very convenient as a researcher, but this time we were not so creative and just used our initials to give the dataset a name (CV = Coelho and Vanhoucke).

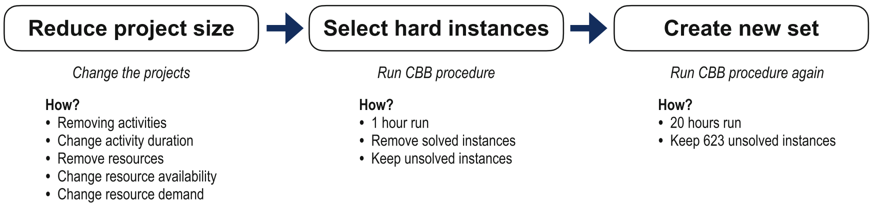


Fig. 11.5 Searching for small but hard projects (*going to the core*)

for a project can be found, the instance is said to be *easy*, and then the procedure tries to change this instance to make the project both smaller and more difficult. To do this, the procedure uses five elementary operators to gradually shrink the project instance. With each change, the newly modified project is solved, and if an optimal solution can be found, the project is again reduced by one of the 5 operators. From the moment that an optimal solution can no longer be found, the procedure decides that the project instance is *difficult* to solve, and this project is then transferred to Phase 2. However, when the increasingly smaller project can always be solved to optimality, it is not held back and the procedure continues with the next project.

The five operators consist of: (i) removing activities from the project, (ii) removing the resources from the project, (iii) changing the availability of resources, (iv) changing the duration of the activity, and finally (v) changing the demand for resources. Any change by one of those operators results in a slightly different project instance, and the changes normally result in simpler instances. However, very occasionally (for unknown reasons), the hardness of the instances increases as a result and that is what we ultimately wanted to achieve. Since there are many possible changes for each of the instances, this process could easily lead to millions of project instances to be tested for their hardness, which would simply take too much computing time.

To speed up this endless search for hard instances, the procedure does not solve every changed project instance with the CBB procedure, as this would consume all available computing time after several thousands of changes. Instead, we defined a so-called *project hardness indicator* that can be used as an estimate of the hardness of the modified project instance. Instead of experimentally verifying again and again whether a particular project instance is difficult to solve or not with the CBB procedure, this estimate is based on a number of quick and simple calculations. They calculate lower and upper bounds for the project duration to predict whether it is close to a possible optimal solution. This process of continuous changes and hardness predictions is repeated until the project is smaller than 30 activities. Therefore, it is important to realise that this process does not guarantee that the remaining projects, which are estimated to be very difficult to solve, are really difficult. The ultimate goal of Phase 1 is therefore to remove parts of projects very quickly and easily, while keeping the core of the hard parts, and each instance that meets the previous conditions is then transferred to Phase 2.

Phase 2. Select Hard Instances (*Hardness Check*) The remaining batch of modified project instances that have passed Phase 1 are believed to be hard to solve, but of course that is no guarantee. Indeed, the *project hardness indicator* of Phase 1 did not really solve the projects with the CBB procedure, but only acted as a hardness prediction. To verify whether the remaining instances are indeed difficult to solve, they must be solved to see if the obtained solution is optimal. We used the CBB procedure under different settings and performed it for 1 hour for each project instance. If, after such an intensive search, the instance could still not be optimally solved, the instance is transferred to Phase 3. Otherwise, the instance is considered too easy and discarded.

Phase 3. Create New Set (*Heavy Runs*) After the previous two phases, just over 1000 hard instances had been maintained in the database. In the third phase, we used the CBB algorithm again, but now we ran it under different settings for a long time (20 hours per setting), and if no optimality could be proven, the instance would be very difficult to solve. In addition, the remaining instances were also solved with the MIP formulations, again to check whether they could be solved to optimality. Ultimately, after many experiments, this search yielded a new set of 623 instances that could not be optimally solved with our set of algorithms. As mentioned before, these instances are in the so-called CV set, and it consists of all small (maximum 30 activities) but hard (no optimal solution found yet) projects for which new algorithms have to be developed to optimally solve them. The challenge now lies with the researchers who have the time and courage to develop such algorithms.

As I mentioned earlier, the article describing this experiment was published in the journal *Computers and Operations Research*, and both José and I consider it one of our best articles that we ever wrote together. Maybe it is the good time that we spent together in the basement of the university in Lisbon (staring at experiments) or maybe the positive comments that we got from the reviewers. Usually, such a review process consists of a long series of comments on why the article is not good enough, but this time the reviewers not only wrote that the paper was *well-written* and *well-structured*, but also that they *really enjoyed this study*. Wow! After only 1 round with some significant changes, the editor of the magazine (Francisco Saldanha da Gama, also from Lisbon) decided to accept the paper and publish it in the 2020 issue of this flagship journal. Afterwards I got to know Francisco better, and we have already agreed a few times to drink a few fresh cold beers (Super Bock preferably) in sunny Lisbon. For some reason, I always get on well with people from Portugal, and I think my ancestors must have been born there. I want to thank Francisco for how well he manages this excellent journal and how he manages to provide authors with very fair and astonishingly fast review processes. I am sure that we will meet again on a regular basis, now that I have bought my own place in this city, and I hope that we will meet many more times in joint jury meetings for PhD defences (as we did for the first time at the PhD defence of Jakob Snauwaert in 2022 in ice-cold Belgium). I also hope that the CV set will be used by many researchers to generate better, hopefully optimal solutions. It may lead to a completely new way of building algorithms that solve project scheduling problems completely differently than before. Time will tell whether this will indeed be the case.

11.8 Equivalent Data

Now that we had generated a new set of hard instances, we thought to stop looking for new insights into artificial project data. We thought we had done just about everything we could and were now happy to leave the follow-up studies to other researchers. But the Portuguese sun and the inspiration we gained from our meetings decided otherwise. Somehow we felt like our story was not quite finished. After all, we kept thinking about how other researchers would manage to solve these difficult CV projects. We were very aware that such a quest to better scheduling algorithms would not be an easy task and that a better understanding of why some projects are difficult to solve, and others very easy, could make that quest somewhat easier. These insights would enable many researchers to devise better algorithms by tailoring the search of these new algorithms to specific characteristics of the projects. We therefore intuitively knew that the key to finding these project-specific characteristics, measured by the network and resource indicators, lies in the previously discussed concept of *phase transitions*. More specifically, a better understanding of the specific values of the network and the resource indicators where a phase transition occurs (from easy to hard, or vice versa) should allow researchers to direct the search of their algorithms to complex zones of the project, hopefully leading to better solutions that would otherwise never be found.

The importance of correct knowledge about the phase transitions for the RCPSP cannot be overstated. Many studies test algorithms on one of the benchmark datasets discussed in this chapter and often publish results as average values (for all projects in the dataset) without splitting them for different values of the network and resource indicators. Such studies thus provide few insights into the complexity of the problem and often create a limited understanding of the value ranges for the network and resource indicators that make the problem difficult. That is why we thought that too few studies look very specifically at ways to improve the project scheduling algorithms for these specific hard projects, and we felt it necessary to investigate it more deeply. In a new follow-up study (Vanhoucke & Coelho, 2021), we aimed at improving the knowledge and insights about the complexity patterns of these resource indicators for the RCPSP by analysing the resource characteristics of projects in depth. We came up with a new concept, called *instance equivalence*, and arrived at some surprising results, which I will briefly describe in this section. In fact, a few years earlier, I had already attempted to integrate the resource indicators into a project scheduling algorithm so that it can take into account the phase transitions during its search to a resource-feasible schedule. I developed a genetic algorithm in collaboration with a PhD student Vincent Van Peteghem[12] for the *multiple-mode* extension of the RCPSP, and the title of this study clearly reflects our

[12] Vincent investigated the multi-mode resource-constrained project scheduling problem as a PhD student between 2005 and 2011. Nine years later, he became the Minister of Finance in the Belgian government. Not only am I proud of how his hard work has enabled him to realise his political ambitions, but it also feels good to have a project planner at the head of our country.

ambition of this study as *"Using resource scarcity characteristics to solve the multi-mode RCPSP"* (Van Peteghem & Vanhoucke, 2011). Indeed, this study explicitly took into account the presence of phase transitions by implementing different versions of local search algorithms. The genetic algorithm selects the specific local search procedure based on the specific values of the resource indicators of the project instance to be solved. The research shows that including such resource-specific information leads to better solutions than a generic approach, illustrating the importance of a good understanding about phase transitions for project scheduling algorithms.

It goes without saying that the success of incorporating information about phase transitions into algorithms is highly dependent on the ability of the network and resource indicators to detect phase transitions, and there is not always a clear agreement in the literature as to whether such phase transitions actually can be predicted. The impact of the *network structure* on the complexity of the scheduling problem is well-known, but the ability of *resource indicators* to predict the complexity of the problem has been questioned in the literature for decades. For example, De Reyck and Herroelen (1996) have experimentally shown that the *resource strength* (RS) cannot distinguish between easy and difficult problem instances, while the *resource constrainedness* (RC) does, and therefore they implicitly claim that the RC has a clearer predictive power of the problem complexity than the RS. However, the question then remains what the correct values are for the RC indicator to see a transition from easy to difficult. Thus, in order to fully exploit the presence of phase transitions for developing new scheduling algorithms, the values of the resource indicators must be unambiguously defined. This makes it easier to predict whether a project instance falls in the right complexity class (*easy* or *hard*), and the scheduling procedure can take this into account for generating better solutions.

The contribution of the follow-up study with José is that we have shown that the values for the resource indicators are not always clearly defined, sometimes leading to inaccurate or unreliable results about the presence of phase transitions. Indeed, we have found that you can easily find two project instances with different values for the resource indicators where all possible solutions (i.e., all possible resource-feasible schedules) for these two instances are exactly the same (and thus the projects have a similar complexity). In such a case, the values for the resource indicators should not differ, as one instance can be transformed into another without changing the number of possible schedules. Therefore, their complexity (to find the optimal schedule) is identical. This observation has been the starting point of a search for the *true values* of resource indicators that best describe the presence of phase transitions. More specifically, we developed an algorithm to transform each project instance into a different instance by iteratively changing the values for the resource parameters without changing the set of possible solutions. We show that the complexity of these two *equivalent instances* is exactly the same, even though they have different values for the resource indicators. When the transformations of projects involve few changes in the resource indicator value, this indicator is more reliable than when a transformation involves a large change in the value of a resource indicator. The transformation algorithm for changing the resource indicators of a project instance

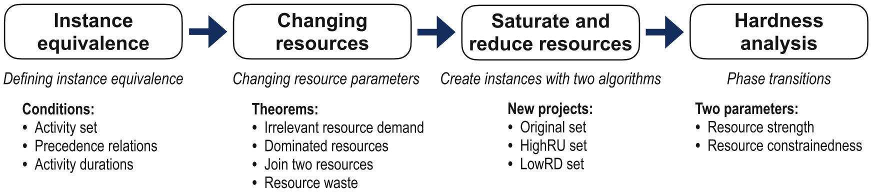


Fig. 11.6 Searching for equivalent projects

without changing the set of possible schedules is graphically illustrated in Fig. 11.6 and consists of four phases discussed along the following paragraphs.

Phase 1. Instance Equivalence In a first phase, the necessary conditions for the new *instance equivalence* concept are introduced. A project instance I^1 is equivalent to another project instance I^2 if and only if the set of all possible solutions (resource-feasible schedules) for both projects are the same. This new concept makes use of the set of *all* resource-feasible schedules for instance I, represented by *Sol(I)*, and is formally defined as follows:

Instance I^1 is equivalent to instance I^2 if and only if $Sol(I^1) = Sol(I^2)$

In order to define I^1 and I^2 as two equivalent instances, they should satisfy the following three conditions:

- *Activity set*: The set of activities for both instances should be the same.
- *Precedence relations*: The set of precedence relations between the two instances should be the same. Logically, if a precedence relation between activity i and j in I^1 does not exist in I^2, then activity j can be scheduled before activity i for I^2, which is obviously not a precedence feasible schedule for I^1.
- *Activity durations*: The duration of each activity must be the same.

These necessary conditions imply that equivalent instances should have the same network structure, but *not* the same resource demand and availability. Hence, we focused in the next phase on changing the resource indicators (resource demand and resource availability) of the project instance, while keeping the network structure unchanged, hoping to find equivalent project instances.

Phase 2. Changing Resources In this second phase, we developed a new algorithm to search for equivalent instances by changing the resource parameters of projects in a systematic and iterative way. The algorithm starts with any project instance and transforms it into two new equivalent instances. The algorithm relies on four new theorems that each change the resource demand and/or availability of the project without changing the set of possible solutions. More specifically, the theorems aim at reducing the resource demand (Theorems 1 to 3) or increasing the resource demand (Theorem 4), while keeping the set of all possible solutions unchanged (i.e., guaranteeing instance equivalence):

- *Irrelevant resource demand*: The first theorem states that, under certain conditions, the resource demand for some activities has no influence on the project schedule. If that is the case, this demand is set to zero without changing the set of all feasible schedules.
- *Dominated resource*: The second theorem removes a resource from the project if it is dominated by other resources, which leaves the set of all feasible schedules unchanged.
- *Join resources*: The third theorem connects two resources of the project and merges their data into a single resource. It searches for activities that use only one of the two resources and for which all precedence-compatible activities also use only the same resource. If such an activity can be found, the resources can be merged and the set of feasible schedules remains the same.
- *Resource waste*: The fourth theorem is aimed at increasing the resource demand of a resource based on calculating the maximum possible utilisation of resources of a particular activity i and all other activities that can be scheduled in parallel with activity i. The difference between this maximum and the resource availability is called *resource waste*, which expresses the unused resource capacity and leads to the possibility of increasing the resource demand without changing the set of feasible solutions.

Phase 3. Saturate and Reduce Resources The four theorems were implemented in two transformation procedures to create two additional instances of each project instance. The first procedure *saturates the resource demand* by increasing the resource demand for each project activity as much as possible. The second procedure *reduces resource usage* by minimising the number of resources used by the project activities. Consequently, the search algorithm transforms each project instance into two new equivalent project instances with more compact resources, and thus very different values for the known resource indicators. These three project instances are called the *base* project, the *highRD* project, and the *lowRU* project. In the study, we used 10,793 *base* project instances from five known datasets, resulting in a total dataset of $3 \times 10{,}793 = 32{,}379$ instances. Each triplet of instances has an identical network structure, different values for the resource indicators, and the same set of feasible solutions (and thus the same optimal solution).

Phase 4. Hardness Analysis To find out which resource indicator is the most reliable indicator to predict the hardness of the project, we performed a huge computational experiment using the *composite lower bound branch-and-bound procedure* of Sect. 11.5 to classify all 32,379 instances as *easy* (easy to solve) or *hard* instances (impossible to solve within a predetermined time). Next, we checked whether the hard instances were clustered around certain values of the resource indicators, both for the new (*highRD* and *lowRU*) and the original (*base*) instances. If the clusters of hard instances are around specific values for a particular resource indicator, the resource indicator is said to be reliable. Indeed, when a significantly higher percentage of hard instances is clustered around certain values for a particular resource indicator and other indicator values contain almost no hard instances, then the indicator may be able to better distinguish between easy and hard instances. Otherwise, if the cluster of hard instances has values for the resource indicator evenly distributed between the lowest and highest values, it means that this particular indicator cannot clearly distinguish between easy and hard instances. We learned from our experiments that the *resource strength* (RS), and not the *resource constrainedness* (RC), shows the highest discriminative power for the existing datasets. Despite the aforementioned criticism raised for the RS, its relatively high reliability did not come as a surprise. The RS is the only resource indicator that integrates both the network structure *and* the resource characteristics into one single number, and it is well-known in the literature that both must be used to predict the hardness of an RCPSP instance. In the study of 1996 by Bert De Reyck[13] and Willy Herroelen (De Reyck & Herroelen, 1996), the authors concluded that both the network and the resource indicators should be used to predict the problem hardness of the RCPSP, as they wrote:

> It seems evident that the structure of the network, in whichever way it is measured, will not be sufficient to reflect the difficulty encountered in the resolution of such problems.

I believe that this study has shown the importance of resource indicators in predicting the hardness of a project instance with limited resources and that these indicators can be biased due to the fact that different values do not always lead to different instances (*instance equivalence*). I hope and believe that this will encourage researchers to pay more attention to the existence of phase transitions to look for better resource indicators that are not subject to large changes in their values between equivalent instances. In fact, we should aim to find *true* resource indicator values that perfectly predict the difficulty of the problem and do not change the values between equivalent instances, but that does not seem like an easy task. We have already made a number of attempts at this in a number of follow-up studies, but we are certainly not yet at the point where we can perfectly predict the difficulty of the RCPSP. For example, we presented a theoretical framework in collaboration with Rob Van Eynde, a PhD student at the OR&S group, to better understand the complexity of the resource-constrained project planning problem (Van Eynde & Vanhoucke, 2022), and experimentally tested the newly proposed resource indicators in a second study in Van Eynde et al. (2023). It is my hope that through such studies the presence of phase transitions will be much better understood, hopefully leading to new and completely different algorithms for solving the challenging problem of project planning with limited resources. Time will, as always, tell.

[13] It is a small world. Bert De Reyck was a PhD researcher under the supervision of Willy Herroelen (my co-advisor) a few years before I started, and later became the director of the UCL School of Management in London where I currently work.

11.9 From a Distance

The attentive readers may have (*correctly*) noticed that our previous studies on artificial data generation have mainly focused on better understanding the complexity of the RCPSP by using a branch-and-bound procedure to generate optimal solutions. This search for insights into the complexity of the problem is, from an academic point of view, very interesting, but of course does not ensure that large projects can be scheduled very quickly. For that, we cannot use the exact branch-and-bound procedures (as they are too slow for large projects), and we have to switch to other scheduling algorithms such as priority-rule-based scheduling techniques or meta-heuristic methods (as briefly described in Sect. 3.4). However, the data instances of the CV set that we generated earlier were quite small (maximum 30 activities) and thus far too easy for these fast and efficient heuristic solution methods. Moreover, since the projects in the CV set were made to be difficult to be solved by *exact* methods, there was no indication that the same criteria would apply to describe the difficulty when a different kind of scheduling algorithms was used. And so our story of the quest for difficult projects began again, this time with *meta-heuristic algorithms* to solve the RCPSP in mind. Sometimes as a researcher you have to question yourself and look at your past research results *from a distance*. And this new perspective also brought new insights.

I have mentioned before that the RCPSP is notoriously complex, and because of this known complexity, many researchers have abandoned the path of exact algorithms (which guarantee an optimal solution) and instead propose fast and efficient meta-heuristic solution procedures. These methods cannot guarantee optimal solutions, but they nevertheless provide very good, sometimes near-optimal solutions. In recent decades, the research on such meta-heuristic algorithms has exploded, and the review article by Pellerin et al. (2020) provides an impressive overview of the performance of most of the meta-heuristic procedures available in the literature. Despite this impressive summary, what is most striking in this study are the marginal improvements made in recent years (often less than fractions of percentages). This may indicate: (i) that most project instances have been solved almost to optimally or (ii) that most meta-heuristics all rely on the same or similar building blocks and no more progress towards better solutions is made. And so we realised that the search for project complexity depends on the type of scheduling algorithms. In fact, our previous studies focused on *"searching for small but hard project instances"* (Sect. 11.7) and *"searching for reliable resource indicators"* (Sect. 11.8). However, since we had always used exact algorithms to solve the RCPSP, we never thought whether our research results would still be valid for other solution methods. We immediately started working on a new idea to generate projects that would be *large* and *difficult* for fast and efficient meta-heuristic algorithms. Time for another visit to Lisbon[14] to start a series of experiments carried

[14] It was at this moment that I started to realise that splitting my research time between Ghent (Belgium) and Lisbon (Portugal) not only gives me a lot of efficiency benefits, but also makes me

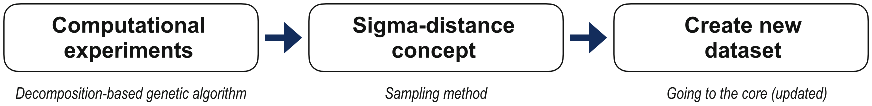


Fig. 11.7 Searching for hard and large projects

out in three separate phases as shown in Fig. 11.7 and discussed in the following paragraphs.

Phase 1. Computational Experiments To conduct our research, we needed to have access to a fast and high-performing meta-heuristic procedure that could solve the RCPSP to near-optimality and that could compete with the best performing algorithms in the literature. Luckily, I had the code of the *decomposition-based genetic algorithm* developed by Debels and Vanhoucke (2007). Dieter was my first PhD student ever[15], and his algorithm still ranks highly in the study of Pellerin et al. (2020) after all these years. Namely, this study shows that the algorithm can perfectly compete with the state-of-the-art algorithms in the literature, and it is even considered the fastest algorithm to generate a predefined number of solutions. We therefore solved the 10,793 previously discussed project instances using this procedure, and we aborted the search process only after 500,000 generated schedules to ensure the algorithm could not find better solutions. In addition, we also used the exact *composite lower bound branch-and-bound algorithm* of Sect. 11.5 under very high stopping criteria (up to 20 hours per project instance), which of course does not mean that all instances could be optimally solved. However, the solutions of both algorithms were compared to each other so that the following conclusions could be drawn:

Observation 1.

"Meta-heuristics cannot find optimal solutions for hard instances"

During the very intensive 20-hours search, the exact algorithm was sometimes able to find better solutions that were not found by the meta-heuristic, even if the latter procedure was used very intensively (up to 500,000 schedules). Of course, it comes as no surprise that meta-heuristics do not always provide optimal solutions even under high stopping criteria, but as hard as we tried, we could not find any value for a network or resource indicator that could explain why some projects were optimally solved by the meta-heuristic procedure, while for other projects this was not the case. This led us to the conclusion that the current network and resource indicators did not provide sufficient insight into the complexity of the project and therefore cannot be used to predict the hardness of a problem instance.

very happy. My wife and I therefore decided to finally make our dream come true as we bought an apartment in Parque das Nações in Lisbon.

[15] Your first PhD student is like your first love. Every researcher has a special place in my heart, but the first one touches you in a way that is inherently unique and determines your future life (career).

Consequently, further analysis of instance complexity for meta-heuristic searches was needed to solve the RCPSP, which became the subject of our new study.

Observation 2.

"Meta-heuristics cannot find optimal solutions for easy instances"

In a second experiment, we tried to classify each of the instances in an *easy* or *hard* cluster for both algorithms. Surprisingly, 2008 of the 10,793 instances were not classified in the same cluster for both algorithms (what was difficult for one algorithm was easy for the other), resulting in a misclassification error between the two algorithms of almost 19%. This *misclassification error* further illustrated that defining and predicting the hardness of a problem instance depend on the procedure used and that more research was needed to better describe the problem hardness when using meta-heuristic search procedures. A further analysis of this experiment also showed that even the relatively easy projects could not always be optimally solved by the meta-heuristic search procedure. While we saw that the meta-heuristic could find optimal solutions for most of these easy instances, there were still some instances that were classified as *easy* by the exact algorithm (meaning an optimal solution could be found relatively quickly), but this solution could not be found by the meta-heuristic algorithm even after 500,000 schedules. Again, a closer look at the project instances did not tell us much and this once again illustrated that predicting the hardness of a problem instance for a meta-heuristic is no easy task. We therefore introduced a new concept to better understand the difficulty of projects for meta-heuristic solution methods, which is discussed in Phase 2 of this research.

Phase 2. Sigma-Distance Concept To better understand the performance of meta-heuristic search algorithms, we decided to introduce a way for describing the *solution space* of a project instance. A project instance's solution space consists of all possible resource-feasible schedules that can be generated. Of course, such a space can grow quickly, even for medium-sized project instances, so two different procedures were used to generate the solution space: First, a *full enumeration* method was used that simply lists all possible schedules and reports the frequency of schedules for each makespan value found. Such a method can, of course, only be used for relatively small projects where the solution space is not very large. When the project instances get bigger, a *sampling method* (SM) was used that randomly generates a series of activity lists, then constructs a schedule using a schedule generation scheme, and stops after generating a pre-specified number of activity lists. Each schedule is constructed by one of four possible schedule generation methods to convert the generated activity list into a resource-feasible schedule. More specifically, both the *serial schedule generation method* and the *parallel schedule generation method* were used, and they were implemented as both a *forward* or *backward* planning method (resulting in four different ways to generate a project plan for a randomly generated activity list). When a huge number of activity lists are generated, the solution space can be described as best as possible, but not completely. We decided to use the sampling method (and not the enumeration method) to explore the solution space of the 10,793 project instances. The average

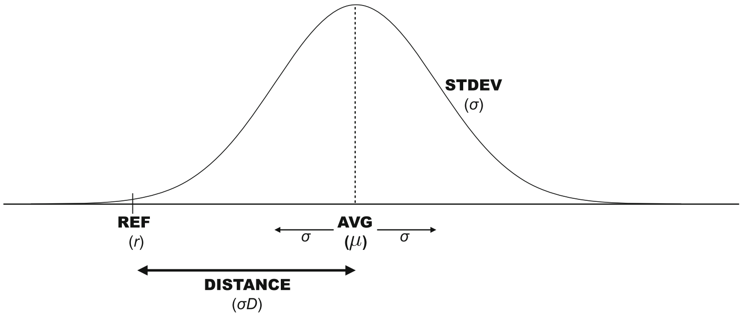


Fig. 11.8 The sigma-distance concept (*measuring the hardness of a project*)

makespan of all generated solutions is then compared to a very good solution for each project instance, and the closer the sampled solutions are to the good reference solution, the easier it seems to find this optimal solution for the project instance. This distance—which we called the sigma distance σD—serves as a new hardness indicator and expresses the ease/hardness to find a very good, preferably optimal solution for the instance. The core idea of this method is shown in Fig. 11.8. The figure shows the frequency of solutions found by the sampling method for each possible makespan (x-axis), represented as a normal distribution with a known mean and standard deviation. This distribution is compared to a known reference point, and the distance between the two expresses the hardness of the instance. The main components for obtaining such a distribution to measure the hardness indicator are explained along the following lines:

- *Sampling method* (u_0, σ): Using the sampling method to explore the search space generates a series of unique distributions for each of the four schedule generation methods. Each distribution is assumed to be symmetrically formed with a known average makespan u_0 and a standard deviation σ. Since the distribution must represent the collection of all possible resource-feasible schedules, it is important that the sampling method is as *"pure"* as possible to ensure that it generates the entire solution space as complete as possible. This means that sophisticated elements normally included in an efficient solution procedure to guide the search for the promising regions of the solution space must be excluded in order to describe the solution space as completely random as possible. Therefore, we believe that our choice to use randomly generated activity lists with four schedule generation schemes is the purest way to generate solutions because they are very simple and do not contain advanced elements that favour good solutions over weak solutions.
- *Reference value* (r): While the sampling method's search space exploration reveals information about the *size* and *spread* of all feasible schedules, it says little to nothing about the *quality* of the generated solutions. For the latter, a reference point (REF) must be chosen, which consists of a very good or

preferably an optimal solution for the project instance, although the latter is not always known for each instance. The closer the sampled solutions are to the known reference point, the easier it seems that the optimal solution can be generated by a solution procedure that heuristically solves the instance. However, the choice of a reference point is not always easy. Ideally, we should use the optimal solution found by an exact algorithm, but this solution is not always known. Instead, we used as much information as possible to find a good reference point, including the best solution found after a long search, but also lower bound information for each project instance to estimate what could be a very good reference point.

- *Sigma-distance* (σD): Based on the average and standard deviation of the solution space generated by the sampling method, and the chosen reference point as discussed earlier, a new hardness indicator (σD) is proposed that measures the distance between the reference point and the average makespan (u_0). This distance is expressed as the number of standard deviations and is measured as shown in Eq. (11.1), with higher σD values indicating harder project instances.

$$\sigma D = \frac{\mu - r}{\sigma}. \tag{11.1}$$

Phase 3. Create New Dataset The new σD concept was used in a third phase to generate a new dataset with projects with low (easy) to high (difficult) σD values. More specifically, we decided to generate a new dataset using a modified version of the *"going to the core"* algorithm from Sect. 11.7. The original procedure focused on creating the smallest possible instances that cannot be optimally solved with the currently available *exact* scheduling algorithms by transforming project instances into new, more difficult instances. Although the same logic was applied in the modified version, it no longer focused on searching for small instances, but instances with different values for the σD metric. After a huge computational experiment, we were finally able to present the newly constructed dataset consisting of 390 new hard project instances with σD values ranging from 3 (easy projects) to 16 (difficult projects). Several computational experiments were performed to show that the new dataset—referred to as the sD dataset—contains instances that are more difficult than the existing instances available in the literature (for meta-heuristic algorithms). The value of σD was also shown to be positively correlated with instance complexity, which was the starting point of our study. Therefore, in the study, we also recommended researchers to test their new meta-heuristics on these new sD sets using different runs for each instance to report not only the quality of the solutions, but also the stability of the found solutions. We hope and believe, as for the research of the previous sections, that this new dataset can help improve our understanding of what makes a project instance difficult, but this time for constructing resource-feasible schedules by a meta-heuristic algorithm. The study has been published in *Computers and Operations Research* (Coelho & Vanhoucke, 2023). You might have noticed, but many of our papers have been published in this excellent journal. I guess it has become our favourite outlet for our research.

11.10 Final Words

You have reached the end of this long chapter. I have spent quite a few pages of this book on the research story on artificial project data, and I have not even finished yet. Chances are that José and I have continued to work on creating new artificial project data to improve our understanding of the complexity of the challenging *resource-constrained project planning problem* (RCPSP) with hopefully promising new results that could not be presented at the time of publishing this book. In this chapter, I have mainly discussed project data for the RCPSP, but Chap. 15 will present more project databases for extended versions of the RCPSP, including problem features such as human skills, the use of alternative technologies, and the presence of multiple projects in a portfolio. All project instances of all datasets discussed in this book can be downloaded from www.projectmanagement.ugent.be/research/data. The main reason why I am not quite finished with the discussion about artificial project data is mainly because in the current chapter only *static data* were presented (in the form of project networks with resources). But since this book wants to go further, and expand static *baseline scheduling* with *schedule risk analysis* and *project control*, there must, of course, also be dynamic progress data available that fictitiously describe the execution of these artificial projects. The generation of these artificial progress data is mainly done on the basis of simulations (to imitate the real project progress), and three completely different models to obtain these data will be discussed in detail in the next chapter.

References

Coelho, J., & Vanhoucke, M. (2018). An exact composite lower bound strategy for the resource-constrained project scheduling problem. *Computers and Operations Research, 93*, 135–150.

Coelho, J., & Vanhoucke, M. (2020). Going to the core of hard resource-constrained project scheduling instances. *Computers and Operations Research, 121*, 104976.

Coelho, J., & Vanhoucke, M. (2023). New resource-constrained project scheduling instances for testing (meta-)heuristic scheduling algorithms. *Computers and Operations Research, 153*, 106165.

De Reyck, B., & Herroelen, W. (1996). On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research, 91*, 347–366.

Debels, D., & Vanhoucke, M. (2007). A decomposition-based genetic algorithm for the resource-constrained project scheduling problems. *Operations Research, 55*, 457–469.

Demeulemeester, E., Dodin, B., & Herroelen, W. (1993). A random activity network generator. *Operations Research, 41*, 972–980.

Domingos, P. (2018). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. New York, NY, USA: Basic Books Inc.

Herroelen, W., & De Reyck, B. (1999). Phase transitions in project scheduling. *Journal of the Operational Research Society, 50*, 148–156.

Kolisch, R., Sprecher, A., & Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science, 41*, 1693–1703.

Pellerin, R., Perrier, N., & Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 280*(2), 395–416.

Tavares, L., Ferreira, J., & Coelho, J. (1999). The risk of delay of a project in terms of the morphology of its network. *European Journal of Operational Research, 119*, 510–537.

Van Eynde, R., & Vanhoucke, M. (2022). A theoretical framework for instance complexity of the resource-constrained project scheduling problem. *Mathematics of Operations Research, 47*, 3156–3183.

Van Eynde, R., Vanhoucke, M., & Coelho, J. (2023). On the summary measures for the resource-constrained project scheduling problem. *Working paper (Under Submission).*

Van Peteghem, V., & Vanhoucke, M. (2011). Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem. *Journal of Heuristics, 13*, 77–93.

Vanhoucke, M., & Coelho, J. (2018). A tool to test and validate algorithms for the resource-constrained project scheduling problem. *Computers and Industrial Engineering, 118*, 251–265.

Vanhoucke, M., & Coelho, J. (2021). An analysis of network and resource indicators for resource-constrained project scheduling problem instances. *Computers and Operations Research, 132*, 105260.

Vanhoucke, M., Coelho, J., & Batselier, J. (2016). An overview of project data for integrated project management and control. *Journal of Modern Project Management, 3*(2), 6–21.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research, 187*, 511–524.

Chapter 12
Progress Data

The previous chapter gave a complete overview of generating artificial project data with network generators to design datasets with varying degrees of complexity, size, and diversity. The discussion about generating *artificial project data* was very clearly focused on generating *static* project data, consisting of project networks with resources to solve the well-known *resource-constrained project planning problem*. Therefore, the research discussed in Chap. 11 focused only on the *baseline scheduling* component and said nothing about the *scheduling risk analysis* or *project control* components. This is exactly why all project network generators use network topology and resource scarcity indicators to generate the projects, rather than include generation processes to generate dynamic risk and control data. This limited focus on baseline scheduling is somewhat at odds with a book that integrates the three general components (*scheduling*, *risk*, and *control*) into one dynamic scheduling system as shown in Fig. 3.1. In fact, most chapters of Parts II and III of this book mainly dealt with these two *dynamic* components (risk and control), starting with the basic schedule as a reference point. However, data on the (fictitious) progress of projects are not included in the generation of these artificial projects and had to be generated by the so-called static and dynamic simulation runs, as discussed earlier in Chap. 5. More specifically, I have shown that the static simulations were necessary to calculate the sensitivity metrics of activities (*schedule risk analysis*) as well as dynamic simulations to simulate fictitious project execution (*project control*).

These static and dynamic simulation runs for generating progress data are not only powerful, but also quite easy to implement, so it is very tempting to use these techniques over and over again instead of collecting real progress data for real projects. Caution is advised, however, as these techniques require *probability distributions* to model uncertainty in activity durations, and the choice of these distributions (and the values for their parameters) depends on the researcher's imagination. It is thus difficult to validate whether these choices actually generate progress data that are somewhat similar to the progress of real projects, and

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_12

therefore, these artificial simulation experiments must be performed with great care. The American novelist Tom Clancy expressed it in the following way:

> The difference between fiction and reality?
> Fiction has to make sense.

This chapter proposes three completely different models for generating fictitious progress data in order to bring the fiction as close to reality as possible. When researchers follow the guidelines for the correct use of each of these three models, they can generate dynamic progress data in a structured way to improve the accuracy of dynamic risk and control studies on artificial projects. Of course, such an approach still does not guarantee that the generated progress data fully mimic reality, but they already enable the researcher to generate solid progress data that can be used for academic research. If the reality is to be fully imitated, then the progress data must, of course, come from an empirical project database, and a lot of attention will be paid to this in Chaps. 13 and 14. But before I switch to the real world, in the current chapter I want to step into the fictional world for a moment to discuss the different ways in which the artificial project progress data of the previous chapters were generated.

12.1 Imagination

I have discussed before that generating static project data amounts to generating project networks under a controlled design (by varying the network structure and the resource scarcity) to span the *full range of complexity*. This chapter focuses on the dynamic project data to imitate the progress of a project, and ideally, these data are also generated under a controlled design. Setting up a dynamic project execution study to imitate fictitious project progress therefore requires the simulation of activity delays, cost overruns as well as activities that sometimes finish earlier than expected or were cheaper than initially thought. Since these simulations require predefined probability distributions to generate variability in the project schedule, the correct selection of a specific probability distribution and its parameters is paramount.

Since this selection must be made for artificial projects, it is impossible to take into account practical real-life features, and therefore, the researcher must above all have enough *imagination* when defining these distributions and parameters. I have always seen my job as a researcher as a way to turn my infinite imagination into research questions and therefore find generating progress data as imaginative as reading a thrilling book. Mimicking reality for projects does indeed require a certain amount of imagination, and choices have to be made since there are multiple ways in which the reality of a project can unfold. The central idea is that the researcher imitates a wide range of realities using a diverse set of possible scenarios, so that every possible way the project can be executed will be within the set of generated realities. This controlled approach to data generation is therefore very similar to the concept of *spanning the full range of complexity* used to generate static project network data, but it is now applied to generating dynamic progress data. Thus, the concept must ensure that it *spans the full range of possible realities* by relying on the researcher's imagination of what the progress of a real project might look like. My favourite scientist Albert Einstein formulated the importance of fantasy much better and shorter than I did, as he said:

> Your imagination is your preview of life's coming attractions.

Of course, the use of fantasy is as dangerous as it is exciting and must always serve the ultimate research topic. Despite the fact that the researcher has enough freedom to define this fantasy in a personal way, there are a number of rules that must be followed. In order not to fantasise completely, this chapter proposes a general framework as shown in Fig. 12.1. The three models discussed in the following sections fit perfectly into this framework, providing an example of how to generate progress data for artificial project executions. The figure outlines the whole picture, starting with the artificial data generation of the previous chapter, followed by the dynamic progress data generation of this chapter and finally showing the research studies presented in the previous parts of this book. More specifically, the figure consists of five phases with Phase 2 being particularly important for this chapter. It is important for the readers to understand the logic of the figure as it is used not only to explain the generation of artificial project progress data (discussed in this chapter), but also to explain how the calibration procedures of Chap. 14 will be used (when using empirical project data). The five phases are now explained along the following lines.

Phase 1: Static Data The figure starts by generating artificial project data (networks and resources) with network generators as explained earlier in Chap. 11 to span the *full range of complexity*.

Phase 2: Dynamic Data In a second phase, each project network must be expanded with dynamic project progress data to mimic every possible reality. As mentioned, this simulated reality should not only be meaningful (as realistic as possible), but also should *span all possible realities* as good as possible. This second phase is the subject of this chapter where three different models for generating artificial project progress data will be presented. The first model is called the *variation model* and is the easiest way to generate artificial project progress data. It provides simple guidelines for selecting an appropriate probability distribution for variability in activity durations, as suggested in Sect. 12.2. The second so-called *risk model* is a bit more advanced because it models dependences between the project activities using a systematic, but elegant approach (Sect. 12.3). The last *scenario model* of Sect. 12.4 consists of nine fundamentally different building blocks that are used to imitate the duration variability of activities. These nine scenarios are specifically designed in such a way that they each have a specific goal and measure the progress

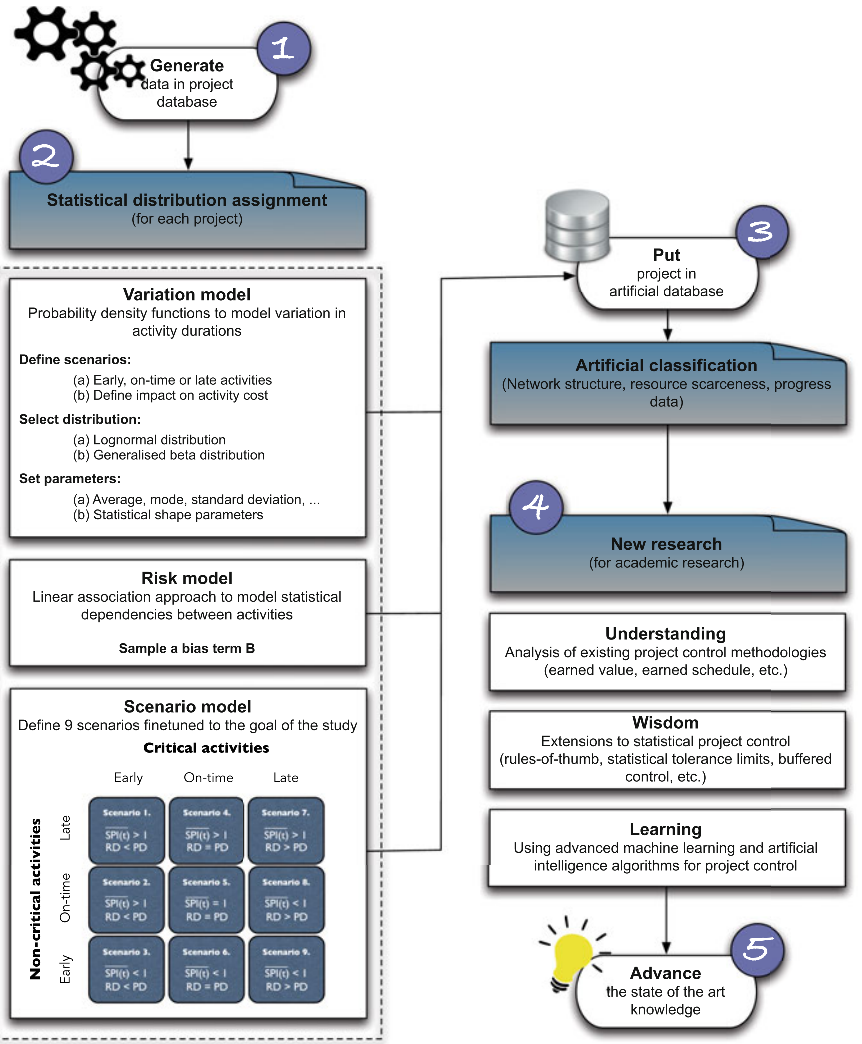


Fig. 12.1 Generating artificial project progress data (*using your imagination*)

of a project in a way that is completely different from the other scenarios. The latter model was the one that I used in the comparative study of prediction techniques as described in Chap. 4.

Phase 3: Data Classification When both the static data (Phase 1) and the dynamic data (Phase 2) are generated, they can be combined together often resulting in a huge set of data that will, almost literally, blow up your hard drive. As a researcher, it is of course nice to have so much data available, but it is also one of the weaknesses of artificial data: sometimes there is simply too much data available so that they all seem rather unstructured. In order to find good structure in this overwhelming amount of data, they must be classified into well-defined categories before they can be used for further research. The static project data are best classified by the static network and resource indicators discussed in the previous chapter (cf. Sect. 11.2). For the dynamic project data, a similar classification model can be used by classifying the progress of projects using the input values of the parameters used in the three models (*variation*, *risk*, or *scenario* model) or the output values after the simulation has been run (*early*, *on-time*, or *late* projects).

Phase 4: New Research Once the data are available, the life of a researcher is one big challenging search for results with endless possibilities and enough *freedom to explore* them from different angles. Academic research is indeed a search for improvements in project management, and the rich set of project data can be used to conduct all kinds of new experiments. They can be used to validate existing concepts (cf. Chap. 4), to make improvements (cf. Chap. 5) or even to test totally new ideas (cf., Chap. 6). Academic research is the act of exploring new territories aiming at improving our *understanding*, creating *wisdom*, and enhancing *learning*, and static and dynamic project data play a crucial role in this search.

Phase 5: Advancing Knowledge Exploring new territory and enjoying academic freedom without strict deadlines does not mean academic research is an easy job. We must not forget that the ultimate goal of academic research is to find new ways of managing projects. This quest for advancing the very latest knowledge is a never-ending quest, and a good scientist is well-aware that it is a gradual quest for better knowledge that will never be completely finished. Advanced knowledge will be replaced by newer and better results, and every step, however small or insignificant, can be a good step in the right direction. It is a warning to any young researcher considering starting a career in academia: "*Think before you start!*" Increasing knowledge has a beginning, but it never has an end, and despite the fact that it is the most beautiful job in the world, every step on an untrodden path will be a beginning of a new journey with more data, new experiments, sleepless nights, and no end.

The next three sections will discuss the three models to generate artificial project progress data. Despite the fact that they can be used as general models for various dynamic project risk and control studies, it is important to note that they are sprouted from my imagination and fantasy and are therefore not the only ways to generate these dynamic data.

12.2 Variation Model

Variability in the duration of activities is inherent in the actual project progress and can lead to delays of activities (*behind schedule*) or early delivery of activities (*ahead of schedule*). This variability in the activity durations can undoubtedly have a significant impact on the costs of these activities, and therefore examining the variability of the activity costs is just as important as the variability of their durations. In the variation model, the variability of activity duration is modelled by a predefined statistical distribution, and the variability of costs follows from the variability in activity duration. More specifically, costs are assumed to follow a *linear relationship* with duration. This means that each day of delay brings a linear increase in costs, while each day that an activity is ahead of schedule, costs also decrease linearly. While such a situation may not always reflect reality, it is a reasonable assumption that a researcher can make when validating new methods of time and cost control. This is exactly what researchers mean when they write in their study that "*without loss of generality we assume that…*". This simply means that they have made an assumption that can be easily changed, although they do not expect the results of their studies to change if this assumption is not followed. For example, I am convinced that the linear relation between time and costs was a reasonable assumption in the study of Chap. 4 that, if changed, would not lead to fundamentally new insights. That does not mean, however, that researchers can simply define assumptions without thinking too much about whether they are reasonable or not. For example, it is generally accepted that the choice of the statistical distribution to model the duration variability is a crucial choice to be made by the researcher. It is said that this statistical distribution choice (using, e.g., the normal distribution, beta distribution, lognormal distribution, etc.) and its parameter values can have a major impact on the generated progress data for the simulations. Consequently, this choice also has an impact on the specific results of the study, which is why most believe that this choice must be taken with great care!

But to be honest, I do not fully agree with that.

Of course, I fully agree that researchers should choose a distribution that is as close as possible to the actual real-life behaviour of projects, but I think that the specific choice of the *type of distribution* is not as relevant as some researchers (and reviewers of our research) think. I believe that the parameters of the chosen distribution (mean values, standard deviations, skewness, etc.) are much more important, because through these parameters the progress data are generated as projects that finish early or late (depending on the skewness) with a low or high deviation from the baseline schedule (depending on the standard deviation). Despite this, I have personally experienced that most researchers consider the type of distribution to be (more) important than the parameters of the distribution, and I do not fully understand why.

Let me give the example of my first paper on *statistical project control*, which I discussed in Chap. 5 and was eventually published in the highly respected journal *Omega—The International Journal of Management Science* (Vanhoucke, 2010). In the first version of this study, I simulated the variability of activity durations using the so-called *triangular distribution*, a very simple distribution requiring only three-point estimates. I knew that this choice went a bit against the common practice of the academic community, but I thought keeping things simple would make the paper more readable. But this referee had a very different idea and could not accept my paper because of an *unrealistic choice of statistical distribution*. Fortunately, the referee was kind enough to offer some advice on how to improve the research study and told us (me and my co-author Jeroen Colin) to look at the literature and use a much better distribution to model the variability in durations. It was indeed known in the literature that activity duration variability can be best modelled by a distribution closely related to a *beta distribution*, with a mode (most common value) close to the deterministic estimate of the baseline schedule. The advantage of this distribution, according to the literature, lies not only in its realism, but also in the ease with which tails to the right (or left) allow a greater chance of delays (or earliness). Other studies in the literature give different recommendations and propose distributions such as the *beta rectangular distribution* (i.e., a mixture of a beta and a uniform distribution) (Hahn, 2008), the *generalised beta distribution* (Kuhl et al., 2007), and the *lognormal distribution* (Mohan et al., 2007). So it was not easy to follow the referee's recommendations and choose the right type of distribution, as there was no unequivocal opinion in the literature. In any case, it was clear to the referee that the triangular distributions could not be used for academic research. So I had to choose a more common (i.e., more complex) distribution at all costs to model variability in the activity durations; otherwise, my research would have had no chance of success (getting published).

I have learned from experience that listening to referees and taking their advice seriously is the best thing that you can do if you want to get your research published. In the end, I decided to use the so-called *generalised beta distribution* because it is much more manipulable than triangular distributions (and because the referee gave a slight hint in that direction). The problem, of course, was that I had to rerun all the experiments, so I went back to working on the supercomputing infrastructure at my university and ran all the experiments again. After the new experiments, I rewrote the paper with updated results and submitted a revised version and got an acceptance this time. *Yes!* In the published version of the article, I wrote the following comments, inspired by the reviewer's comments:

> In this simulation study, the choice of using the generalised beta distribution has been based on the comments made in Kuhl et al. (2007). These authors motivate that the generalised beta distribution is generally a better choice than the triangular distribution in cases $c - a \ll b - c$ or $c - a \gg b - c$, that is, in situations in which there is a pronounced left or right-hand tail on the distribution of the stochastic variable (activity duration).

Obviously, I was very happy with the publication of our research, but I honestly never really understood why the study was now so much better with those complex distributions than when we used the simple triangular distributions. In any case, the results were not fundamentally different, and I hardly had to change any conclusion except for some rounding of numbers. To a layman, the triangular distribution, the lognormal distribution, and the beta distribution are very similar in some forms, but a statistician will have a very different opinion. The statistician will argue that all positive values in the domain have a positive probability of occurrence in a lognormal distribution, while the beta distribution has the characteristic that positive probability values only exist for a limited range of values. The generalised beta distribution (the distribution we ended up using) is much more flexible and can take many different forms depending on the parameter values, including a skewed unimodal distribution that is very close to that of a lognormal distribution. I do not want to go into detail, but the generalised beta distribution uses many parameters and the values of these parameters can be set to be very similar to any of the distributions discussed earlier. I once have made four graphs for the four distributions in R and plotted them side by side. I could easily set the parameter values so that you do not really see a big difference, which is why I do not quite understand why it all matters so much.

I am not telling this story to complain about the journal's review process, nor to say that the referee's comments were useless. On the contrary! Besides, the referee is always right, even if the comments do not make much sense, because the publication depends on it. I am just telling this story to show that this is the life of a researcher: *the quality of research is in the smallest details*. What I have learned from this long review process is that some choices are more readily accepted than others, and it is often much better to follow the generally accepted guidelines provided by previous studies. I initially thought running the experiments with new distributions again was a huge waste of time, and when I saw that I could not draw any new conclusions, I was even more convinced of the futility of looking for the most appropriate statistical distribution. In retrospect, I had to refine that vision as it was perhaps the most interesting referee report that I have ever received. Indeed, the comparison between different distributions has made me think about how the distributions are best chosen, and this has piqued my interest in a new stream of research in which I wanted to look for the best parameter values to estimate variability. This new stream of research is presented in Chap. 14, which examines the most appropriate choice for a realistic activity duration distribution. It will be shown that the *lognormal distribution* is a good candidate for modelling variability in the duration of activities based on theoretical arguments and empirical evidence. I want to thank the referee for this great idea for future research. Indeed, the quality of the research often lies in the details (as requested by most referees).

12.3 Risk Model

In the variation model, statistical distributions were used to model variability in activity durations, and the assumption was made that costs were linearly dependent on them. However, what I have not explicitly stated is that an additional assumption presumes that the variability in the activities was generated *independently* from each other, meaning that a change in one activity (relative to the planned duration) does not necessarily involve a change in the other. This may be a realistic assumption in some cases, but it is easy to find examples where this is not true at all. The risk model omits this assumption and adds *dependences between activities* in a very simple, but elegant way.

There are numerous reasons to believe that activities are correlated with each other, in which a problem in one activity causes a problem in the other activity. Just think of sharing the same resources between activities, where if one activity is delayed, the other simply cannot start because the resources are not available.

Of course, assuming independence does not mean that there are no connections between activities in the variation model. The precedence relationships in the project network still mean that a delay in one activity can influence the successor activities and thus leads to a general delay of the project. But beyond this dependence in the network logic, the variation model assumes that there are no correlations between the variability of different activities. The *dependence* assumption used in the risk model stems from the observation that there exist very clear dependences between the durations of activities in practice. For example, it may be that there is bad weather for an entire period, which for an open-air project can affect all activities that are carried out during that particular period. However, if project management professionals want to specify correlations between activities to model these dependences, they face a challenging task as there is no established methodology for estimating correlation coefficients between activities. Even researchers struggle to model these dependences in the variability between activities (and therefore often use the variation model as a simplification), as we wrote ourselves in our article (Colin & Vanhoucke, 2014):

> If researchers are required to specify correlations between activities, they face a challenging task since there is no established methodology for estimating correlation coefficients. In order to reject the independence assumption in our simulations, we employ the linear association technique, where a systemic error B provides objective information on correlations.

So we had to look for a simple and easily implementable way to bring dependences into our progress data, and after a search in the literature, we came up with a very elegant, yet simple concept. The so-called *linear association technique* to model dependences between activities is a concept initially proposed by Trietsch and Baker (2012) to extend the variation model to the risk model. As with a linear relationship between two variables, where any given change in an independent variable will always cause a corresponding change in the dependent variable, the linear association approach assumes something similar happens with the durations of two or more random activities. The beauty of using the linear association approach to model activity dependences lies in its simplicity and ease of implementation as it does not require advanced statistics nor any additional features to very clearly distinguish activity variation and risk. Instead, it simply assumes that the activity *risk* (*dependent variability*) is added to the activity *variation* (*independent variability*) by drawing random numbers from two, rather than one, predefined distributions. More specifically, the first distribution is used to generate independent variability (as in the variation model), while the second distribution adds an extra dimension to this variability in order to imitate statistical dependences between activities. Implementing the risk model to generate progress data for activity durations is done using the following three steps:

Step 1. Simulate variation: In a first step, a real duration $\hat{d}_i$ is simulated for each activity from a statistical distribution as discussed in the variation model (distribution 1). As discussed earlier, these actual activity durations are simulated separately for each activity and are assumed to be independent of each other.

Nevertheless, these durations may differ from the initial duration estimates of the baseline schedule and thus may result in a project that is ahead of schedule or a delayed project, identical to the variation model.

Step 2. Simulate risk: In a second step, a general systematic error B from a statistical distribution is simulated (distribution 2). The systematic error B should ideally provide objective information about correlations between activities and represents *external events* that can occur in the project and affect all activities. The example of bad weather mentioned above is such an external event and can therefore have a negative influence on the entire project. Another example that is often used is the systematic underestimation of duration estimates in the project schedule, where the real durations are a factor B higher due to the fact that the initial estimates were too optimistic. This bias term B should ideally be regressed from historical data (if real data are present), but can also be generated randomly from a probability distribution (e.g., the lognormal distribution is proposed by the "founders" of this technique). In the latter case, the inputs are, of course, largely dependent on subjective estimates for the parameters of these probability distributions, but this was also the case with the variation model. We will see later in Chap. 14 that calibrating the parameters to historical data should always be preferred.

Step 3. Integrate variation and risk: In a third step, the independently generated durations of Step 1 and the systematic error of Step 2 are brought together. More specifically, each generated activity duration $\hat{d}_i$ is multiplied by the systematic error B to obtain a final activity duration $B \times \hat{d}_i$. If we want to put it a little more formally, then n positive random variables Z_i are linearly associated if $Z_i = B \times X_i$ where $\{X_i\}$ is a set of n independent positive random variables (variation model) and B is a positive random variable, independent of $\{X_i\}$ (systematic error of the risk model). The $\{X_i\}$ variables here represent the durations $\hat{d}_i$ that were generated by means of the variation model, where the B factor adds an extra form of uncertainty (risk) to the activities in order to model the dependences. Multiplying each generated activity duration by the B systematic error thus assumes that a portion of the activity delays comes from external sources that apply to all activities in the project (thus making the activities correlated).

According to the authors Trietsch and Baker (2012), the *lognormal distribution* can be used when statistical dependence is modelled with linear association. In their paper, they defend this assumption using both theoretical and empirical arguments, and I recommend that readers take a close look at this paper to better understand the linear association concept. The practical implementation, however, is quite simple as it requires the generation of two values for uncertainty (which I called *variation* per activity and *risk* for the whole project), which makes the risk model in my opinion preferable to the simple variation model. I had the privilege of meeting Dan Trietsch, one of the authors of the article proposing the linear association concept, at my home university in Ghent (Belgium) in 2019. He gave an inspiring research workshop for the members of my team, and we had a lot of interesting discussions afterwards about statistical models to model variability in activities. It was pretty clear that Dan masters the matter to the core and since then I have been using the linear association model in most of my simulations to generate progress data. Some of his ideas will also be discussed later in this book (Chap. 14), and I am grateful to Dan for inspiring my research team with his new theories.

12.4 Scenario Model

The two models discussed earlier make use of statistical distributions to model the variability in activity duration (*variation* and *risk* models) and external sources of risk (*risk* model), and I have argued that their parameters must be chosen well to mimic the actual project progress. This is completely different for the *scenario model* that will be discussed in this section for which the correct use of the parameters in the distributions is much less important. The goal of this third model is to control the simulations so that different scenarios are generated, and some of these scenarios are so extreme that they may not occur very often in practice. Nevertheless, all the generated scenarios, realistic or not, are interesting to study because they serve to answer a number of specific research questions.

To fully understand the scenario model, I have to take the readers back to the study in which different forecasting methods were compared using Earned Value Management (EVM, cf. Chap. 4). This comparative study used three methods (planned duration, earned duration, and earned schedule methods) to predict the duration of a project and led to our first joint paper with Stephan Vandevoorde in the *International Journal of Project Management*. You may recall that I wrote in this chapter that these results were presented at a number of workshops and that not everyone was equally pleased to see that the *earned schedule* method outperformed the traditional *planned value* and *earned duration* methods. I still do not quite understand why there was so much controversy about our research results, but I think it may have to do with the level of control in the *work breakdown structure* (WBS), a theme that was also discussed repeatedly in previous book chapters. I mentioned this issue for the very first time in Sect. 3.3 where I referred to a quote from Walt Lipke from his article in the same journal in which he argued that the proper level of control should not be done at the activity level because that would lead to too much detail. He argued that it is therefore much better to go to higher levels where an overview can be obtained more easily. Much later, in Chap. 7, I came back to that by introducing the WBS, and calling this form of EVM control *top-down project control*. The reason for this naming came from the observation that performance measures for time (schedule performance index) and cost (cost performance index) are best measured at the high WBS levels, and only when these indices show values representing problems, the project manager should descend to the activity level to find the causes of these problems.

I know I am repeating myself, as these concepts were used continuously throughout the book. It should be noted, however, that using performance measures at high WBS levels (with the necessary drill-down when thresholds are exceeded) may be the only realistic option a project manager has to efficiently monitor ongoing projects, but that this choice also carries dangers. It is obviously much more accurate to measure the performance measures at the activity level (i.e., the detailed level at the bottom of the WBS) because then every small change will be measured, but such a detailed control of the project would take far too much work and time. However, not everyone agreed with the view that the level of control should be at higher levels, and I wrote the following words in my book "*Measuring Time*":

> This concern has also been raised by other authors and has led to a discussion summarised in articles such as Book (2006a,b), Jacob (2006), and Lipke (2006). Although it is recognised that, at high WBS levels, effects (delays) of non-performing activities can be neutralised by well performing activities (ahead of schedule), which might result in masking potential problems, it is the only approach that can be taken by practitioners. Indeed, the earned value metrics are set up as early warning signals to detect problems and/or opportunities in an easy and efficient way (i.e., at the cost account level, or even higher), rather than a simple replacement of the critical path based scheduling tools. This early warning signal, if analysed properly, defines the need to eventually drill down into lower WBS levels. In conjunction with the project schedule, it allows to take corrective actions on those activities which are in trouble (especially those tasks which are on the critical path). Lipke et al. (2009) also note that detailed schedule analysis is a burdensome activity and if performed often can have disrupting effects on the project team. EVM offers calculation methods yielding reliable results on high WBS levels, which greatly simplify final duration and completion date forecasting.

It was precisely this controversy and difference of opinion that prompted a second study with Stephan, in which the scenario model was proposed. We wanted to find out whether controlling projects at a higher level is much more error-prone than controlling them at the low activity level. It is indeed true that the higher-level performance indicators (SPI(t)[1] and CPI) can give incorrect warning signals (e.g., that a project is overdue, when in fact it is not), something that will not happen if each activity is monitored individually. Perhaps, this potential error is so large and so common that some could simply not accept the advice of the top-down methodology. I will not repeat the results of our second study because they were summarised in Sect. 4.2, but here I will take a closer look at the nine scenarios that we have defined in the scenario model to provide the best possible answer to this new interesting research question. The nine scenarios to model the variability of the duration of activities are shown in Fig. 12.2. For each scenario, a random duration per activity is simulated from statistical distributions. These distributions are not as tightly controlled as in the previous models, and sometimes those numbers come from distributions with very extreme values for the parameters (average and standard deviation) to steer the data generation in the desired direction. The following sections will show that these scenarios are specifically designed to extensively test the much-discussed margin of error of top-down control, and therefore we sometimes have had to simulate very extreme, perhaps not so common,

[1] Remember that the SPI(t) abbreviation was used to represent the new *schedule performance index* of the *earned schedule method*, which is more reliable than the classic schedule performance index (SPI) because the SPI always ends at 100%, even for projects that finish late.

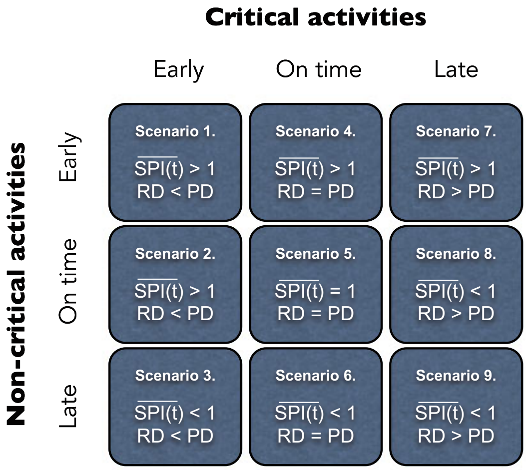


Fig. 12.2 Generating progress data with the *scenario model*

scenarios. The figure looks quite simple, but it is actually not. It contains a lot of data to properly understand the nine scenarios, which can be classified into three groups (*activity criticality*, *warning signals*, and *project status*) that must be well understood in order to understand the difference between the nine scenarios.

Activity Criticality First of all, the figure distinguishes between the variability in the duration of critical and non-critical activities. Both a critical and non-critical activity can be ahead of schedule (*early*), have a duration exactly equal to the scheduled duration (*on time*), or suffer from a delay (*late*). The reason for this division is very important in order to answer our research question. It is quite clear that delays in critical activities are potentially more damaging to the project than delays in non-critical activities. After all, for the non-critical activities, delays can be (partially) offset by the activity slack, which eventually leads to a small (or even no) delay of the project. The three possibilities (early, on time, or late) for the two classes of activities result in the nine scenarios of the figure, and it will be shown that the generation of the progress data with statistical distributions is different for each scenario.

Defining duration variability in critical and non-critical activities:

Early	:	Activity ahead of schedule
On time	:	Activity on time
Late	:	Activity delay

Warning Signals The generation of real durations for critical and non-critical activities leads to a project execution that deviates from the baseline schedule. During these simulation runs, the progress of the project is measured at regular time intervals (the so-called *tracking periods*), and the performance measure SPI(t) is measured periodically to monitor the time performance of the project. For each tracking period, the SPI(t) value can indicate a project that is ahead of schedule (SPI(t) >1), on time (SPI(t) = 1), or delayed (SPI(t) <1) and, as is known, is used as a warning signal to possibly take actions. At the end of the project, these SPI(t) values are averaged over all tracking periods, resulting in an average value $\overline{\text{SPI(t)}}$ for the time performance of the project. This average value gives an indication of the average state of the project during all tracking periods from its start till its end, which can also be lower than, equal to or higher than 1 (or 100% if expressed as a percentage) as shown in the nine blocks of the figure. The first three blocks in the first row contain projects where, on average, early project signals are measured ($\overline{\text{SPI(t)}} > 1$). The next three blocks in the second row show a different average project performance, depending on the scenario, ranging from early warning signals (Scenario 2), on-time signals (Scenario 5), and late warning signals (Scenario 8). The last row consists of projects that, on average, show a signal of lateness ($\overline{\text{SPI(t)}} < 1$).

Measuring the average schedule performance of the project (during progress):

$\overline{\text{SPI(t)}}$: average early warning performance signal		
>1	:	Average positive signal (ahead of schedule)
$=1$	:	Average on-time signal
<1	:	Average negative signal (schedule delay)

Project Status A third criterion that was carefully controlled in the simulation was the final actual status of the project after it was completed. For this reason, a distinction is made in the figure between the *planned duration* of the project in the baseline schedule, abbreviated by PD, and the actual real duration after the project has ended (i.e., after the simulation run), abbreviated by RD. The first column of the figure shows projects finishing earlier than expected (RD < PD). The second and third columns represent projects that finish on time (RD = PD) or late (RD > PD).

Getting the final project status (after finish):

PD and RD: Planned and Real Duration of the project		
RD < PD	:	Early project
RD = PD	:	Project on time
RD > PD	:	Late project

At this point, it should be clear that simulating this average project performance warning signal status, measured by $\overline{\text{SPI(t)}}$, was not easy. In fact, for some scenarios, the simulations were a huge challenge. For example, in Scenario 3, the critical activities finish early, while the non-critical activities are late, and since the mean SPI(t) value indicates late project behaviour ($\overline{\text{SPI(t)}} < 1$), the lateness of the non-critical activities far outweighs the earliness of the critical activities. However, as if that was not complex enough, Scenario 3 also consists of projects that eventually end too early (RD < PD). This means that the much heavier lateness of the non-critical activities should never fully consume the slack; otherwise, those activities would become critical and lead to a project that ends too late. Other scenarios were easier to simulate. For example, it is quite easy to get an average early SPI(t) statistic for projects with early critical activities and non-critical activities (Scenario 1), since in this case, the project will automatically finish early too (RD < PD). My point of this description is that the simulation of activity durations could not be done simply using statistical distributions with known parameter values, but instead had to be carefully designed to lead to the desired values for $\overline{\text{SPI (t)}}$ and RD. Since I had to push the random activity duration generation to the limit, it may sound like I was cheating to skew the results towards my desired outcome, but this was exactly the whole purpose of the study. I had to adjust the input parameters so that I could test exactly what I wanted to test, so I manipulated the input parameters to the limit to get these nine scenarios.

Nine Scenarios Keep in mind that the ultimate goal of the study was to measure the reliability of the SPI(t) warning signals by looking for errors when used at the highest level of the WBS. A warning signal is reliable if it measures the correct project status. This means that the average SPI(t) values lower than 1 (indicating a project delay) must indeed lead to an eventual project duration overrun. Likewise, SPI(t) values higher than 1 indicate early progress and should result in early project delivery. In all other cases, the EVM system is unreliable and thus an error occurs. The nine scenarios in Fig. 12.2 were therefore specially designed to model these reliable (without an error) and unreliable (with an error) warning signals. The scenarios are classified into three categories as described below, each with a different meaning and purpose:

True scenarios: Scenarios 1 and 2 report an average project "*ahead of schedule*" progress ($\overline{\text{SPI(t)}} > 1$), and the project finishes indeed earlier than planned (RD < PD). Scenarios 8 and 9 report an average "*project delay*" progress ($\overline{\text{SPI(t)}} < 1$), and the project finishes indeed later than planned (RD > PD). Scenario 5 reports an average "*on-time*" progress ($\overline{\text{SPI(t)}} = 1$), and the project finishes indeed exactly on time (RD = PD). Consequently, these five scenarios report a *true* situation (i.e., what you measure is what you get) and are therefore completely reliable.

Misleading scenarios: Scenario 4 reports an average project "*ahead of schedule*" progress ($\overline{\text{SPI(t)}} > 1$), but the project finishes exactly on time (RD = PD). Likewise, scenario 6 reports an average "*project delay*" progress ($\overline{\text{SPI(t)}} < 1$), but the project finishes exactly on time (RD = PD). Consequently, these two scenarios report a schedule deviation, but the projects finish exactly on schedule, and hence, they are called *misleading* simulation scenarios.

False scenarios: Scenario 3 reports an average "*project delay*" progress ($\overline{\text{SPI(t)}} < 1$), but the opposite is true as the project finishes earlier than planned (RD < PD). Scenario 7 reports an average project "*ahead of schedule*" progress ($\overline{\text{SPI(t)}} > 1$), but the project finishes later than planned (RD > PD). Consequently, these two scenarios report a false warning signal, and hence, they are called *false* simulation scenarios (as they are completely unreliable).

The results of this study were interesting, not only from an academic point of view, but also for professionals, as avoiding the potential error in the false scenarios was the exact reason why some did not believe in using EVM at high WBS levels. As I mentioned before, the main results were summarised in Fig. 4.2, and so I will not go over the results again in detail. However, I would like to remark that the study mainly showed that using EVM with the SPI(t) metric to measure project performance works well for the reliable scenarios (*true scenarios*) but fails miserably for the unreliable scenarios (*false scenarios*). While this sounds like an obvious result, I believe that it shows that using an EVM system as a top-down control system works surprisingly well (and therefore there is no need to monitor every single activity of the project). Indeed, if the SPI(t) system works well when the input is reliable (true scenarios) and works poorly when the data are unreliable (false scenarios), that is a very clear indication that the system is simply working fine (*garbage in, garbage out*). In any case, this system is much better than any other system that delivers average performance, regardless of the reliability of the scenarios, which indicates an arbitrary system rather than a well-functioning system. This also explains why the results showed that an earned value/schedule system is more reliable for project control when the network structure is closer to a serial network than to a parallel network. Since serial networks have more critical activities, the chance of unreliable measurements decreases significantly because any delay in an activity almost automatically leads to a project delay. These research results have convinced myself, and many others, that the use of the *earned schedule* metrics for project control as a top-down control system (high WBS levels) can be classified as a reliable system. I hope that the controversy surrounding the appropriate level of detail for project control can therefore be closed. Until, of course, someone proves otherwise.

12.5 Fiction

In the last two chapters, the use of artificial project data was extensively discussed. The static data to create *baseline schedules* for the resource-constrained project scheduling problem were described in Chap. 11. The current chapter added dynamic project progress data to perform *schedule risk analysis* and *project control* studies as academic research. Both chapters thus showed that scientific research can count on a lot of project data, whereby the static artificial project data are generally available from the project data website, while the dynamic project progress data can be easily generated by one of the three simulation models from this chapter.

Despite the widespread use of these static project data and the dynamic progress simulation models, everything remains completely fictitious with no link whatsoever to the real world. In fact, the static project data are generated by setting the network and resource indicators to as many different values as possible so that there is a good chance that any real project is somewhere in the generated data (*spanning the full range of complexity*). To simulate the project progress data, statistical distributions must be used to model the variability of the activity durations, whereby the values of their parameters can also be set to different values, again in the hope that *real* project executions are imitated (*spanning the full range of possible realities*). The three models to generate these progress data try to imitate reality in three different ways and were referred to as the *variation* model (independent variability of activity durations), the *risk* model (dependences between activities), and the *scenario* model (forced simulations to measure the reliability of project control systems). All this was proposed to make the academic research as realistic as possible, but still, no guarantee can be given that this is indeed the case.

Nevertheless, I hope that after reading these two chapters, you are convinced of the importance of artificial data for academic research in project management. These data have steered the data-driven project management research into all sorts of interesting directions, and I am convinced that without such data the state of research would not be that far at all. However, if you are still not convinced of the usefulness of artificial project data, then I will not try to convince you further and refer you to the next chapter where we finally get into the real world. In Chap. 13, and also the subsequent chapter, I will mainly focus on the use of empirical project data for scientific research.

References

Book, S. (2006a). Correction note: "Earned Schedule" and its possible unreliability as an indicator. *The Measurable News, Fall*, 22–24.

Book, S. (2006b). "Earned Schedule" and its possible unreliability as an indicator. *The Measurable News, Spring*, 24–30.

Colin, J., & Vanhoucke, M. (2014). Setting tolerance limits for statistical project control using earned value management. *Omega The International Journal of Management Science, 49*, 107–122.

Hahn, E. (2008). Mixture densities for project management activity times: A robust approach to PERT. *European Journal of Operational Research, 188*, 450–459.

Jacob, D. (2006). Is "earned schedule" an unreliable indicator? *The Measurable News, Fall*, 15–21.

Kuhl, M. E., Lada, E. K., Steiger, N. M., Wagner, M. A., & Wilson, J. R. (2007). Introduction to modeling and generating probabilistic input processes for simulation. In S. Henderson, B. Biller, M. Hsieh, J. Shortle, J. Tew, & R. Barton (Eds.), *Proceedings of the 2007 Winter Simulation Conference* (pp. 63–76). New Jersey: Institute of Electrical and Electronics Engineers.

Lipke, W. (2006). Applying earned schedule to critical path analysis and more. *The Measurable News, Fall*, 26–30.

Lipke, W., Zwikael, O., Henderson, K., & Anbari, F. (2009). Prediction of project outcome: The application of statistical methods to earned value management and earned schedule performance indexes. *International Journal of Project Management, 27*, 400–407.

Mohan, S., Gopalakrishnan, M., Balasubramanian, H., & Chandrashekar, A. (2007). A lognormal approximation of activity duration in PERT using two time estimates. *Journal of the Operational Research Society, 58*(6), 827–831.

Trietsch, D., & Baker, K. R. (2012). Pert 21: Fitting PERT/CPM for use in the 21st century. *International Journal of Project Management, 30*(4), 490–502.

Vanhoucke, M. (2010). Using activity sensitivity and network topology information to monitor project time performance. *Omega The International Journal of Management Science, 38*, 359–370.

Chapter 13
Empirical Projects

In Chap. 11, an overview was given of artificial project data to allow researchers to test their models under a broad set of assumptions. The importance of such research was underlined as a way to develop scheduling algorithms that can perform well under as many settings as possible, where it is not always necessary to check whether all these settings can occur in practice. The results of such research could potentially motivate new researchers to develop other, completely different algorithms, which perform better for some projects and less well for others, ultimately to continuously improve the state-of-the-art in algorithmic developments in project scheduling. However, the research should not stop after these studies on artificial project data, and the theoretical results and insights obtained must continuously compete with observations from reality in order to make the research relevant for practice.

Empirical research is research using *empirical evidence*, aimed at gaining insights through observations rather than through artificial well-controlled experiments. An empirical study is based on practical experience to confirm or reject existing or new theories and is often considered superior by project management professionals. This is not surprising, of course, as the goal of all data-driven project management research is ultimately to increase our understanding of the key drivers that influence project performance. Therefore, empirical project data—rather than artificial data—are best used as a source of inspiration for academic research, since by definition they contain *real* sources of uncertainties and risks that are never fully replicated in artificial data studies. However, the best research results are obtained when both perspectives (general results with artificial data or specific observations based on empirical data) are well combined. In scientific circles, one often speaks of deduction and induction. In *deduction*, an inference is often made from the general to the particular, and the inference is thus a logical consequence of the assumptions. The *inductive* methods often work the other way around, whereby a general rule is defined from a number of specific observations (generalisation). The

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_13

greatest scientist among all scientists, Albert Einstein[1] often started with thought experiments that later changed the way he thought about his experiments and summarised the true nature of research as follows:

> The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms.

I have never really thought much about whether my research is deductive or inductive,[2] but I have understood that both are best combined and that therefore both theoretical results (*on artificial data*) and real observations (*on empirical data*) can be an enrichment. I already mentioned in Chap. 10 that empirical project data are often preferred by project managers, simply because these data are much richer and therefore more realistic. I also argued in that chapter that artificial project data are more structured and thus can be more easily adapted to the specific subject of the study (which is why I paid so much attention to them in Chap. 11). This chapter tells the story of the collection of empirical project data by my OR&S group, which has led to what is, to my knowledge, the largest publicly available empirical project database for research purposes.

13.1 Curiosity

Regardless of the class your particular research topic belongs to, I believe that the true nature of research is *curiosity*, seeking answers to questions, or formulating questions that you do not fully understand at first. I have always really enjoyed discussions with fellow researchers where we came up with the most relevant and irrelevant questions, where we were only interested in the answers but certainly not in the practical implications. I suspect that this is the main reason why I mainly carried out research on artificial project data in the first decade of my academic career and felt absolutely no need to test my results in practice. Albert Einstein, he again, has put it this way:

> Curiosity has its own reason for existing.

[1] I sometimes wonder if there is anyone who is not a fan of Albert Einstein. His theories are one of the most important chapters in scientific history and have contributed to the modern understanding of our universe. In my project management lectures, I cannot help but talk about this scientist, and I confide to my students that my *Measuring Time* book was actually inspired by his theories. This obviously does not make much sense, but I can then easily steer the class discussion to the fact that Einstein was born on the same day as me: π-day. I already told you before (in Chap. 11) that I share the same birthday with my scientific superhero.

[2] There are a lot of articles to find out whether your research is inductive or deductive, and I still do not know to which class my research belongs. Some say that inductive research is an innovation, while deductive research is a discovery. Others claim that inductive research proposes a new theory (*experimental study*), and deductive research is to test the theories with data (*empirical study*). Let us not think too much about it.

That gradually changed when I started lecturing to people with professional experience. I suddenly realised that not everyone is as enthusiastic as I am about optimisation algorithms and automated methods, and I learned from my MBA students at Vlerick Business School that academic curiosity must be accompanied by practical relevance. Teaching these MBA students has literally opened my eyes as they have made me aware that I also have to look at the outside world. In particular, I learned from them that the interaction between my theoretical experiments and their practical experience can be enriching and that this interaction can improve the quality of my lectures, but also of my research. It is mainly thanks to them that I have decided to forgo the path of relying solely on artificial project data and started to collect empirical project data.

The search for empirical project data started sometime in early 2003, but I quickly realised that collecting empirical project data is much more than receiving some MS Project files from professionals and merging them into a new empirical database. Despite the fact that most project managers claimed that they had a lot of data available, this was very disappointing when I finally wanted to collect their data. I gradually learned that collecting empirical project data requires a formal process and that it can therefore also be the job of an academic to define this formal process. In the next sections of this chapter, I will briefly describe the process that I developed, refined, and finally implemented with several members of my team, ultimately arriving at an empirical dataset of 181 projects.

13.2 Classification

I mentioned earlier in Chap. 11 that the *classification* of projects is key in the process of generating *artificial project data* to allow researchers to generate the data under a structured design in order to span the *full range of complexity*. However, project scheduling is only the first component of the *dynamic scheduling* framework of Fig. 3.1, and more and more researchers have expanded their research from project scheduling to the two other components *schedule risk analysis* and *project control*. They realised that the artificial project data do not contain data for assessing project risks or measuring project progress. Therefore, researchers had to fall back on simulation studies to create additional (artificial) project progress data to extend their project scheduling research into dynamic risk and control studies. In Chap. 5, I have shown that both static simulations were necessary to calculate the sensitivity metrics of activities (*schedule risk analysis*) as well as dynamic simulations to simulate fictitious project progress (*project control*). These static and dynamic simulation models have been discussed in more detail in Chap. 12 where fictitious risk and progress data were generated for artificial projects using three different models.

Fortunately, empirical project data are much richer in nature. Since these data come from real-world observations, they contain not only known values for the static project characteristics (network topology and resource scarcity indicators), but also real dynamic progress data on how the projects were actually executed. The empirical project data can therefore not only be used to test algorithms for *baseline scheduling*, but are also very interesting for *schedule risk analysis* and *project control* studies. In that regard, empirical project data are better suited than artificial data and do not require simulations at all to generate progress data. However, this advantage comes at a price. First and foremost, the empirical project data come from people, usually project managers who do not have much interest in nicely structuring the data. Therefore, researchers analysing these empirical projects often struggle to understand exactly what all the numbers mean. Moreover, despite the empirical data being richer than the artificial data, the empirical data are often incomplete, missing some data points or containing unreliable or sometimes completely wrong numbers that require additional interpretation and assumptions for a better understanding. For these reasons, we decided to classify the empirical project data differently from the artificial data, proposing two new indicators—*completeness* and *authenticity*—to validate the quality of the data and to facilitate their use for academic research:

- **Completeness:** Data completeness is measured as the extent to which each of the three dynamic scheduling components (*schedule*, *risk*, and *control*) is covered by the project data and is expressed by a three-level colour code based on the traffic light approach represented by Anbari (2003). A green, yellow, and orange colour indicates complete, moderate, and rather poor completeness of the data, respectively. The *baseline schedule* dimension is said to be complete when all details for the project network are included, as well as data for resources and costs used by the project activities. For example, projects that do not use resources can only be used for simple planning calculations for critical paths and are therefore not complete. The *schedule risk analysis* dimension is complete when non-standard risk distribution profiles for the duration of activities are defined. The default distribution used is the triangular distribution with symmetric tails to the left and right, but when these distributions are replaced by other (more detailed) distributions, these data are considered more complete. The *project control* dimension requires periodic data on actual durations and costs to generate performance data using the EVM methodology. In the most basic way (and also in many chapters of this book), these EVM progress data are generated using dynamic simulation runs, but when these progress data are fully available (for each review period during project execution), this part of the project data is also considered more complete.
- **Authenticity:** In addition to an indication of whether the data are complete or not, the concept of authenticity reflects the source of the data and the degree of assumptions that were made when entering the data. Full authenticity is said to be achieved when the project data have all been obtained directly from the actual project owner without any necessary adaptations or modifications by the data collector. A distinction is made between project authenticity that is used for the static data parameters and the tracking authenticity that is relevant for the dynamic data parameters. The *project authenticity* is high if all static parameters, including activity, resource, and baseline cost data, are obtained directly from the

actual project owner. Thus, full authenticity of such data implies that the project network is copied and pasted by the project owner without any modification and interpretation of the data collector. It should be noted that it is perfectly possible that no data are available for resources (resulting in a lower completeness value), while this could still be a completely authentic project if no assumptions were made for the other available static data. The *tracking authenticity* is used to assess whether the dynamic progress data are authentic or not. Full tracking authenticity is achieved when the progress data obtained from the project owner contain the actual start dates, durations, and costs of the activities for each tracking period, without any modification or assumption made by the project collector. Both project and tracking authenticity are evaluated using the same colour-coded approach presented for the completeness of the project.

It is worth noting that these two classification criteria are only relevant for empirical data and cannot be used for artificial project data. Since the artificial project data only contain static project data, they always result in 100% completeness and 0% project authenticity (for the *baseline scheduling* component). For the other two components (*risk* and *control*), the artificial projects are 0% complete and 0% authentic because the data just do not exist (except if they are artificially generated by the models discussed in the previous chapter, then they are 100% complete, but still 0% authentic). However, the opposite direction works much better, and the values for the project and resource indicators for the artificial project data can also be calculated for the empirical projects. As an example, consider the lower part of Fig. 7.8. This graph contains empirical project data collected from eight companies in Belgium and shows how close each project is to a fully parallel (left) or serial (right) project. We discussed earlier that this proximity is measured by the *serial/parallel* (SP) indicator, which is used to classify and generate artificial data.

13.3 New Library

Since the empirical data come from observations of reality (rather than the output of artificial data generators), the collection of these data is a time-consuming, cumbersome, and often frustrating task. Before I even thought of collecting empirical projects, I did not realise how spoiled I was when it comes to accessing project data. Generating a new database of artificial projects was just a click of a button on the data generator, and thousands of projects were generated in no time. Collecting my empirical database took several years, and it was a gradual process of trial and error, constantly adapting the way that I collected these data and changing the classification model discussed in the previous section. It started out as a fast and unstructured process without much thought, but gradually it turned into a standardised process that eventually resulted in a freely available project database of 181 projects (and still growing). This process spanned more than two decades in four phases, with all project data collected from the first two phases eventually being lost. It was not until Phase 3 that I decided to collect the empirical projects in a standard way. For each attempt during these phases, an article was published in the renowned *International Journal of Project Management*, which attracted interest from the field. It all started with a collaboration with a friend (Phase 1), followed by a collaboration with my master thesis students (Phase 2), then building further in collaboration with a PhD student (Phase 3) to finally end up where we are now with a database and a whole team contributing to this data collection (Phase 4). Let me give you a brief overview of two decades to build the empirical database that we ended up calling the DSLIB set (dynamic scheduling library).

Phase 1. 2004–2006 The first attempts to collect real project data started during my previously discussed collaboration with Stephan Vandevoorde who worked for *Fabricom Airport Systems* at Brussels Airport (Belgium). Together we worked on a number of Earned Value Management studies that I described in Part II of this book where we decided to use data from three projects of his company. We collected the data from MS Excel files, analysed them, and sometimes modified them to meet the needs of our study. We did not use a structured approach for collecting and analysing the data, relying solely on observing and reporting what we saw. Despite this simple approach, our observations resulted in the study in which we compared different methods for project duration prediction (as described in Chap. 4) leading to our first joint publication (Vandevoorde & Vanhoucke, 2006). Despite the fact that we only used three projects, it has started a lot more research in that direction with the aforementioned collaborations with Walt Lipke (USA) and Kym Henderson (Australia) in our European organisation *EVM Europe* (2009–2017). These collaborations evoke fond memories to this day, and despite the fact that I somehow lost these three projects over the years, I consider this fledgling search for real data as the moment where I started thinking about a formal way of collecting empirical project data.

Phase 2. 2007–2010 Driven by the initial success and interest in our research, I was looking for more empirical project data. In 2007, I started collaborating with my students *Business Engineering* at Ghent University. I gave them some fragmentary empirical project data from my own consulting activities and asked them to analyse the data for their master's thesis. At that time, I had just introduced my new project management software tool *ProTrack*, which could be used by the students to more easily analyse the existing project data. Most of the data were collected through interviews and/or converted from existing MS Project files to *ProTrack* files, and we focused on cleaning up all the project data to build a new empirical database. Despite the more structured approach compared to the first data collection with Stephan, a lot of the data were not yet *empirical enough* because we had to make quite a lot of assumptions that we only defined afterwards, when the project was finished and the contact with the company was lost. For example, we interviewed project managers for the design of the project network, which contains the activities and the precedence relations between them, but it was not always easy to reach agreement on the best possible project network structure. We therefore decided to create the so-called *project network templates* that all had roughly the same logical structure, and we used them to define the project networks in our database without any additional input (and endless discussions) from the project managers. The dynamic progress data reflecting the actual duration and cost of the activity were even more difficult to obtain, and they were often highly unreliable and inaccurate. We therefore used statistical curve-fitting techniques to analyse existing data and generate probability distributions that allowed us to simulate additional missing data. It was certainly not the most ideal way to obtain reliable data, but it was certainly better than the previous study, resulting in 48 new projects. The 48 projects were eventually classified into 13 classes of project templates from 8 different companies and were used in 2010 to compare our theoretical results from previous studies (using artificial data) with new empirical results. To my delight, the comparison showed very similar results between these two sources of project data (*artificial* and *empirical*) as shown in the top part of Fig. 7.8. This comparative study was published under the title "*Measuring the efficiency of project control using fictitious and empirical project data*" (Vanhoucke, 2012a). Unfortunately, only a few of these 48 projects were retained and most projects were lost (again). This time, the reason was that our software tool *ProTrack* was still in the development phase and most of the project data could no longer be read by the new releases of our software tool. We urgently needed a better approach to collect (and save) our empirical projects.

Phase 3. 2011–2016 To avoid further data loss, we decided in 2011 (*finally!*) to work in a more formal way when collecting empirical project data. Learning from the mistakes of the previous years, and with the good fortune that Jordy Batselier joined my team as a PhD student, we decided to build an empirical project database that could last longer than one research study. It soon became clear that Jordy was the right man in the right place. With his everlasting smile and constant dedication to cleaning up the mess of our data, he gradually replaced our random data collection process with a formally structured methodology. With our new ambition in mind to build the largest publicly available empirical project database for academic research, we recruited our best *Business Engineering* and *Civil Engineering* students and asked them if they were willing to closely follow project managers from different companies for two years. We trained our students to become familiar with the concepts of *data-driven project management* and created a structured and detailed tutorial on how to collect project data. We gave every student free access to our software tool *ProTrack* and wrote a tutorial for it that I briefly described in Chap. 2 (*"Dynamic scheduling on your desktop"*). This tutorial became mandatory reading material before the start of the two-year master's thesis for every student. Soon after the first students started using it, Facebook group pages appeared where students asked all kinds of questions about our software, which created a dynamic interaction between students and professional project managers. It was impressive to see our young ambitious students sharing ideas and opinions to collaborate in ways I did not know existed. Their enthusiasm and hard-working spirit made the whole process a fantastic journey in search of real data, and it was one of the most joyous periods of my career so far with my university students.

After the first graduations and successful master's thesis projects, Jordy started analysing the data of the students and developed the so-called *project card* approach. More specifically, he compiled a project summary card for each individual project file to provide a tool that categorises and evaluates all project data. Each project card is split into three main parts, which summarise statistics for the three components of dynamic scheduling (baseline scheduling, risk analysis, and project control). The project cards facilitate the evaluation of the initial database according to the previously discussed criteria of *completeness* and *authenticity*. In this way, gaps in the database attributes could be easily identified. For example, Jordy detected an under-representation of projects with real tracking data (i.e., progress data) (which was also the biggest problem in the first two phases) and aimed to expand the database only further with fully authentic data. This particular need for fully authentic projects with full tracking information was communicated to the data providers (i.e., the master students) and translated into the more visual and understandable requirement that both project and tracking authenticity should be given a green colour on the project card. These project cards have undoubtedly made it easier for students to collect better and more complete project data.[3] By following this approach, we ultimately presented an available set of empirical projects that exceeded all existing empirical databases in the project management literature in both size and diversity. On March 14, 2014, the day I turned 41, we had collected a total of 51 projects from 47 different companies from different sectors. The results of this study were published in—again—the *International Journal of Project Management* (Batselier & Vanhoucke, 2015a) under the title "*construction and evaluation framework for a real-life project database*". After this publication, our search for real data did not stop, and the following two years a new cohort of students joined our program, ready to continue this process of data collection. It was not until 2016 that Jordy decided to leave the OR&S group in search of other challenges, leaving behind an empirical project database of a total of 125 projects. His legacy is very valuable to academic research in data-driven project management to this day.

Phase 4. 2017–now After Jordy left, we decided to temporarily suspend the time-consuming process of data collection in order to focus on other research projects, but that break did not last long. We eventually started all over again with the collection of new empirical project data together with PhD student Annelies Martens (in 2017). We started by cleaning up the existing database and created a summary Excel sheet of our existing data. Then, we decided to collect new projects in collaboration with new master students at our university. PhD student Tom Servranckx joined us and also started collecting additional data for projects with alternative technologies. PhD student Jakob Snauwaert even collected some empirical projects with skilled resources. The efforts paid off as more and more researchers started using the empirical dataset. In 2018, I collaborated with a new Chinese PhD student (Jie Song)

[3] An example project card is given in Appendix G.

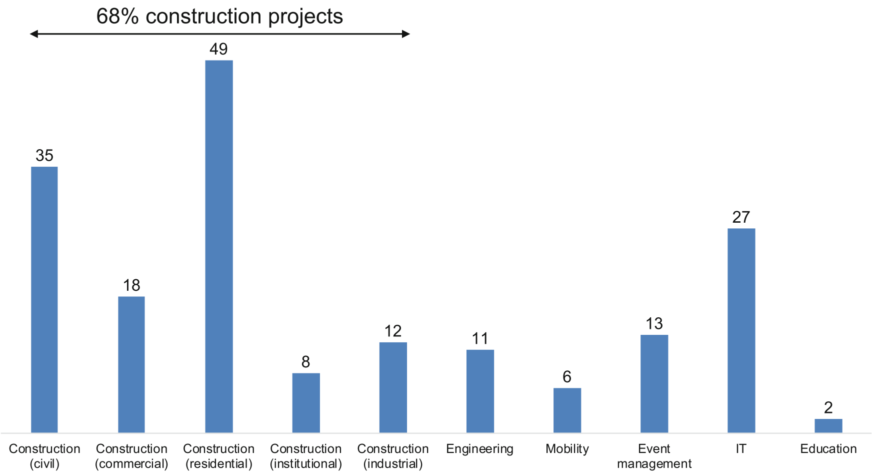


Fig. 13.1 The sectors of the empirical project database

who used our empirical database to develop new project control methods. We also collaborated with Paulo Andrade, a retired professional from Brazil with an interest in scientific research who began to analyse our empirical data (see e.g., Andrade et al., 2019, 2023). These research studies have resulted in a number of publications that no longer only use artificial project data, but also contain empirical case studies to validate our artificial results. In recent years, we have expanded both our empirical database and our artificial project database with many other project data, and a brief summary of our latest developments is given in Chap. 15 of this book. At the time of writing this book (2023), our empirical database has grown to an impressive 181 projects, all of which are available through the project data website that I mentioned earlier.[4] Figure 13.1 shows the distribution of the 181 empirical projects over the different sectors. The figure shows that 68% of the projects come from the construction sector, which is why we have split up this sector into subsectors. The different sectors for the empirical project database are as follows:

- **Construction**: The construction projects consist of construction works related to infrastructure (*civil*, e.g., bridges), projects for the private sector (*commercial*, e.g., offices, stores), house construction projects (*residential*, e.g., apartments), construction projects related to healthcare, education, recreation, or public works (*institutional*), and construction of specialised facilities (*industrial*, e.g., factories, warehouses).
- **Engineering**: Design and construction of engines, machinery, or processes.
- **Mobility**: Development of vehicle sharing platforms.

[4] If you cannot find the link in Chap. 11, the data can be downloaded from www.projectmanagement.ugent.be/research/data.

- **Event management**: Creation and development of events.
- **Information Technology**: Development of software and other IT technologies.
- **Education**: Program development for educational purposes.

Table 13.1 shows some important statistics for the empirical project database. It shows the average values and the range (minimum and maximum values) for the static planned project data, including the number of activities in each project, the planned project durations (PD, in days), the total planned budget (BAC, in €), and the number of resources. The table also shows that only 69 projects (38%) use renewable resources (ranging from 1 to 27 types of resources). The projects without resource data are scheduled using the *critical path method*, while the resource-constrained projects can be scheduled using the algorithms discussed in Chap. 11. Dynamic project data are not available for all projects, and 62% of the project database contains project time performance data (early (E) or late (L)), and 60% contains cost performance data (either under budget (U) or over budget (O)). The table shows the minimum and maximum percentage of time performance and indicates that the best project is completed 33% faster than planned, while the worst project has an actual duration that is twice (100%) the planned duration. In terms of costs, these numbers are even more pronounced with the actual cost of the best project being 57% less than the planned budget, while the worst project has a real cost that is 143% higher than the planned budget. The table also shows how many projects finish sooner or later (under or above budget) in the last two rows. Finally, the values for the network topology indicators of Table 11.1 (average, minimum, and maximum values) are displayed in the last four columns of the table.

13.4 Reality

I closed the previous chapter with the remark that the artificial data, both static and dynamic data, have stimulated a lot of research, but nevertheless always lie outside the real world, and are therefore completely fictitious. This chapter has solved this problem since the empirical project data contain both static and dynamic data. I could therefore end this book with this chapter and the beautiful message that both artificial and real projects are now available for academic research. Unfortunately, that story is a little too good to be true. Despite the high level of reality, it is not so easy to use the empirical projects in academic research for carrying out project control studies. The problem lies not so much in the quality of the data, but in the fact that the database only covers a very limited spectrum of all possible projects. Not every study needs this, and it is perfectly possible to draw general conclusions from a small amount of real data, but sometimes a researcher simply wants more. For example, if someone wants to use the empirical project data for the various simulation studies discussed in the previous parts of this book, then it is much better to simulate the progress data with *real* probability distributions coming from these empirical projects. But since each empirical project has only a single execution, it

Table 13.1 Key statistics for the empirical project database

	Planned data				Progress data		Network topology			
	# Act	PD (days)	BAC (€)	# Res	Time (E/L)	Budget (U/O)	SP	AD	LA	TF
Average	92.4	279.7	43,389,766	7.3	11.9%	3.3%	41%	57%	15%	36%
Minimum	7	2	1210	1	−33.2%	−57.8%	1%	17%	0%	0%
Maximum	1796	2804	5 billion	27	100.0%	143.6%	95%	100%	100%	100%
# Projects	100%	100%	86%	38%	62%	60%	100%	100%	100%	100%
% Under					−12.9%	−16.4%				
% Over					20.9%	14.0%				

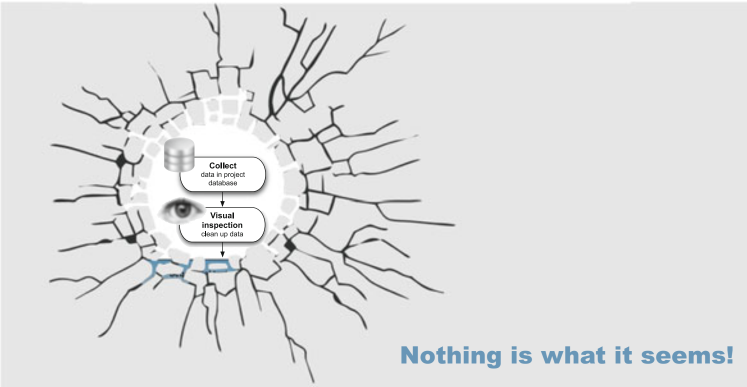


Fig. 13.2 Using empirical project data for academic research is not as easy as it seems

remains difficult to extract general distributions from that single sample, and the simulations remain dependent on the choice of the distributions (and its parameter values) as with the artificial progress models of Chap. 12.

I once discussed this thorny issue in a previously mentioned workshop in Berlin (Germany) where I gave a keynote presentation on using data for research. During this discussion, I referred to Fig. 13.2 to demonstrate that using empirical data for academic research is not so easy as it might seem. I wanted to show that the use involves more than a simple copy/paste of the data from a software package into a researcher's project database. After all, many people assume that making empirical project data available to researchers is a matter of signing a confidentiality agreement, then sending the data and that is it. However, before these data can be used, a lot of additional steps are needed, such as the classification described in this chapter (and many additional steps), and I strongly believe that more research is needed to perform those additional steps very accurately.

In the next chapter, I want to elaborate on one specific preparatory step for analysing the dynamic progress data of the empirical project and preparing them for further simulation studies. This preparation step will consist of a series of *calibration procedures* that convert the real progress data from a number of projects into statistical distributions with realistic parameter values, so that they can be used to generate new realistic progress data with the models from Chap. 12. This new calibration methodology consists of a series of tests that are a combination of: (i) statistical hypothesis testing, (ii) analyses to filter out inaccuracies and human errors, and (iii) expert judgements of the project manager who collected the empirical data. The three different versions of this data calibration method can be very useful in bringing fiction and reality closer together, in order to ultimately prepare both forms of project data (artificial and empirical) for academic studies.

References

Anbari, F. (2003). Earned value project management method and extensions. *Project Management Journal, 34*(4), 12–23.

Andrade, P., Martens, A., & Vanhoucke, M. (2019). Using real project schedule data to compare earned schedule and earned duration management project time forecasting capabilities. *Automation in Construction, 99*, 68–79.

Andrade, P., Vanhoucke, M., & Martens, A. (2023). An empirical project forecasting accuracy framework using project regularity. *Annals of Operations Research*, Forthcoming (doi: 10.1007/s10479-023-05269-7).

Batselier, J., & Vanhoucke, M. (2015a). Construction and evaluation framework for a real-life project database. *International Journal of Project Management, 33*, 697–710.

Vandevoorde, S., & Vanhoucke, M. (2006). A comparison of different project duration forecasting methods using earned value metrics. *International Journal of Project Management, 24*, 289–302.

Vanhoucke, M. (2012a). Measuring the efficiency of project control using fictitious and empirical project data. *International Journal of Project Management, 30*, 252–263.

Chapter 14
Calibrating Data

In Chap. 12, I showed that the selection of the appropriate *probability distribution*, with realistic values for its parameters, to model the variability of activity duration, can be done using different models. When academics rely on artificial projects to model project progress, the parameters of the statistical distributions are often set to a wide range of values to ensure that it covers the *full range of possible realities*. However, when these distributions are obtained with empirical project data, it should, in principle, be easy to choose the parameter values since the data with actual project progress are present. So it is tempting to think that the selection of the right statistical distributions and their parameters for imitating project progress is much easier when empirical project progress data are available. In fact, with the overwhelming amount of past project data available in companies, many think that it is super easy to predict the future progress after a straightforward statistical analysis of these data. In the world of big data, where data abound, the future is a matter of proper analysis. But it is not that easy at all…

I concluded the previous chapter, where I discussed empirical data, with the remark that it can be useful to have a lot of data to perform statistical analyses, but that it is also a big misunderstanding that the use of empirical data for academic research is easy. I am very pleased that it is much easier to get project data from companies today than it was 15 years ago, but there is a feeling from some companies that the academics can just use that data without any modification or deeper analysis. Nothing could be further from the truth, and I would even argue that the growing availability of data has made it all a bit more difficult, or at least more challenging. The use of empirical project data for academic project management research has always fascinated me,[1] so that I finally decided to start a number of studies that will be summarised in this chapter. It presents a number of statistical techniques for analysing project data in order to ultimately obtain

[1] The arrow in Fig. 10.1 pointing from the professional world to academia is "*today's challenge in project management*" and the subject of the current chapter.

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_14

realistic values for the probability distributions of activity durations. When such distributions can be obtained from empirical data, the models from Chap. 12 can be used by professional project managers who want to investigate not so much the full range of possible realities, but *one particular possible reality* for their specific projects. The techniques in this chapter, which I will call *calibration techniques* from now on, will therefore draw up a number of classes of distributions that can be representative for the future on the basis of a set of empirical projects from the past. Thus, the calibration methods assume that past projects can be used for future projects, and while this assumption may raise eyebrows, several techniques rely on this assumption. Just think of the *reference class forecasting* technique discussed in Chap. 9 where the future time and costs of projects were estimated by an analysis of data from the past. The methods in the current chapter follow a similar approach, but now look at the data from a more statistical perspective, and try to use some hypothesis testing and expertise from the project manager to predict the future.

I must confess that the studies in this chapter were met with mixed enthusiasm. During the submission of the papers to journals, referee reports were sometimes quite harsh and the belief in these techniques was sometimes low. However, other referees were very enthusiastic and clearly saw the potential of these techniques, which made me decide to continue with them. I think the reasons for these mixed reactions are twofold. First of all, the studies in this chapter are still in their infancy, and a lot of additional research is needed to improve them. Moreover, I think that the criticism also stems from the belief of many that past projects say little about future projects, and that therefore the whole idea of observations from the past to better understand the future is a meaningless task and certainly not an assignment for scientists. So, in my lectures on this subject, I often use the quote from Washington Post journalist Robert J. Samuelson who warned us to use the word *science* carefully in forecasting futures, as he quoted:

> Probably the only people left who think that economics deserves a Nobel Prize are economists. It confirms their conceit that they're doing 'science' rather than the less tidy task of observing the world and trying to make sense of it. This, after all, is done by mere historians, political scientists, anthropologists, sociologists, and (heaven forbid) even journalists. Economists are loath to admit that they belong in such raffish company.

Of course, I am pleased with the growing attention to the use of statistics in project data, and I do not mind at all that there is some opposition that makes some studies difficult to publish. Indeed, the different methods of data calibration described in the chapter are only the start of much more much-needed research, but I strongly believe that they can contribute to better project scheduling, better analysis of risk and better use of project control methods to measure project progress. I am particularly pleased that the calibration methods can be a first step towards replacing the *imagination* that I talked about in Sect. 12.1 by *real data* when designing project management and control studies. After all, that is why I wrote this book about data-driven project management.

14.1 Calibrating Data

As is always the case in academic research, there are of course several ways to convert empirical progress data into statistical distributions, but this chapter will draw on the ideas originally proposed in an excellent paper written by Trietsch et al. (2012). You may recall that I mentioned Dan Trietsch earlier for the *risk model* of Chap. 12, and it is also he (and his co-authors) who came up with the original idea of data calibration. When I first read his article, I was intrigued by their ideas and immediately had the reflex to test these methods on our empirical project dataset. I will describe the results of these experiments in the current section and then expand on how we have further refined and improved this calibration technique in Sects. 14.2, 14.3 and 14.4. After all, that is how it always goes with academic research: if it has been proven that something works well, then it can and should be done better.

The naming of the methodology—the odd term *calibration procedure*—is used to maintain consistency with the authors' original article, and I believe the term pretty much reflects what it does. Indeed, the procedure involves "*calibrating*" empirical progress data for activity durations in various ways to make them suitable for further research by testing whether they correspond to a predefined statistical distribution. Since empirical data are now assumed to be available for the planning and execution of the project and the projects have already ended, a calibration method relies on the following two sources of data:

- The *planned duration* of each activity i, abbreviated as PD_i, is known from the time a project schedule is drawn up and is a deterministic estimate made by the project manager prior to the start of the project.
- The *real duration* of each activity i, abbreviated as RD_i, is only known after the project has ended. Real durations of activities may differ from planned durations due to random events, management interactions, and much more. This results in a set of *early* ($RD_i < PD_i$), *on-time* ($RD_i = PD_i$), and *late* ($RD_i > PD_i$) activities in the empirical project database.

The calibration procedures start with these two data sources. More specifically, the procedure assumes a priori a certain distribution for the duration of the activity, and using the PD_i and RD_i data, it will test whether this distribution can actually be used to extract the real data. If the calibration method shows that the empirical data do not fit the distribution, the calibration methods will sequentially remove some data points (activities) from the project and continue on the remaining part of the activities to test back whether it follows the predefined distribution. When it can finally be proven that the predefined distribution is valid for a part of the activities or projects, then the values for the parameters (mean and standard deviation) can be estimated, which gives the project manager a tool to predict the future of a project specifically on the basis of simulation. The distribution defined a priori to model activity duration times is called the *Parkinson distribution with a lognormal core* and explained in the following paragraphs.

Parkinson distribution with a lognormal core
(lognormal distribution + Parkinson's law + rounding errors)

The central hypothesis in the calibration procedure is that RD_i/PD_i follows a *lognormal distribution*. The ratio of RD_i to PD_i for activity i is used by the calibration procedure as a test statistic for hypothesis testing and is easy to interpret: If $RD_i/PD_i < 1$, the activity ended earlier than planned, if $RD_i/PD_i = 1$, then the activity ended exactly on time, and if $RD_i/PD_i > 1$, the activity ended later than planned. Several arguments have been made in the academic literature as to why the lognormal distribution is a good choice for modelling variability in activity duration. The choice of the lognormal distribution is consistent with the assumption that the natural logarithm of that ratio, $\ln(RD_i/PD_i)$, is normally distributed, which is good news for our further analysis. In fact, the normal distribution is known, and even non-statisticians have heard of it. A wide variety of statistical methods exist for dealing with normal distributions, including hypothesis testing methods to test whether data follow such a distribution or not. And this is exactly what the calibration procedure will do.

However, this does not mean that the calibration procedure is reduced to a simple lognormality test, which would reduce the calibration method to a traditional hypothesis test to check whether the actual data fit an a priori assumption. However, the calibration method is so much more than that: It acknowledges that the empirical data were collected by humans (the project manager), and since people are strange[2] when they collect data, the calibration method takes that into account. Indeed, when data on the duration of activities are collected during project progress, the input values are not necessarily exactly identical to the actually observed values. People often report figures that may differ from the real duration for various reasons. The calibration method includes two human biases in the analysis, known as the *Parkinson effect* and the *rounding effect*. By including these two strange effects, the statistical distribution is not simply called the *lognormal distribution*, but the *Parkinson distribution with a lognormal core*, abbreviated as PDLC from now on. This distribution assumes that activity durations are indeed lognormally distributed, but the calibration procedure looks at these data from the human lens and tries to purify the data as best as possible from these human errors. These two effects of human behaviour—both of which cause activities to sometimes be incorrectly reported as being on time—are as follows:

Parkinson's law tells us that "*work expands to fill the time available for completion*" (Parkinson, 1957). In a project context, this means that employees are not always willing to report an activity as completed early. Indeed, they may benefit from labelling an activity as on time when they were actually completed earlier than expected because this would provide a safety buffer for similar activities in

[2] I cannot help but think about the beautiful song by The Doors "*People are strange*" when writing these words. An amazing song!

the future (when estimated from comparable historical activities). This effect is called *hidden earliness* and thus reflects Parkinson's law at the level of activity durations. When using empirical data, any calibration procedure must take this law into account to ensure that this human bias in the data is not simply carried over into the statistical analysis, leading to erroneous results.

Rounding effects are a direct result of the coarseness of the timescale used to report activity performance, usually a single unit of time is used for this. For example, if most activities have a scheduled duration of at least several weeks, the chosen base time unit is likely to be one week (i.e., 5 business days). However, if there are some activities in the project that take 3–4 days instead of the planned week (5 days), they would—due to the coarseness of the timescale—be reported as on time because 3–4 days would be rounded to a week.

Taking both human biases into account, the calibration procedure thus tests whether the relative empirical distributions RD_i/PD_i follow a Parkinson's distribution with a lognormal core. Thus, the procedure will set up a series of statistical hypothesis tests to test whether these lognormal distributions can indeed be accepted or not. An overview of this procedure is presented in Fig. 14.1. The figure shows the four sequential steps described in the following paragraphs. As I mentioned before, the procedure will repeatedly remove a series of activities that do not meet the lognormal assumption (or that were subject to the human biases) until the remaining part of the project does. When sufficient activities remain, the procedure has shown that the lognormal distribution can be followed and these activities are then kept in the database for which its parameter values can then be estimated. The four sequential steps, including feedback loops, can be described as follows (pay attention to some technical descriptions, even if I tried to keep them as short and concise as possible):

Step 1 (S1). Lognormal distribution test: In the initial step, the procedure assesses whether or not $\ln(RD_i/PD_i)$ is normally distributed when considering all activities in a certain project (which corresponds to checking whether RD_i/PD_i is lognormally distributed). First, the Pearson's linear correlation coefficient R needs to be calculated by performing a linear regression of the $\ln(RD_i/PD_i)$ values on the corresponding Blom scores (Blom, 1958). The Blom scores can be calculated as $\phi^{-1}[(i-3/8)/(n+1/4)]$, with i the index of each activity and n the number of activities in the project ($i = \{1, ..., n\}$). This formula provides the z-value for which the cumulative distribution function (cdf) of the standard normal distribution $N(0, 1)$ attains a probability of $(i-3/8)/(n+1/4)$. The calculated R can then be compared to the values tabulated by Looney and Gulledge Jr (1985), through which a p-value can be obtained. This p-value forms the basis for accepting or rejecting the hypothesis that the activity durations follow the PDLC. Given a significance level of 5%, the hypothesis is accepted when $p \geq 0.05$ and rejected when $p < 0.05$. It is very important to mention that S1 will in fact also be executed at the end of any of the following steps, albeit on a reduced selection of activities (i.e., not on the complete project). This will become more clear during the explanation of those steps.

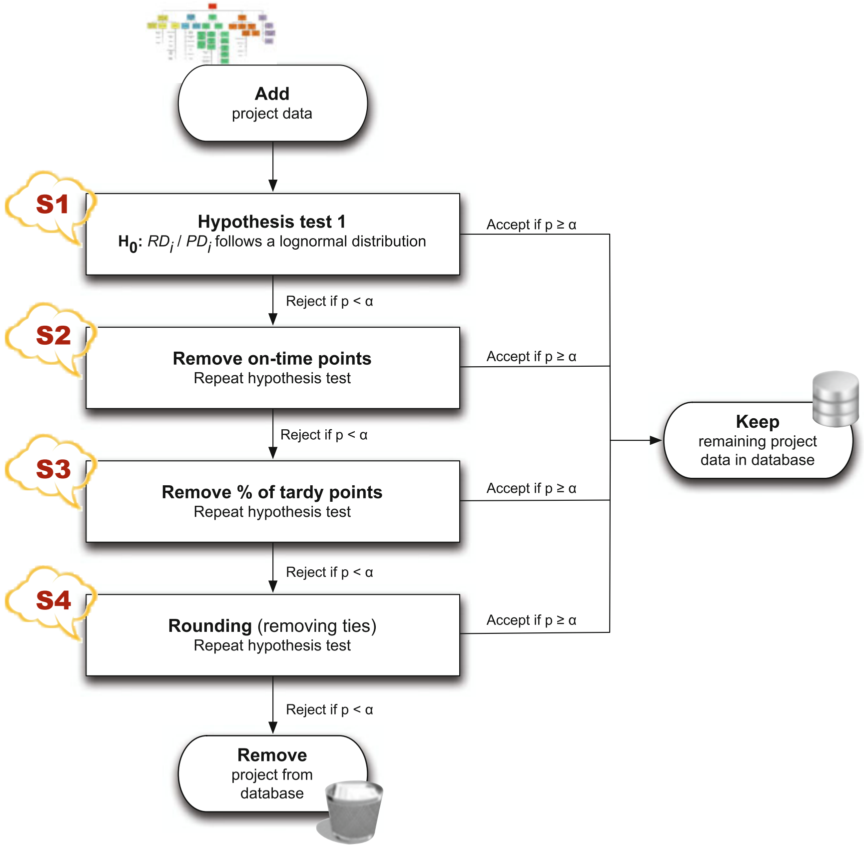


Fig. 14.1 Calibrating empirical data using hypothesis testing

Step 2 (S2). Parkinson's test: Remove on-time points: To take the Parkinson effect into account, all timely activities are taken out of the project. That way, the activities that were falsely reported to have completed as planned can no longer skew the interpretation of the duration data and affect the acceptance of the lognormality hypothesis. Step 1 (S1) is then run for a limited set of activities again, which now includes only the early and late activities of the project. It should be noted that removing *all* on-time activities may be a bit extreme as some of those activities may have actually been on time. In an extended calibration procedure (cf. later sections), a solution is therefore proposed to remove only *part of* the on-time activities, so that it can be assumed that not all on-time activities were subject to the Parkinson effect.

Step 3 (S3). Trimming: Remove $x\%$ tardy points: In this third step, a portion of strictly late activities is eliminated that is equal to the proportion of on-time activities in relation to the total initial number of activities. In other words, if we eliminated all on-time activities in S2, and this accounts for $x\%$ of all activities, then we now eliminate $x\%$ of the strictly late activities. In this way, we produce a trimmed dataset of activities that rebalances early and late activities. Note that S3 must always be run in conjunction with and after S2; otherwise, the ratio x would not be known. In addition, at the end of S3, it links back to S1 that again runs on a reduced selection of activities to test for lognormality.

Step 4 (S4). Rounding: Remove ties: Finally, the remaining activity data can be further modified by removing data with the same RD_i/PD_i values (the so-called *ties*). Concretely, this means that the average Blom score is calculated for all clusters of the so-called tied activities that show identical RD to PD ratios. It should certainly be noted that these tied activities are not merged into a single point, but rather into a series of coincident points to maintain their correct composite weight, which is necessary to ensure the validity of later calculations. After all, it is assumed that the mentioned ties arise from the rounding effect due to the coarse timescale for reporting activity performance, which therefore consists of activities that were all rounded to the same value. Just like for these preceding steps, S1 needs to be executed on the adapted dataset at the end of S4.

Aside from the technicalities that I have occasionally mentioned, I think that the procedure is understandable for everyone. The technical details are only important when coding the procedure, but it should be understood that the general idea behind the above calibration procedure is that the PDLC can be validated in any of the steps (S1 to S4) and that at each step a set of activities is removed from the project's original activity set. Each time, after removal, it goes back to S1 and retests whether the remaining activities follow the lognormal distribution. As is customary in statistics, S1 is executed with a p-value of 0.05. As the activity duration data become increasingly "calibrated" during the steps to better fit the proposed distribution, the p-values—and thus the probability of validation—are expected to increase as the steps progress (i.e., from S1 to S4). The procedure eventually stops when the remaining set of activities follows the lognormal distribution (or when this set is empty).

This calibration procedure was validated on 24 empirical projects from our database by Jeroen Colin (Colin & Vanhoucke, 2016), a PhD student that I mentioned earlier in this book for the tolerance limits research of Chap. 5. He achieved promising results, although he also saw a number of shortcomings that were later corrected in follow-up studies. For example, the original procedure requires S3 to remove some of the late activities but does not describe which activities to choose. The procedure only dictates that it must be $x\%$ and so several possible selections can be made, meaning the procedure can be run in different ways (with different results and different p-values each time). For some projects, it may even happen that the lognormality hypothesis is accepted after S3 in one run and rejected in the next. Obviously, this computational instability is not desirable, and thus the follow-up studies rightly resolved this issue. Some of the results of this calibration procedure will be briefly summarised in Sect. 14.5, but I will first move on to the more extensive calibration procedure that tries to solve some of the shortcomings of the initial calibration procedure as best as possible.

14.2 Partitioning Heuristic

Despite the promising results of the calibration method in the previous section, the procedure had one major drawback that we did not discover initially when we tested it on the 24 empirical projects but only discovered when we started using more data. When we used the procedure on a hundred projects, we saw that many data points (activities) were removed from the analysis (in steps S2, S3, and S4) and ultimately only a very small set of activities per project remained for which the lognormal distribution could be accepted. We initially assumed that this is mainly because the data were full of Parkinson effects and rounding errors, but after further analysis we saw that the calibration procedure makes an assumption that does not hold true. After all, it is assumed that all activities of the same project always follow the same lognormal distribution, as if they belong to one large class of comparable entities, which is of course not correct. Projects often consist of subprojects (phases), and the activities of one phase often have completely different characteristics than the activities of another phase. It therefore does not seem improbable that the probability distributions could also be different, and thus these activities are better treated separately. This observation inspired us to extend the calibration method to a *partitioning method* that first clusters project activities into subclasses and then uses the calibration procedure for each of the clusters individually. This new partitioning method is the subject of the next section.

The central idea of partitioning project data lies in the observation that the project consists of a series of activities that can often be placed in clusters of activities. Each activity should closely resemble one cluster but should not have much in common with the remaining activities of other clusters of the same project. The fact that projects are often planned in separate phases means that project managers treat each phase as a separate entity, a sequence of mini-subprojects as it were. Project managers implicitly treat their projects as consisting of clusters of activities within a project that I will now call *partitions*. This idea of dividing activities into clusters is not entirely new as we already made a similar attempt in Sect. 9.4 when *similarity properties* were defined for the *reference class method*. When we thought about how we could implement the partitioning method in the data calibration procedure, we first had to answer the question of whether we were willing to greatly extend the calibration procedure to a very *heavy* statistical method. The previous method from Sect. 9.4 took a very different approach and tried to classify the projects based on a list of managerial characteristics (cf. Appendix C), but never went to the level of activities (only the project similarity was looked at) and also never used any statistical technique such as the hypothesis testing from the previous section. The use of statistical techniques obviously had a number of advantages, mainly that it could work without the input of human judgements. However, there was the danger that it would lead to a fully automatic method that works without anyone having a very good understanding of what is going on. This choice between full automation and human adjustment is of course a choice that has to be made for many systems, and I have the feeling that nobody is afraid of some statistics and automation these days. That was once different, at the beginning of my career, where I often had to justify why my project management lessons were so quantitative. At that time, nobody cared about algorithms and data, and the soft skills were the only thing people wanted to hear. Nowadays, *data science* has become so popular that I sometimes have to warn my data-enthusiastic students about the danger of overusing statistics because it can lead to procedures that do something that nobody understands anymore. Times are changing. My students too.

Of course, the interaction between human intuition and data-driven algorithms is an old discussion, and as long as people made decisions, they relied on both (data and intuition) to make their decisions. Every decision, private or business, requires some form of data, and the more you have, the more likely you are to make better decisions. However, the amount of data has grown to such an extent that it is becoming difficult to distinguish between relevant data (the *signal*) and redundant data (the *noise*). While data were originally a supportive addition to human intuition, today it has often become a nuisance, and it cannot be ruled out that we may soon enter an era where we will look back with nostalgia for a time when data were not so abundant. In the book "*The Master Algorithm*" (Domingos, 2018), the author Pedro Domingos described the search for a master algorithm that will solve all problems for us based on data, and he wrote the following quote during this search:

> Data and intuition are like horse and rider, and you don't try to outrun a horse; you ride it.

The author assembled a blueprint for the future universal learner and introduced the readers to *the master algorithm* as the last ultimate thing humankind has to invent. From then on, this ultimate algorithm will take over all our problems and solve them spontaneously without human intervention. The author even goes as far as stating that data-driven tools with machine learning *"might even help you to become a better person"*. It is a wonderful book, and I recommend everyone to read this exciting story.

During the development of the partitioning heuristic, we decided not to go that far, and rather work on an algorithm that can combine both statistical data analysis (the *horse*) and human intuition (the *rider*). Figure 14.2 shows the general framework of the partitioning heuristic to allocate a project's activities into clusters (*partitions*). The figure shows that the heuristic consists of a manual partitioning step (left branch) and an automatic phase (right branch) to ensure that the rider (people with intuition) and the horse (data and algorithms) can work together. It will be shown that the so-called partitioning heuristic presented in the next two sections aims to unite the horse and rider in a single approach for improving the calibration procedure outlined earlier. It will be shown that the human partitioning branch focuses on characteristics of activities in the work breakdown structure, while the statistical phase will rely on a series of hypothesis tests to accept or reject the lognormal distribution assumption for early (E), on-time (O), or late (T) activities in each partition. Since some parts of the heuristic require human input and other parts can be fully automated using statistical methods, the extended calibration method should be better called the *semi-automatic partitioning heuristic calibration procedure*.

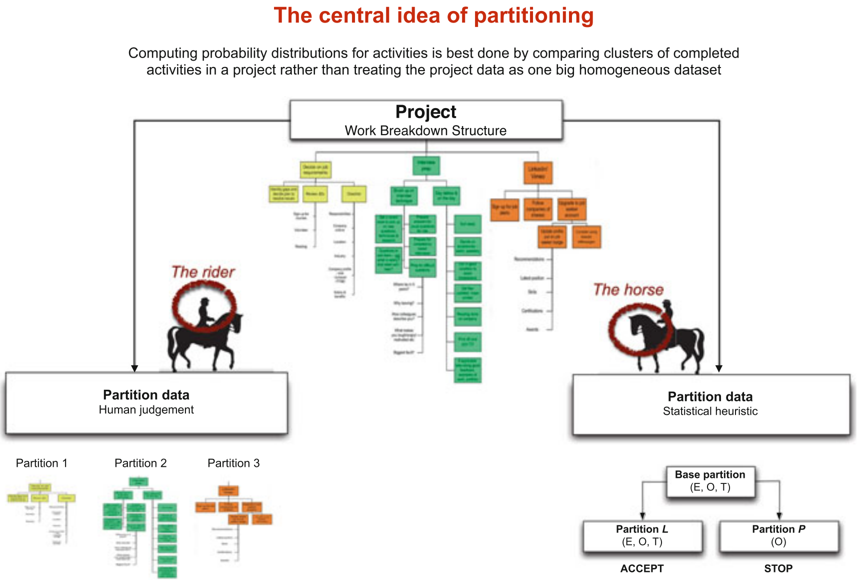


Fig. 14.2 Partitioning empirical data: The *rider* and the *horse*

14.3 Human Partitioning (*the rider*)

The idea of dividing the project into clusters arose from a discussion with Dan Trietsch, he again, and a study of 83 empirical projects (mostly construction projects) comprising a total of 5134 activities. As I have already discussed, the main reason for splitting the project data into partitions (clusters) lies in the fact that within a given project not all activities are perfectly related to each other and can therefore be very different from each other. For example, both very long and labour-intensive programming tasks of high complexity and routine software training afternoons can be part of the same IT projects. Even though intuitively both types of activity seem fundamentally different, it is not difficult to see that both would have different risk profiles, and thus a different statistical distribution for activity duration. If this logical classification of activities could already take place prior to the actual calibration procedure, it is likely that the PDLC could be accepted for a larger part of the activities. Thus, instead of testing the calibration procedure on all project activities simultaneously, it will now be tested on the data of each partition separately, potentially resulting in a better PDLC fit and thus statistical distributions with different parameter values for each partition. We published our findings in Vanhoucke and Batselier (2019a) where the *human partitioning heuristic* is called the *extended calibration procedure* to indicate that it extends the original calibration procedure with human input. Indeed, in the analysis of the 83 projects, we found that the partitioning heuristic could accept the lognormality hypothesis for many activities from different clusters. Figure 14.3 shows a summary of the human

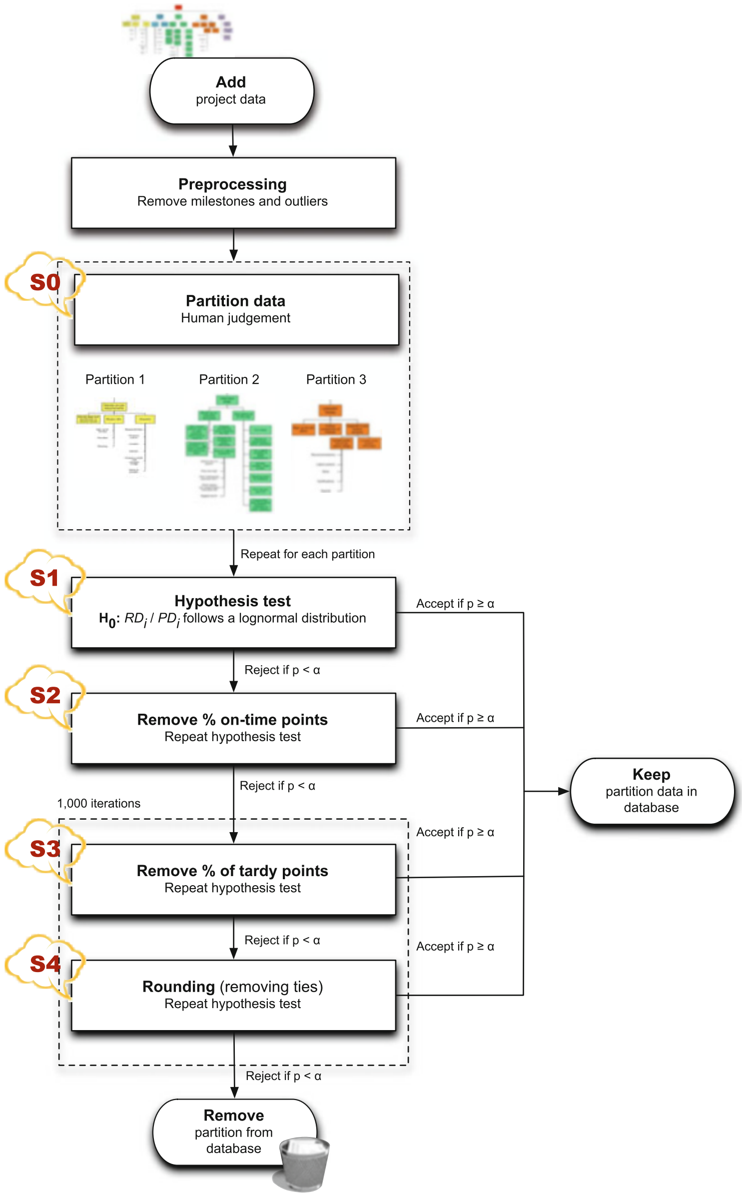


Fig. 14.3 Human partitioning (the *rider*)

partitioning heuristic, which consists of three different phases, as explained in the following paragraphs.

Phase 1. Pre-processing In a first phase, two classes of activities are removed from the project before the actual partitioning can begin. First of all, *milestone activities* should be removed as they do not represent actual work to be performed, but rather represent some sort of interim deadline with both a planned and real durations always equal to zero. Second, activities that could be considered clear *outliers* should also be eliminated. A first type of outlier occurs when an activity that was not planned is executed anyway, representing additional work that was not originally included in the work breakdown structure. In the analysis on 83 projects, an activity is identified as this type of outlier only when the activity name explicitly states "extra work", "additional work", or something similar. The second type of outlier refers to activities that were planned, but never actually carried out. Obviously, these kinds of activities can have any name, whether or not they were carried out. Both types of outliers actually and effectively alter the scope of the project. During the analysis of the 83 projects, it was necessary to eliminate only 66 activities because they were clear outliers (coming from only 9 projects), which seems negligible compared to an initial 5134 activities in total.

Phase 2. Manual Partitioning The remaining 5068 activities of the 83 projects are subject to manual distribution in a second phase, which relies entirely on human expertise (i.e., a *rider*, not a *horse*), indicated by S0 in Fig. 14.3. Because the set of empirical projects was collected over a time span of several years (cf. Chap. 10), contact with the project managers was lost and therefore easy and often pragmatic choices had to be made in the research study due to the inherent limitations of our real-life project database. More specifically, the criteria for human partitioning were limited to three criteria (planned durations, work packages, and risk profiles). Despite the decoupling of the data from the original project manager who collected the data for us, the empirical database contained project data rich enough to test the capability of human partitioning. Some details about the three partitioning criteria are given as follows:[3]

- *Planned durations*: Using activity duration values from the plan to manually assign activities to clusters is an easy way to create partitions. While PD_i reflects the expected duration of an activity according to the pre-project baseline schedule, the RD_i indicates the real duration of the activity after execution. Both measures are expressed in (working) hours and can be used as a separation criterion. However, in the study, we decided not to use the RD_i because this is only known *after* project completion and thus would not be suitable as input for a new project. The PD_i, however, is by definition available for any activity before the start of the project, as it reflects the plan (or baseline schedule) for the project

[3] We could have used the criteria described in the third study of Chap. 9 (Sect. 9.4) to partition the data, but these data were not available for the 83 projects that we used in the current study, and therefore the analysis was limited to only 3 management criteria:

under consideration, and can easily be used to define the partition of activities of a new project yet to be started. We therefore decided to use the planned duration as a partition criterion, and we adhered to the (rather arbitrary) rule that the PD_i of the longest activity in a partition should not be more than four or five times that of the shortest (in the same partition).

- *Work packages*: The assignment of work to a particular work package (WP) in the work breakdown structure (e.g., a particular phase in a construction project, performed by one and the same subcontractor) was done by the project manager from whom the data were obtained. The project manager sometimes identified multiple WP levels. If this was the case, we considered the WPs only at one chosen level, i.e., with enough activities to get a decent partition. In our study, WPs were defined for 53 out of 83 projects, with an average number of approximately 8 WPs per project and a maximum of 26 WPs for project C2013-02 from the empirical database.
- *Risk profiles*: The classification of the risk profile (RP) may need some explanation. First of all, an RP is actually an abbreviation for *activity duration distribution profile*. Such a distribution profile reflects the nature of the risk within a particular activity, hence the term "*risk profile*". When collecting the project data, four default RPs were identified (being symmetrical, left skewed, right skewed, and no risk), and the project manager was asked to assign each activity to one of these profiles. The default 3-point estimate values (i.e., best case, most likely, and worst case) for the default RPs can be seen in Fig. 14.4. The no-risk profile, of course, simply consists of one peak at 100% of the expected (i.e., planned) duration and is therefore not included in Fig. 14.4.

Phase 3. Automatic Calibration After the manual partitioning phase, the procedure continues with the four steps S1 to S4 of the original calibration procedure of Fig. 14.1. This time, however, the four-step hypothesis testing approach is performed on each individual partition created in Phase 2 to determine which partition can accept the lognormal distribution (PDLC) (and add it to the database) or not (in which case the partition is removed from further analysis). The details of

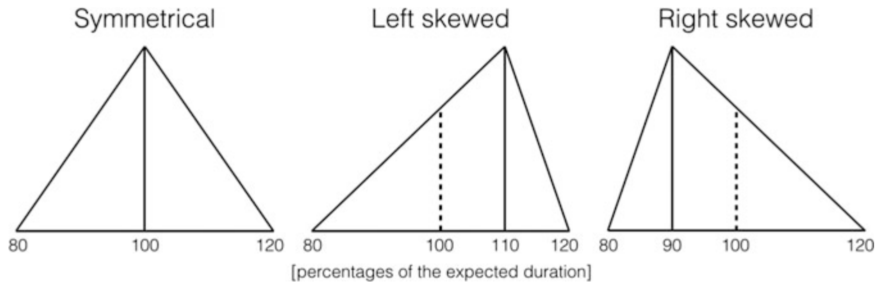


Fig. 14.4 Standard risk profiles used in the empirical database

the four calibration steps have been discussed previously and will not be repeated here, except for two minor differences:

- Improved removal of tardy points (Step 3): I have previously argued that in the original calibration procedure, a portion of $x\%$ of late activity is randomly selected for removal, and this selection can significantly affect the accuracy of the method. In the extended calibration method, Step 3 performs 1000 iterations randomly selecting a portion $x\%$ of late activities. The results are calculated for each iteration. After that, the average of the 1000 iterations for all outcomes (e.g., average p-value) is finally chosen to select the proportion of late activity to be removed. In this way, arbitrariness is circumvented, and the computational instability is thus solved. Moreover, 1000 iterations proved sufficient to make the variances in outcomes between different simulation runs negligible, while still maintaining an acceptable run time under the most complex procedure settings.
- Direct rounding (Step 4): In the original calibration procedure, the rounding step S4 always includes performing both S2 and S3 first. In the extended calibration, the rounding step can be performed without performing S2 and S3 first. This change is only a minor procedural extension to test whether steps S2 (delete on-time activities) and S3 (delete late activities) are really useful for calibrating the data, and will not be discussed further in this book.

14.4 Automatic Partitioning (*the horse*)

Given the promising results that we got with the human partitioning heuristic to treat project data as separate clusters rather than a single entity, we toyed with the idea of replacing the rider (human partitioning) with the horse (statistical partitioning) to find out which of the two has the most power for partitioning project data. In the aforementioned book "*The Master Algorithm*" by Domingos (2018), the author insinuated that the horse is perhaps the most powerful player, as he wrote:

> A frequently heard objection is 'data can't replace human intuition' but in fact, it's the other way around as 'human intuition can't replace data'.

Since we ourselves had no idea which of the two would yield the best results, we decided to develop a new calibration algorithm that extends the human partitioning method (left branch of Fig. 14.2) to a statistical partitioning method (right branch of the figure) and tested the new algorithm on more than 100 empirical projects with *and* without human intervention (published in Vanhoucke & Batselier, 2019b). This so-called *automatic partitioning heuristic* extends the previous calibration methods to a new extended partitioning process by integrating the hypothesis testing approach of the original calibration method with the human partitioning method and then extending it further with statistical components to automate almost everything. Consequently, much of the process follows a similar methodology to the previous calibration procedures, but the statistical extension results in a number of significant

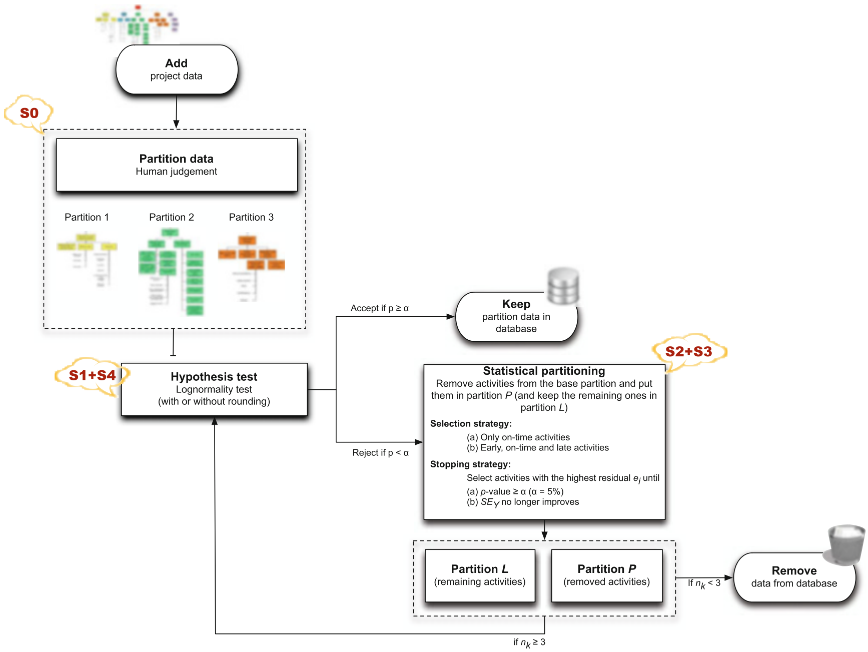


Fig. 14.5 Automatic partitioning (the *horse*)

changes, which are graphically summarised in Fig. 14.5. The figure consists of three separate phases, as discussed in the following paragraphs. I warn the readers that the following paragraphs contain quite a bit of detail, and should they not be easily digestible, I refer the readers directly to Sect. 14.5 where some conclusions are drawn.

Phase 1. Human Partitioning (*the rider*) The procedure starts with a human partition step that is identical to the initialisation step (S0) of the human partitioning method of Sect. 14.3. This first phase is optional and must be performed prior to statistical testing and statistical calibration.

Phase 2. Hypothesis Testing (*the original calibration method*) The *hypothesis test* of the statistical partitioning heuristic follows the same methodology as in the previously discussed calibration procedure of Sect. 14.1, and it incorporates the lognormal hypothesis test (S1) and the correction for rounding errors (S4). As mentioned earlier, the hypothesis test assesses whether or not $\ln(RD_{ij}/PD_{ij})$ is normally distributed by employing Blom scores and the table of Looney and Gulledge, while the correction for *rounding* errors (S4) corresponds to the averaging of the Blom scores for all clusters of tied points. Since no changes are made on these two steps, it is not necessary to elaborate on each aspect of the S0 and S4 procedures in detail again.

Recall that the hypothesis (S1) was also tested in steps S2 and S3 of the calibration procedure, after the removal of all on-time points and a portion of tardy points to incorporate the effect of Parkinson. As a matter of fact, the major difference between the previous calibration procedures and the new statistical partitioning method lies exactly in the treatment of the data for the Parkinson's effect (S2 or S3). The human partitioning method splits the project data into partitions (clusters) and then aims at removing data from the project partitions to be never used again (since it follows the Parkinson effect) and only continues the hypothesis testing on the remaining portion of the data. In contrast, the new statistical partitioning heuristic does not automatically remove data points from the clusters but aims at splitting each partition further into two smaller clusters (subpartitions) and then continues testing the same hypothesis on both subpartitions. This iterative process of splitting data and testing continues until a certain stop criterion is met and the data of all created subpartitions that pass the test are kept in the database. More precisely, each subpartition will be either accepted (i.e., the data follow a lognormal distribution) or rejected (i.e., the data do not follow a lognormal distribution or the sample size of the cluster has become too small) at a certain moment during the search. The way partitions are split into subpartitions is defined by two newly developed statistical strategies (referred to as the selection and stopping strategies), which will be discussed in the next phase.

Phase 3. Statistical Partitioning (*the horse*) The statistical partitioning heuristic performs the steps S2 and S3 slightly different than the original calibration method. It iteratively creates clusters of data with similar characteristics ((sub)partitions) based on statistical testing, similar to the human partitioning approach that aims at creating data clusters based on human input. More precisely, the statistical partitioning heuristic iteratively selects data points from a current partition and splits them into two separate clusters, and this process is repeated for each created cluster until a created subpartition can be accepted for lognormality. The approach to split these partitions into subpartitions does now no longer require human input but will be done using two new statistical strategies as discussed along the following paragraphs.

The so-called *selection strategy* defines the points of the current partition that should be selected for removal when splitting a partition. Each removed point will then be put in a first newly created subpartition, while the remaining non-removed points are put in a second new partition, now with less points than in the original partition. This process of removing data points from the original partition continues until a certain stopping criterion is met as defined by the so-called *stopping strategy*. Once the process stops, the original partition—which we will refer to as the *base partition*—has been split into two separate subpartitions that both will be subject to the hypothesis test again and—if still not accepted—will be further split into subpartitions. In the remainder of this manuscript, the term *partition L* will be used to indicate the subpartition with the set of activities that have not been removed from the base partition, while the set of activities that were eliminated from the partition and put in a newly created subpartition is now referred to as *partition P*.

It should be noted that the naming of the two partitions P and L found its roots in the testing approach of the previously discussed calibration procedures. Recall that steps S2 and S3 remove all on-time points and a portion of the tardy point from a partition. These removed points are assumed to be subject of the Parkinson effect (hence, partition P) and are thus removed from the database. The remaining data points in the partition were subject to further testing for the lognormal distribution (hence, partition L) and—if accepted—are kept in the database. A similar logic is followed for the statistical partitioning heuristic, although the treatment of the two partitions P and L now depends on the selection and stopping strategies that will be discussed hereafter.

Both the selection strategy and the stopping strategy can be performed under two different settings (*standard* or *advanced*), which results in $2 \times 2 = 4$ different ways for performing the statistical partitioning heuristic. Of course, these two strategies cannot work in isolation but will nevertheless be explained separately in the following paragraphs. A summary is given in Fig. 14.6.

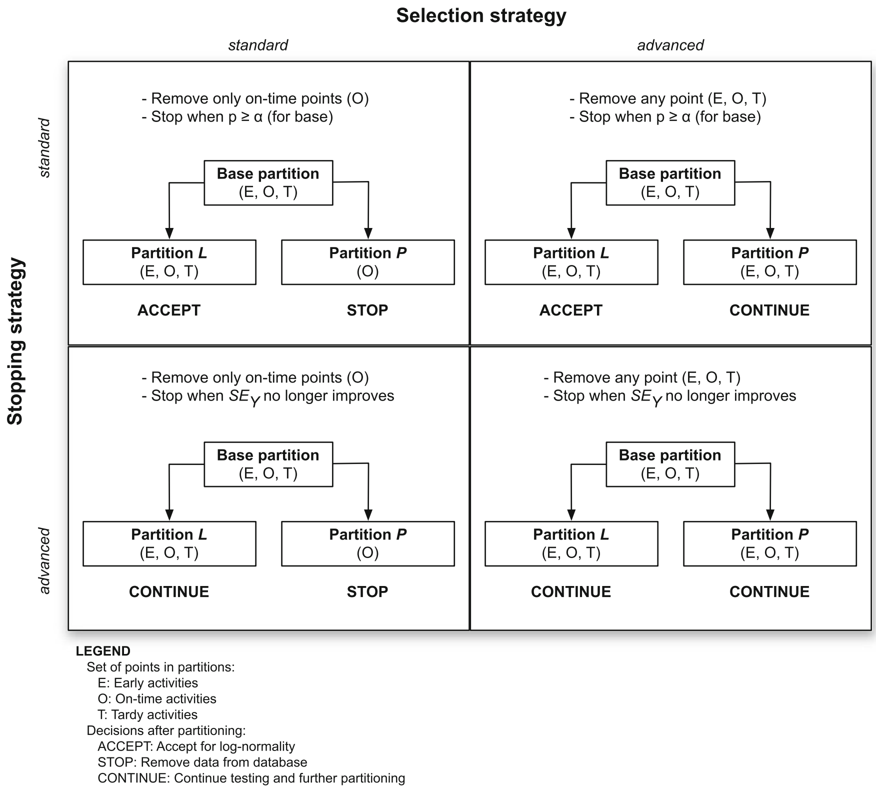


Fig. 14.6 The selection and stopping criteria of automatic partitioning

Selection Strategy
Recall that the partitioning heuristic splits up a partition into two new subpartitions. Partition P contains all the points that are removed from the base partition, while partition L contains all the non-removed points (but now contains less data points compared to the base partition). The *selection strategy* defines which points will be removed from the base partition to be put in partition P, and which points will be kept to create partition L. This selection can be done in a standard or an advanced way.

The *standard selection strategy* does not differ very much from the original calibration method and defines that only on-time points can be eliminated from the base partition. As a result, partition P with the removed activities will then obviously exhibit a pure Parkinson distribution (since all points are on time) and no further statistical partitioning will be performed for partition P. Partition L can still consist of early, on-time, and tardy points and will be further used by the partitioning heuristic. As shown in Fig. 14.6, no further partitioning will be performed for partition P, and its data are therefore thrown away (cf. STOP in Fig. 14.6). However, the specific treatment of partition L (ACCEPT or CONTINUE) depends on the setting of the stopping strategy, which will be discussed later when I explain the stopping strategy.

In the *advanced selection strategy*, *all* activities (not only on-time activities) are potential candidates to be selected for removal, and thus both the resulting partitions L and P can now contain early, on-time, and tardy points. This approach is called advanced since it is fundamentally different from the approach taken by the calibration procedures (S2 and S3). The most important implication of the advanced setting is that partitions in which not all activities are on time can now be created *automatically*. Indeed, the base partition will be split by eliminating activities from it, put them in partition P, and keep the remaining activities in partition L until L attains (optimal) fit (this optimal fit will be defined by the stopping strategy discussed in the next section). The set of removed activities (partition P), however, can now contain both on-time and early/tardy activities (just as partition L) and will thus most likely not exhibit a trivial pure Parkinson distribution (as was the case for the on-time activities of partition P under the standard selection strategy). Therefore, this partition P of removed activities should also undergo a hypothesis test and possibly a partitioning phase and so should all later partitions that are created as a result of this consecutive application of the partitioning heuristic. In that way, there is an automatic creation of partitions—hence the name statistical *partitioning* heuristic for the method—that should comprise activities that are similar to each other. Unlike the initial human partitioning method, no human judgement has interfered with this type of partitioning, which is the reason that it is called *statistical* partitioning.

While the set of activities to be removed from the base partition differs between the standard (only on-time points) and advanced (all points) selection strategies, the partitioning heuristic still needs to determine the sequence in which these activities are removed until a stopping criterion is met. In contrast to the calibration procedures, the statistical partitioning heuristic needs to select the activity to be eliminated in every partitioning step. The term *partitioning step* is used for an iteration of the partitioning heuristic in which one activity is removed. So if there were 10 partitioning steps for a particular project or partition (under certain settings), then 10 activities were eliminated from that project or partition. For this purpose, the procedure calculates the residuals for all activities in the base partition. The residuals e_i are calculated as the deviations between the empirical values $\ln(RD_i/PD_i)$ and the linear regression line of those values on the corresponding Blom scores. As a heuristic approach—hence the name statistical partitioning *heuristic*—the activity i with the biggest residual e_i in the base partition is selected for elimination (and put in partition P) as it is expected that this would yield the strongest improvement in the goodness-of -fit (since the created partitions will be subject to a new hypothesis test again).

Stopping Strategy

The selection strategy defines how the base partition is split into two different partitions by iteratively removing data points (activities) to create partitions L and P. Despite the fact that this selection mechanism controls the sequence of points to be removed using the calculation of the residuals, it does not define any stopping criterion during this iterative removal process. To that purpose, the statistical partitioning heuristic also introduces two different versions for the stopping strategy. When the stopping criteria are satisfied, the removal of activities is stopped, and the resulting partitions (L and P) are then subject to a new partitioning iteration (by going back to S1).

The *standard stopping strategy* employs the p-value to define the stopping criterion. More specifically, the elimination of activities stops when p reaches or exceeds the significance threshold $\alpha = 0.05$ for partition L. Since the p-value is also the condition for accepting the lognormality hypothesis in S1, this implies that the lognormality test is automatically accepted for this partition L and all its activities are assumed to follow the lognormal distribution. In this case, no further partitioning is necessary for partition L and all its data points are added to the database (cf. ACCEPT in Fig. 14.6). The data points in partition P are treated differently, and the treatment depends on the option in the selection strategy. Since the partitioning heuristic is always applied anew to the newly created partitions, every partition P that is created should go back to S1 and should be tested for lognormality if the advanced selection strategy is chosen. Under the standard selection strategy, however, partition P only contains on-time points, and these points will obviously exhibit a pure Parkinson distribution. In this case, no further statistical partitioning will be performed and the data points are removed from the project (cf. STOP in Fig. 14.6).

In the *advanced stopping strategy*, the statistical partitioning is no longer limited to the use of the p-value as the only measure for goodness-of-fit, but the activity removal halts when SE_Y (or R_a^2 as a secondary stopping criterion) does no longer improve. Indeed, it uses the standard error of the regression SE_Y as the main basis for assessing the fit since SE_Y is the preferred measure according to the literature. The formula for SE_Y is given below.

$$SE_Y = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n-2}}. \tag{14.1}$$

The denominator is the number of activities in the partition (n) minus two since there are two coefficients that need to be estimated in our case, namely the intercept and the slope of the regression line. SE_Y is also chosen as the primary optimisation criterion, which means that the fit to the PDLC is improved when the removal of the selected activity has decreased the SE_Y. Obviously, the lower the SE_Y, the better the fit. A perfect fit is obtained when all data points are on the regression line, as all residuals are per definition zero, which implies that SE_Y is also zero in Eq. (14.1) . However, in about 20% of the cases, the partitioning heuristic did not reach the optimal SE_y when *only* that SE_y was considered as optimisation criterion; it got stuck in a local optimum. To get out of this local optimum, we added the adjusted R^2 or R_a^2 as a secondary stopping criterion, which—although a very straightforward approach—proved a highly effective solution to the problem. Indeed, after adding R_a^2 as a secondary optimisation criterion, only 1% of the projects did not attain their optimal SE_y. For completeness, we mention the formula for R_a^2 with respect to the standard coefficient of determination.

$$R_a^2 = 1 - \frac{n-1}{n-2}(1 - R^2). \tag{14.2}$$

Notice that, unless $R^2 = 1$, R_a^2 is always smaller than R^2. In our context, we need to employ R_a^2 instead of R^2 to allow comparison of regression models with different numbers of observations (activities indeed get removed from the original dataset). Just like for the p-value, the higher the R_a^2, the better the fit, with a maximum of 1 to reflect a perfect fit.

As mentioned before, the two settings for the stopping strategy should be used in combination with the two settings for the selection strategy, and it is important to draw the attention to the two fundamental differences with the original and human calibration procedures. First, the treatment of the Parkinson points is fundamentally different. Recall that *all* on-time points are removed in the calibration procedures since they are assumed to be the result of the Parkinson effect. In the standard selection strategy, the procedure also removes on-time points, but it is no longer true that the only possibility is to remove *all* on-time points from the project. The statistical partitioning heuristic allows the elimination of just a fraction of the on-time points in order to get a better fit (defined by the stopping strategy, i.e., p-value or SE_Y). The rationale is that not all on-time points are necessarily the result of the Parkinson effect, as the calibration procedures implicitly assume. Some activities *are* actually on time and should thus effectively be part of partition L. Secondly, not only on-time points are removed, but also early and tardy points are now subject to removal. While the calibration procedures only remove a portion of tardy points to bring the number early, on-time and tardy points back to the original proportions, the statistical partitioning heuristic takes a different approach and removes early, on-time as well as tardy points (under the advanced selection strategy) until the stopping criterion is satisfied. Such an approach creates partitions (L and P) that contain all kinds of activities (early, on time, and tardy) that must be subject to further partitioning, if necessary, which is fundamentally different from the approach taken by the calibration procedures.

14.5 Calibration Results

The development of these different calibration methods was *a hell of a ride* as they were somewhat outside my traditional research comfort zone. Testing these methods was a trial-and-error process with many failed experiments and endless discussions with various people from academia *and* practice. I mentioned earlier that the original idea of calibrating data came from a paper by Trietsch et al. (2012), so we were lucky enough to use the original idea as a basis for incorporating human and statistical partitioning into the method. Each extension was therefore extensively tested on a growing body of empirical project data and ultimately resulted in a more versatile, understandable, and practically applicable method that partially overcomes the limitations of the original calibration procedure. It eventually resulted in the statistical calibration heuristic of Sect. 14.4 that is a combination of hypothesis testing, human expertise, and statistical partitioning. The main results of this trial-and-error process are summarised in the following paragraphs.

Original Calibration Method When I came across the idea of calibrating project data, I only had 24 projects available in my empirical project database. I nevertheless decided to validate the calibration method on this relatively small sample of data, which led to the study of Colin & Vanhoucke (2016). The results showed that only one project passed the lognormality test after S1 of Fig. 14.1, and 11 projects after S2. However, 13 projects had passed the lognormality test by the end of S4. When S3 expanded to the 1000 iterations (instead of one iteration) as mentioned earlier, this grew to 18 projects. This indicates the advantage of simple technical extensions as the previously discussed multi-iteration approach in S3 significantly improves the performance over the single simulation run of the original procedure. In addition, the results also indicated that the Parkinson effect (S3) is significantly larger than the rounding effect (S4) when it comes to lognormal distribution fitting. All in all, the results were in my eyes impressive and well worth investigating further.

Human Partitioning ***(The rider)*** The extension of the human partitioning approach was tested in Vanhoucke and Batselier (2019a) on a significantly larger database of 83 projects (instead of 24 in the original study). We started the analysis with 125 projects, but 28 projects did not contain authentic time registration data and so the true duration (RD) was not known. These projects were removed from the analysis (97 left). In addition, because 14 projects only included activities that ended exactly on time (which are believed to be subject to the Parkinson effect), these were also excluded from the analysis, ultimately using the 83 projects for further experimentation. The total number of activities of these projects amounted to no less than 5068 activities (or an average of 61 activities per project), which can be seen as a large database for partitioning.

The experiments showed that adding human expertise (for partitioning) can significantly improve the calibration method and thus improve the acceptance rate of partitions. The experiments showed that the WP (*work package*) and RP (*risk profile*) are the superior partitioning criteria compared to PD (*planned duration*). This indicates that it is indeed relevant and valuable for project managers to define WPs and RPs for their projects as these can form the basis for a more realistic division of activities. And in turn, more realistic partitions—and their corresponding and specific distribution profiles—can lead to better risk assessments of activities, more tailor-made, and targeted project control methods (e.g., by focusing only on the most risky partitions), and more accurate project forecasts (based on, for example, Monte Carlo simulations). However, there is an essential requirement that the resulting partitions are realistic, which means that the project manager must correctly allocate the activities to the correct WPs and define the correct RPs for them. For the projects that we have used, the project managers have clearly succeeded in this task. But if they fail—intentionally or unintentionally—then the resulting partitions are not reliable, and it may be better to use the more certain PD as the partition criterion rather than the flawed WP and RP. It is clear that these results only illustrate that human partitioning adds value and that human expertise can significantly improve the calibration method, but more research is needed to find the best possible criteria for partitioning. Nevertheless, the experiments resulted in an impressive 97% acceptance rate of the partitions created, which is a clear indication that human expertise can add value to the calibration of project data.

Automatic Partitioning Without Human Intervention *(The horse)* The automatic partitioning heuristic was tested on the same 83 projects, but creating partitions was now left entirely to the statistical horse, rather than the human rider, as discussed in Sect. 14.4. While automatic partitioning has the advantage of eliminating any possible human bias, it carries the risk of creating too many partitions. In the worst case, each activity is placed in a single partition, rendering the partition heuristic completely useless. For this reason, the algorithm requires each partition to have at least 3 activities (otherwise, it will be removed from the database). Fortunately, our computational experiments have shown that the best fit can be obtained with a maximum number of partitions equal to five per project, which is not too much to become clumsy to work with. In addition, the results also showed the positive effect of rounding correction (S4), resulting in fewer partitions and a better fit (higher p-value). Finally, the positive impact of the selection strategy (resulting in an average of 2.6 partitions per project) and the stop strategy (resulting in the best goodness-of-fit) was a clear indication that automatic partitioning is a worthy calibration extension.

Human and Automatic Partitioning ***(The rider and the horse)*** Our ultimate goal of the various studies was, of course, to show that the rider and horse should work in harmony, creating a relatively small number of partitions with a high acceptance rate. Our experiments revealed promising results and indeed indicated that both human and statistical partitioning work well, but their integration should be used with caution. In fact, the experiments showed that the so-called *double partitioning effect* can sometimes lead to too many partitions with too few activities. Indeed, the human partitioning method creates partitions based on PD, RP, and/or WP, and each of the partitions obtained from human partitions can be further divided into smaller partitions—therefore called *subpartitions*—using the statistical partitioning heuristic. This means that each project actually goes through two successive partitioning phases, which can lead to too many partitions in a project. We saw that when the WP criterion is used for human partitioning, the number of partitions even reaches 631 across 53 projects, which amounts to almost 12 subpartitions per project. This is perhaps a bit much to be practical and less relevant as this implies an average of only 6 activities per subpartition. However, this was not a problem when one of the other management criteria (PD or RP) was applied, with an average of about 5 subpartitions per project. The main reason was that project managers apparently define far too many work packages, an average of 8 per project, with an exorbitant maximum of 26 WPs for one project. This problem could be solved by encouraging project managers to limit the number of WPs identified by considering higher-level classification criteria. Despite this potential problem, the results nevertheless showed that the absolute best fit can be obtained when the advanced selection and advanced stopping strategies are used in combination with human risk profile (RP) partitioning. The mean p-value of 0.811 was significantly higher than the maximum of all other calibration procedures tested under different settings. In addition, the percentage of accepted partitions was again very high (97%), and so we can conclude that the partitioning heuristic outperforms the other calibration procedures. This indicates that the application of human partitioning criteria is indeed relevant and useful, and thus their definition should be encouraged by project managers.

It is worth mentioning that the obtained p-value of 0.811 was not exceedingly higher than that for the partitioning method under the basic selection/stopping strategies, combined with either of the other managerial criteria (p ranging from 0.756 to 0.783) or even without human partitioning ($p = 0.731$). The reason for this is that a combination of human and statistical partitioning should in fact be seen as a *double optimisation*. Both partitioning approaches already perform very well separately, but combining them takes the distribution fitting another (small) step closer to "optimal" partitioning. Furthermore, human and statistical partitioning do not only perform well on their own, but they are mutually also quite comparable. This observation is in fact hugely promising as it indicates that we can just perform the partitioning heuristic with inclusion of the statistical partitioning and still obtain very relevant partitions without requiring realistic input for managerial criteria (i.e., WPs or—even better—RPs accurately defined by the project manager). Statistical partitioning is no longer—or at least far less—prone to human judgement and bias than human partitioning. In the latter case, project managers indeed need to *accurately* define the WPs or RPs. Otherwise, the resulting partitions would be faulty and unrealistic anyhow. It might be beneficial to bypass this uncertain human factor and thus create a more solid and trustworthy methodology for categorising activities into risk classes and assigning specific distribution profiles to them. Consequently, the results seem to reveal that statistical partitioning is almost able to “replace” human partitioning, which reminds us of Domingos’ quote that *human intuition cannot replace data*. The experiments of the automatic partitioning method are published in Vanhoucke & Batselier (2019b).

14.6 Conclusion

This chapter told the story of empirical data analysis to construct statistical distributions that can be used in academic studies to perform simulations. It is a story that uses both statistical techniques and human experience to make these distributions realistic. It is also a story that only just started and I think that there is a lot of opportunity to expand this story further. In fact, I think that the use of calibration methods is still in its infancy, and I hope that the studies in this chapter can provide an impetus for further research. Current methods only take the Parkinson effect and rounding errors into account as human biases, and I think that there are a lot of other factors that can distort real data. I therefore call on researchers to further expand these methods and to include new confounding factors in the research so that the calibration methods become even more realistic (and therefore more useful). However, I would like to warn the researcher that getting the studies accepted was not an easy task. Many referees had a lot of comments on these studies, and despite the fact that many of these comments were included (sufficiently enough to allow the studies to be published), there were also a number of comments to which I could not formulate a proper answer. Apparently, innovative methods such as this are not very easy to understand, and reviewers comment a lot because—by definition—these calibration procedures still have a lot of gaps.

Despite the opposition that I sometimes encountered, I also see many people (both from academia and practice) who believe in the practicality of this calibration. After all, I think these methods can build bridges between academia and the professional world. If project managers want to use the simulation techniques described in the academic studies of Part II, they can now rely on distributions with realistic parameter values, and so there are not many reasons why they should not do it. Of course, every technique remains subject to the *garbage-in, garbage-out* phenomenon, and a wrong calibration can lead to wrong conclusions. In my opinion, however, this potential danger is drastically reduced when the distributions are calibrated for empirical data rather than when they are completely based on the imagination of the researcher. I therefore see this as an important task for the researcher to consider using empirical projects for calibration first, and to switch to artificial project data only after calibration (when the distributions are known). I know I have argued the opposite in previous chapters (artificial projects first, then empirical validation projects), but I am not ashamed to change my mind if I think the time is urgent. In particular, I think that both types of projects can be useful to improve the general knowledge of project management and control, and a close collaboration between the researcher (*artificial data*) and the practitioner (*empirical data*) will most likely lead to the most interesting and inspiring research results.

In this and previous chapters, I have told the story of project data in great detail, and I am not even at the end (but almost). In the next chapter, I will tell one last closing story, without going into great detail like in the previous chapters. Chapter 15 contains an overview of artificial and empirical project data developed at the OR&S group. Much of these data have been covered extensively in previous chapters, but some are new and presented for extensions of the well-known *resource-constrained project scheduling problem*. I will give only a few details about these extensions (as they are not within the scope of this book), but I chose to include this summary anyway since I know you are interested in project data (otherwise you would have never made it to the end of this chapter). After the next chapter, I will close Part IV about project data, even though I realise very well that this story will never be fully told. I am nevertheless happy and proud that I could contribute, and I hope that it can lead to more and better research in data-driven project management.

I cannot close this chapter without a final quote on the use and relevance of data, but with so many insightful quotes out there, I found it hard to pick just one. Finally, I decided to use a quote from Dan Heath, best-selling author and fellow at Duke University, who, like me, believes that using data is a never-ending story. After all, he wrote down the following quote (although I cannot remember where I ever found it):

> Data are just summaries of thousands of stories.
> Tell a few of those stories to help make the data meaningful.

I hope my stories have given you some meaningful insights.

References

Blom, G. (1958). *Statistical estimates and transformed beta-variables*. Wiley.

Colin, J., & Vanhoucke, M. (2016). Empirical perspective on activity durations for project management simulation studies. *Journal of Construction Engineering and Management, 142*(1), 04015047.

Domingos, P. (2018). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. New York: Basic Books.

Looney, S. W., & Gulledge, T. R., Jr., (1985). Use of the correlation coefficient with normal probability plots. *The American Statistician, 39*(1), 75–79.

Parkinson, C. N. (1957). *Parkinson's law and other studies in administration*. Houghton Mifflin.
Trietsch, D., Mazmanyan, L., Govergyan, L., & Baker, K. R. (2012). Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation. *European Journal of Operational Research, 216*, 386–396.
Vanhoucke, M., & Batselier, J. (2019a). Fitting activity distributions using human partitioning and statistical calibration. *Computers and Industrial Engineering, 129*, 126–135.
Vanhoucke, M., & Batselier, J. (2019b). A statistical method for estimating activity uncertainty parameters to improve project forecasting. *Entropy, 21*, 952.

Chapter 15
More Data

The widespread availability of project data for researchers is good to help them develop better tools and more powerful algorithms to improve the state-of-the-art solutions in scheduling and project control. In the previous chapters, I gave a full overview of the many research studies done by my Operations Research and Scheduling (OR&S) research group for the generation of artificial project data and the collection of empirical project data. This chapter gives a concise summary of the different datasets available for research, with a special focus on the datasets to solve the *resource-constrained project scheduling problem* and many of its extensions. Some of the datasets have been discussed in earlier chapters as summarised in Fig. 11.2. Other datasets are new and pay special attention to new research lines for this challenging project scheduling domain. Figure 15.1 displays the different datasets with references to the journals and is an update of Fig. 11.2. It serves as a guideline to the content of this chapter, and a summary of detailed features for each dataset is given in Appendix H. This chapter should be read as a stand-alone article to get a quick full overview of the sources of data available at our OR&S website. You can simply download them and start your own research journey. Just visit the following website:

www.projectmanagement.ugent.be/research/data

15.1 Resources

The *resource-constrained project scheduling problem* has been discussed previously in Chap. 6 on machine learning and Chap. 11 on artificial project data.

Researchers in the field of project management and scheduling develop tools and techniques to schedule projects under limited resources. The most well-known problem in the project scheduling literature is the **resource-constrained project scheduling problem** (RCPSP) for which many different datasets have been

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_15

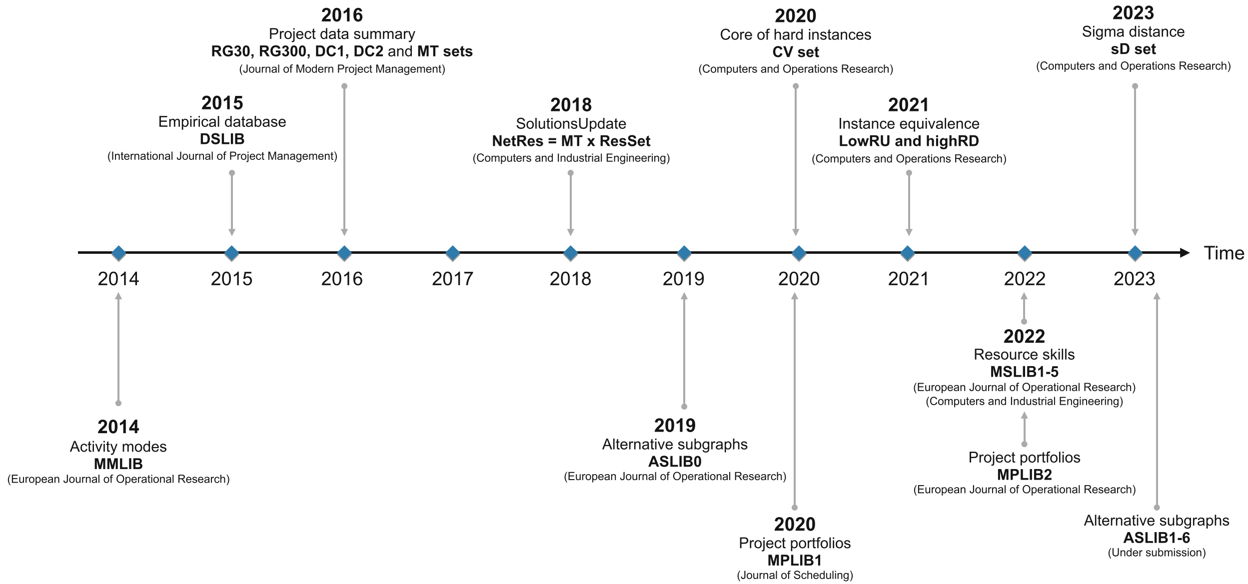


Fig. 15.1 Overview of research on project data (Part 2)

presented in the literature. The Operations Research and Scheduling (OR&S) group has also presented a project data generator to generate project data under a strict predefined design, and this generator is known as RanGen (Versions 1 and 2). The network generator is proposed in the following study:

> Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. V. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. European Journal of Operational Research, 187(2), 511–524. (Vanhoucke et al., 2008)

OR&S was, of course, not the only research group to present network generators and new project datasets to the academic community. Therefore, in 2016, we decided to write a summary of the existing datasets that can be found in the following publication (ever since this 2016 publication, new datasets have been proposed, so this chapter does not contain all existing sets). In this publication, five datasets are made by OR&S known as the **RG30**, **RG300**, **DC1**, **DC2**, and **MT** datasets:

> Vanhoucke, M., Coelho, J., & Batselier, J. (2016). An overview of project data for integrated project management and control. Journal of Modern Project Management, 3(2), 6–21. (Vanhoucke et al., 2016)

Although the previous datasets contained thousands of artificial projects, researchers often still do not understand why scheduling algorithms sometimes perform very well on some project instances and fail miserably on other project instances. Therefore, our most recent research aims at better understanding the complexity of project data. To that purpose, we have generated a dataset of 623 project instances that are—at the time of generation—impossible to solve to optimality with the currently known state-of-the-art procedures. We called this set the **CV** dataset, which is an abbreviation of the authors' names (C = Coelho and V = Vanhoucke) of the paper that proposed this new set to the literature:

> Coelho, J., & Vanhoucke, M. (2020). Going to the core of hard resource-constrained project scheduling instances. Computers and Operations Research, 121, 104976. (Coelho & Vanhoucke, 2020)

While the CV set mainly focuses on solving the project instances to optimality, nothing is said about the complexity of these project instances for meta-heuristic solution procedures. Since these procedures work in a totally different way, aiming at providing near-optimal solutions, a new study has been set up that introduced a new concept called sigma distance. This concept is discussed earlier in this book and measures how easy the project instance can be solved to (near-)optimality by random sampling. The resulting dataset is called the **sD** dataset containing 390 instances and is proposed in:

> Coelho, J., & Vanhoucke, M. (2020). New resource-constrained project scheduling instances for testing (meta-)heuristic scheduling algorithms. Computers and Operations Research, 153, 106165. (Coelho & Vanhoucke, 2023)

Finally, for 10,793 instances coming from different sets, duplicate instances were made by changing their resource data, as proposed in Sect. 11.8. Each of the 10,793 instances has two new instances, referred to as the **lowRU** and **highRD** sets, which are instance equivalent (meaning that their set of solutions is the same as for the original instances). This instance equivalence concept is proposed in:

> Vanhoucke, M., & Coelho, J. (2021). An analysis of network and resource indicators for resource-constrained project scheduling problem instances. Computers and Operations Research, 132, 105260. (Vanhoucke & Coelho, 2021)

Given the previous summary with references, it should be clear by now that an overwhelming amount of artificial project data are available for the resource-constrained project scheduling problem. Each set serves a different purpose, and researchers are invited to freely download and use the data for testing new and exciting project scheduling algorithms! Good luck!

15.2 Modes

A well-known extension of the classic RCPSP is the **multi-mode resource-constrained project scheduling problem** (MRCPSP) in which each activity can be scheduled in different modes. For this challenging scheduling problem, various datasets have been presented in the literature. The OR&S group has proposed three different datasets for the MRCPSP problem, known as the multi-mode libraries **MMLIB50**, **MMLIB100**, and **MMLIB+**, in the following publication:

> Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of meta-heuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. European Journal of Operational Research, 235(1), 62–72. (Van Peteghem and Vanhoucke, 2014)

15.3 Subgraphs

A recent extension of the classic RCPSP is the **resource-constrained project scheduling problem with alternative subgraphs** (RCPSP-AS) in which the network contains various alternative structures, and only one needs to be selected. Once these so-called alternative subgraphs are selected, the scheduling problem boils down to a classic RCPSP. Two new datasets **ASLIB0** are proposed in:

> Servranckx, T., & Vanhoucke, M. (2019). A tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs. European Journal of Operational Research, 273(3), 841–860. (Servranckx and Vanhoucke, 2019)

The project data of the **ASLIB0** dataset contain nested and linked subgraphs in the problem. However, some years later, we extended the RCPSP-AS with richer features (including multiple and split choices and caused and closed subgraphs) and therefore needed new extended data to test algorithms for these extended problems. To that purpose, we decided to generate five new datasets that can possibly replace the initial **ASLIB0** datasets, and we labelled them as **ASLIB1** to **ASLIB5**. A summary of the data is presented in:

> Servranckx, T., & Vanhoucke, M. (2023). New datasets for the resource-constrained project scheduling problem with alternative subgraphs. Working paper (under submission). (Servranckx and Vanhoucke, 2023)

Finally, a set of empirical project data has been proposed (**ASLIB6**) in an unpublished study. The dataset is available on our website, and we plan to write a paper about these empirical projects in the near future.

15.4 Skills

A recent extension of the classic RCPSP is the **multi-skilled resource-constrained project scheduling problem** (MSRCPSP) in which activities require skills to be executed and the resources, i.e., the humans, possess a set of skills. The problem has been investigated in the literature under different settings, and four new datasets **MSLIB1** to **MSLIB4** with artificial projects to test existing and new procedures are proposed. Furthermore, a fifth dataset **MSLIB5** with empirical projects is also included. These 5 datasets are proposed in:

> Snauwaert, J., & Vanhoucke, M. (2022). A classification and new benchmark instances for the multi-skilled resource-constrained project scheduling problem. European Journal of Operational Research, 307(1),1–19. (Snauwaert and Vanhoucke, 2023)

In a follow-up study, the MSLIB datasets are extended in two different ways. First, the first four MSLIB sets are extended to hierarchical skill levels to model resource-constrained projects with skills in various ways. Furthermore, these MMLIB sets have also been converted into the SSLIB set (**SSLIB1** to **SSLIB4**), which contains projects for the so-called software scheduling problem useful for agile project planning with human resources. These extensions are discussed in:

> Snauwaert, J., & Vanhoucke, M. (2022). Mathematical formulations for project scheduling problems with categorical and hierarchical skills. Computers and Industrial Engineering, 169, 108147. (Snauwaert and Vanhoucke, 2022)

15.5 Reality

All the previously discussed project data are generated for researchers to test new algorithms for project planning, but until 2015, little was done to create an empirical dataset with real projects. In 2015, OR&S decided to collect and publish a set of real projects, initially consisting of 52 projects (but now (in 2023) contains already 181 projects). These real project data can be used—just like the artificial data—for **project scheduling**, but since the empirical data are much richer than the artificial data, it can also be used for a **schedule risk analysis** and even for **project control** using Earned Value Management. Since these three components are the foundation for the **dynamic scheduling** (DS) framework, the library is referred to as **DSLIB**. A summary of the 52 projects in the empirical database is given in the following paper (the full database of all projects can be downloaded from our website):

Batselier, J., & Vanhoucke, M. (2015). Construction and evaluation framework for a real-life project database. International Journal of Project Management, 33(3), 697–710. (Batselier & Vanhoucke, 2015a)

As mentioned earlier, two other empirical project databases are available. The first database contains empirical projects for the multi-skilled problem and has been proposed earlier as the **MSLIB5** set. The second set contains empirical projects with flexible network structures and has been proposed earlier as the **ASLIB6** set. When researchers want to use both artificial ànd empirical project data, the OR&S advice is as follows: First test your algorithms on the artificial project data, and then validate your results on the empirical project data.

15.6 Portfolio

Planning and scheduling single projects assume that each project can be scheduled individually, and resources are fully dedicated to one project. While this might be true for very big projects, this is, of course, not always the case. Instead, most companies have a portfolio of projects where resources can be shared between them. In a portfolio of projects, shared resources have an impact on each project and a portfolio of projects is more than merely the sum of individual projects. Therefore, OR&S has generated project portfolio data for the so-called **resource-constrained multi-project scheduling problem** (RCMPSP), which consists of more than merely an assembly of the single-project data. Instead, the generation process to construct the multi-project data libraries is fundamentally different from the generation mechanism used for single-project data libraries. The project portfolio data are available from our website, and the generation process is discussed in two papers. In a first paper, a first multi-project library (**MPLIB1**) is presented and compared with the existing project portfolio data from the literature.

Van Eynde, R., & Vanhoucke, M. (2020). Resource-constrained multi-project scheduling: Benchmark datasets and decoupled scheduling. Journal of Scheduling, 23, 301–325. (Van Eynde and Vanhoucke, 2020)

In a follow-up paper, a second multi-project library (**MPLIB2**) is presented that now better reflects the true characteristics of project portfolios.

Van Eynde, R., & Vanhoucke, M. (2022). New summary measures and datasets for the multi-project scheduling problem. European Journal of Operational Research, 299(3), 853–868. (Van Eynde and Vanhoucke, 2022)

References

Batselier, J., & Vanhoucke, M. (2015). Construction and evaluation framework for a real-life project database. *International Journal of Project Management, 33*, 697–710.

Coelho, J., & Vanhoucke, M. (2020). Going to the core of hard resource-constrained project scheduling instances. *Computers and Operations Research, 121*, 104976.

Coelho, J., & Vanhoucke, M. (2023). New resource-constrained project scheduling instances for testing (meta-)heuristic scheduling algorithms. *Computers and Operations Research, 153*, 106165.

Servranckx, T., & Vanhoucke, M. (2019). A tabu search procedure for the resource-constrained project scheduling problem with alternative subgraphs. *European Journal of Operational Research, 273*, 841–860.

Servranckx, T., & Vanhoucke, M. (2023). New datasets for the resource-constrained project scheduling problem with alternative subgraphs. In *Working paper (under submission)*.

Snauwaert, J., & Vanhoucke, M. (2022). Mathematical formulations for project scheduling problems with categorical and hierarchical skills. *Computers and Industrial Engineering, 169*, 108147.

Snauwaert, J., & Vanhoucke, M. (2023). A classification and new benchmark instances for the multi-skilled resource-constrained project scheduling problem. *European Journal of Operational Research, 307*, 1–19.

Van Eynde, R., & Vanhoucke, M. (2020). Resource-constrained multi-project scheduling: Benchmark datasets and decoupled scheduling. *Journal of Scheduling, 23*, 301–325.

Van Eynde, R., & Vanhoucke, M. (2022). New summary measures and datasets for the multi-project scheduling. *European Journal of Operational Research, 299*, 853–868.

Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research, 235*, 62–72.

Vanhoucke, M., & Coelho, J. (2021). An analysis of network and resource indicators for resource-constrained project scheduling problem instances. *Computers and Operations Research, 132*, 105260.

Vanhoucke, M., Coelho, J., & Batselier, J. (2016). An overview of project data for integrated project management and control. *Journal of Modern Project Management, 3*(2), 6–21.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research, 187*, 511–524.

Part V
Afterword

Fantasy is way more important than knowledge.

Chapter 16
The Perfect Researcher

You have now finished this book and I hope it has given you a good summary of what my research team has been doing for the data-driven project management discipline over the past few years. I loved writing a book about my own research, not only because it forced me to think about my own work from a different perspective but also because it allowed me to look back and see what a wonderful job and team I had (and still have). It was not all fun, of course. Writing such a book was a challenge with many periods of doubt whether I should continue or not. It all started with writing down fragmentary pieces from the past (that is the fun part) but soon needed a common theme in order to tell a story to an audience (and that is the part with doubts and continuous changes). After deliberation, I am nevertheless happy that I have continued writing because I am satisfied not only with the end result (the book you are holding in your hands) but also with the ideas that came to me for future research based on the shortcomings and weaknesses that I encountered while writing. I have no doubt that more research is needed (and will come, I am working on it) and I feel privileged once again to continue this fantastic research journey as a never-ending quest for improvements.

And this brings me seamlessly to the two main topics of this book.

First of all, I sincerely hope that the readers have clearly seen that the research discussed in this book is the result of teamwork. In 1996, I could never have imagined that I would do such an exciting job with a team of young, inspiring, and enthusiastic people. After more than two decades of academic research, I now love my research work mainly because of that: working with young people, eager to find new results, not knowing exactly what we are looking for, yet learning from each other while exploring the unexplored terrain. I wrote this book especially for them, for all members of my team, not just those who I mentioned in the references but also those I did not mention (because their research topic was not in the scope of this book). I would like to thank them all for wanting to be part of this fantastic OR&S journey. For the readers who want to meet these young enthusiastic people, you can find them at www.projectmanagement.ugent.be, learn more about them in my free

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9_16

book "*The Art of Project Management*" (which I mentioned earlier in Chap. 2[1]) or find them in Appendix A.

Moreover, I also hope that this book may be an opportunity to attract young researchers in the wonderful domain of data-driven project management. I therefore want to invite new young people to start a career in academia. I hope that readers of this book—especially those who are not (yet) familiar with research—have clearly seen that research is a combination of hard work with not only a lot of failures but also a lot of fun in the search for the unknown. To anyone looking for a challenge, I highly recommend starting a career in scientific research. It is a fantastic job. You do not have to be super smart (thankfully). You do not have to be a computer expert, math genius, or someone with an extraordinary memory. All you need is an attitude of persistence and an endless passion to find something that you cannot even think of right now. Every year, I hire one or two new young graduates to start such an exciting career in academia, recruiting people from different universities, interviewing them, having them undergo psychological tests, and asking them challenging and sometimes unsolvable questions. Systems exist to guide you through such a time-consuming process, but they rarely lead to the perfect researcher. I usually start from my gut feeling (which sometimes disappoints me) and I am mainly looking for passion and persistence in a person. I have thought a lot about the characteristics of the perfect researcher, but after all these years I finally realise that such a person does not exist. I have learned that supervising young researchers is a process with ups and downs. Every person is different and every time it is a process that is guided by trial and error. And so I have learned, also through trial and error, that each researcher should be treated differently. Despite this uniqueness, I think that there are some typical qualities that a good researcher should have, and I will summarise them in the following paragraphs. These features are of course my own opinion, but they are all inspired by what I was reading while writing this book (novels, papers, and science books). That is because I really believe that one of the qualities you really need in order to do research is a love for reading books. I am not talking about books on project management and control, I am talking about books that interest you, go beyond your own research expertise, and stimulate your imagination. By the way, I rarely read books about my field of research, and I would not know why I would. I meet many researchers to talk about my field of research, I follow the literature closely, and I attend conferences regularly, so I think I am sufficiently informed. I like my field of research, but I am not going to spend my precious free time reading books about my work because then that little free time goes in the same direction again. I therefore advise every researcher not to read many books on project management (except the one you are reading now of course and the ones I discussed in Chap. 2 (I should put a smiley here)), but to get inspired by books from a different field, such as books on mathematics, philosophy, biology, space exploration, the hidden life of trees, and so much more. And so the forthcoming summary of the four important qualities of a good researcher is inspired

[1] The download link for this free book can be found at www.or-as.be/books/wp.

by the books that I was reading at the time of writing. I will refer to these books, but I must admit that this list is a snapshot and I am sure that I would have referred to other books if I had written this book a year later. However, I believe that the four typical characteristics of the perfect researcher would be the same. If you think this list of traits typifies you, then I think doing research is something you should definitely consider.

16.1 Doubt

One of the most important requirements for doing research is being able to doubt. In the second chapter "*Illusions of knowledge*" of the book "*Superforcasting*", Tetlock and Gardner (2015) write that *"doubt is not something terrifying, but something of great value."* The authors refer to the Nobel Prize-winning physicist Richard Feynman (1998) who said:

> When the scientist tells you he does not know the answer, he is an ignorant man. When he tells you he has a hunch about how it is going to work, he is uncertain about it. When he is pretty sure of how it is going to work, and he tells you, "This is the way it's going to work, I'll bet," he still is in some doubt. And it is of paramount importance, in order to make progress, that we recognise this ignorance and this doubt. Because we have the doubt, we then propose looking in new directions for new ideas. The rate of the development of science is not the rate at which you make observations alone but, much more important, the rate at which you create new things to test.

The authors of "*Superforecasting*" further state that good forecasting does not require powerful computers and sophisticated methodologies, but rather evidence from various sources, probabilistic thinking, working in teams, and a willingness to admit mistakes and change course. It seems that they want to say that *if you want to know something for sure, you have to collect data, analyse it, draw conclusions and above all never stop doubting* and that is exactly what I wanted to say in my book. Their book has been described as the most important book on decision making since Daniel Kahneman's "*Thinking, Fast and Slow*", which is the book that I mentioned in Chap. 9 as one of the most important books for my own career. That is the nice thing about reading books... good authors refer to each other all the time, sometimes they reject each other's theories, while other times they confirm each other or add small changes to make them better. So, my dear readers, go read some books by these great minds, enjoy, learn from them, accept, or reject their theories, but never stop doubting (who knows if you will ever find a better theory?). I personally give these great authors the benefit of the doubt (*no pun intended*) and I am humble enough not to question such great minds, but as far as my own research is concerned, I know that doubt has often been the first step to a publication. So never stop doubting!

16.2 Ignorance

While doubt is a feeling of insecurity that leads you to question everything you know so far, ignorance is simply a lack of knowledge and information. Doubting about knowledge that you do not (yet) have is of course more difficult than questioning your existing knowledge, and therefore ignorance carries dangers. Researchers are constantly looking for answers or solutions that are often very difficult to find. Sometimes those answers just do not exist, and if you search for something that does not exist, you never stop looking and you will never be successful in your research. In the book "*How emotions are made*" (Barrett, 2018), the interesting (but often controversial) *theory of constructed emotions* is introduced as a scientific theory to explain the experience and perception of emotions. Although I think that the theory is interesting to understand how emotions are created, the book also tells that the human brain is a master of deception. What we—humans—experience as certainty, i.e., the feeling of knowing what is true about the world, is often an illusion. Ignorance is sometimes disguised as (false) knowledge, directing our quest to avoid the danger of endless searching for something that does not exist. The author of the book refers in the last chapter to the well-known proverb of an unknown author:

> It is very difficult to find a black cat in a dark room, especially when there is no cat.

I really like that quote, although I am still not sure if I understand it correctly. To me, the dark cat represents false assumptions that people make to guide their search for answers and avoid the trap of looking for something that does not exist. When you look for the cat in a room, you have assumed that there is a cat, which means that you have not defined your problem well. It is ignoring your ignorance by hypothesising about unknown knowledge (in that case a cat) beforehand, but perhaps you should have thought further, assuming that perhaps there is no cat at all. Despite the danger of eternally searching for something that does not exist, I believe that ignorance is crucial to any research project. Sometimes it is just better to look for something, not knowing what, by not assuming a cat or anything else. Just looking for something, whatever it may be, should allow researchers to ask deeper questions, even if the answers do not exist. Since Lisa Barret writes that "*progress in science isn't always about finding the answers, it is about asking better questions*", I believe this is best done by being ignorant. In many Internet forums, the difference between philosophy and theology is explained as the fact that philosophy is scientific and open-minded, concerned with evidence, while theologians have "*found their final truth*" before starting to search. When the truth has already been found, you are no longer ignorant and that is why believers so rarely doubt, while philosophers do nothing else.

Precisely, for this reason, I often advise my PhD students not to constantly read the latest state-of-the-art papers and just continue with each idea without knowledge of the existing paths to solve the problem. Explore each idea, dig into it, forget the existing assumptions, and think you are on the brink of a breakthrough, even though you may not be. It is the only way to find a new opening and explore untrodden paths. Admittedly, such an approach usually leads to nothing but sometimes forces you to think differently than you would otherwise have done. Ignorance has its own reason for being, and in some rare cases, it leads to the most beautiful results. Do not become a theologian, but a philosopher, and be ignorant in your search.

16.3 Wildness

I think that ignorance and uncertainty cannot be separated from each other. As I have argued before, a researcher's lack of knowledge can lead to uncertainty in answering questions and so research requires dedication, the ability to deal with uncertainty, and a passion to search for the unknown. So it also requires an attitude of wildness. In the book "*The illusion of separation*", Hutchins (2014) states that our modern way of thinking and learning is based on a world made up of separate blocks. It is an organised world, structured, carefully managed, and controlled, like a safe haven where everyone can find their way. But the author argues that such a world in separation does not exist as life is made up of strongly connected blocks where separation is only an illusion. The book has a broad scope that is interesting to read, but not very relevant to this book chapter. And yet I refer to the book because of chapter 13 titled "*Indigenous wisdom*" where the author writes about *wildness*. The author states that we are all on an all-consuming quest for control that limits our way of thinking about things such as organising things in business, politics, our daily lives, and beyond.[2] By removing the essential need for wildness from our lives, we have all created phobias that have suppressed or even completely wiped out our wild, intuitive spontaneity. To overcome this, the author proposes to allow wildness back into our way of thinking. The author literally writes that "*we should celebrate this wildness, not denigrate*" and he refers to Catherine Keller who said:

> In the wild waters of the world, the fish does not go under. It is in its element. Amidst the unpredictable it swims in grace.

This quote has been pivotal in the way that I guided my researchers. I try to give the researchers as much freedom as possible to think about any idea, wild or not, and give them the space to explore each idea from different angles. That is what makes research so fantastic. In that regard, I believe that research belongs to the class of *antifragile* things, defined by Taleb (2012) as things that benefit from shocks, thrive, and grow when exposed to volatility. The resilient withstands shock and stays the same, while the antifragile gets better. Research is clearly antifragile, and wildness often means going against the usual ways of working in academia. A retired professor who prefers to remain anonymous has put it this way when we

[2] I feel a bit guilty now that I have a book written about *project control*, which might just be my irresistible quest for control in project management.

complained about the endless stream of changes necessary during a review process to get an article published:

> When I have an idea that I think is worth investigating, I explore it in great detail. If I see the results disappoint, I throw it away and regret being naive enough to think the idea was a good one. But when I see results that are promising, something happens in my head. I write them down because I want to let the world know about them. I do not care which journal publishes the results, and as long as there's one that wants to publish them, I will go for it. I would never, and I repeat, never give up on my original idea and modify it just because the journal's referees ask me to. No, I would never do that. I am just too stubborn to do that. I would even pick the lowest-ranked journal and still publish the results, even if it negatively impacts my academic career. As long as I can tell my story and show the results of my hard work to the public, I am a true researcher. I do not think I am a real academic, you better call me a wild man.

Stay wild and free, my dear friends, and keep exploring the unexplored territory.

16.4 Serendipity

I can be touched by language (sometimes to tears to the delight of my children) and I like some words so much that they can compete with whole pieces of music (which can also touch me so much). One of these words is *serendipity*, a word that I came across while reading the fantastic book "*Algorithms to live by*" (Christian & Griffiths, 2016) (in chapter 9 titled "*randomness*"). I had come across the word before, but for some reason I thought it was time to look up the real meaning. I reached for my dictionary to translate the word into my native Dutch and I saw the translation *serendipiteit* (that is Dutch), which did not clarify much. As I read further into the book, I saw that the authors referred to the "*Eureka*" moment of Newton's application and Archimedes' bathtub, claiming that it is a common enough phenomenon that great discoveries are made thanks to the power of chance. Being in the right place at the right time and seeing the slot machine trigger a new idea is what happens so often. A word was invented to capture this phenomenon. The authors refer to Horace Walpole who coined the term *serendipity* based on the fail tale adventures of "*The Three Princes of Serendip*" who were always making discoveries, by accidents and sagacity, of things they were not in quest of. The word seemed to fit my current job as an academic researcher.

Since then, I have been using it at the most inappropriate times in my lectures, simply because it is too good to not do. I believe that everyone should be open to such moments of serendipity because they can make life very beautiful. What it means to me as a researcher is that it is OK to just search based on wild ideas, ignorance, and doubt because there will be a moment of serendipity. Research cannot be captured in a well-defined step-by-step approach from problem formulation to publishable results. It cannot be planned perfectly (just like a project cannot), and progress cannot always be perfectly monitored and adjusted when needed. There must always be a certain degree of randomness to make unexpected discoveries, and that is all, as I understand it, in the word serendipity. I have argued before that research is hard work, trying and failing, experimenting, and throwing away results. Only suddenly you find something beautiful without looking specifically for it, and I keep wondering if that would also apply to projects. Perhaps, my dear readers, you should not pay too much attention to all those strict scheduling algorithms and risk and control techniques that I discussed in my book, and you would better hope for a little serendipity in managing your projects. The hope of serendipity sometimes brings results that you might never have obtained otherwise. It is as American actor Charlton Heston said:

> Sometimes life drops blessings in your lap without your lifting a finger.

Serendipity is an unplanned fortunate discovery. It is the definition of academic research. It is the definition of how I (want to) live my life.

Last Words

> It is time to say goodbye.

I would like to close this book with a brief overview of the general message of this book. It turned out to be a very strange book. It did not end up being a technical book like most of my other books, even though I have provided technical details every now and then. Nor has it become a management book as I once attempted in my business novel. I would not really call it a student book either, because it does not fit into any course that I teach at universities or business schools. I think it has mainly become a book where I have written down my personal view of my own research career with a fantastic team over the last two decades. I discussed many new methodologies, showed the very latest results, and discussed project management issues that hopefully gave you new insights. But I very much hope that it has become clear that this was mainly a story of a research team, in which I mentioned some of my PhD students but forgot others who also did very nice work. I must first of all thank Tom Servranckx for being so willing to proofread my book for errors and inconsistencies. I also have to thank Gaëtane Beernaert, not only because she is the most beautiful and best woman in the world but also because she always manages to look at my books from a different perspective. She gives hints to improve each chapter and draws my attention to the many mistakes over and over again. I do not know how and why she does it, but she keeps reading (and criticising) my stuff. Finally, I also want to thank my entire team. I do not want to venture into names because the risk of forgetting someone is too great. I prefer to refer to the OR&S website to meet my team members.[3] You will see that it is a fantastic team of young (and also less young) people. This is a book by and for them, and I want to thank them once again for that. You have also noticed that in between I have used a number of quotes from (mostly) famous people. I did this because I love it, and there is not

[3] I have mentioned the website before, but I want to show you once more how you can get to know this fantastic group of people. You can see the people of the *Operations Research & Scheduling* group on www.projectmanagement.ugent.be (go to *staff* and then *members*).

much more to it. It has turned out to be a book that I am ultimately proud of, and it was a completely new way of writing for me to achieve this result.

Therefore, I hope that you have enjoyed this book as much as I did writing it. I have received a lot of nice comments for my previous books (in my mailbox, or somewhere on the Internet), and I have to admit that it makes me happy every time again. I also occasionally saw some harsh criticism which, I must confess, sometimes keeps me up for a few nights. Of course, I accept all criticism (both positive and negative), but I try very hard to remember the positive and forget the negative. Usually the latter does not work very well, which is why any negative comment is actually the best way to do better next time. And yet, dear readers, I would like to ask you a favour: If you found this book enjoyable and inspiring to read, just let me know. You would do me a great favour with it, and I promise you, in return, that you will see that pleasure in my eyes when we really meet. This also brings me seamlessly to the last quote of this book. It comes from the song "*Whistlin' past the graveyard*" from my favourite Tom Waits album "*Blue Valentine*":

> What you think is the sunshine is just a twinkle in my eye.

References

Barrett, L. F. (2018). *How emotions are made: The secret life of the brain*. London: Pan Books.

Christian, B., & Griffiths, T. (2016). *Algorithms to live by: The computer science of human decisions*. New York: Henry Holt.

Feynman, R. P. (1998). *The meaning of it all: Thoughts of a citizen scientist*. pub-PERSEUS.

Hutchins, G. (2014). *The illusion of separation: Exploring the cause of our current crises*. Floris Books.

Taleb, N. N. (2012). *Antifragile*. New York: Random House.

Tetlock, P. E., & Gardner, D. (2015). *Superforecasting: The art and science of prediction*. New York: Crown Publishing Group.

Appendix A
Operations Research & Scheduling Group

In this appendix, I have the pleasure to introduce my readers to my fantastic Operations Research & Scheduling research group. You can find more about them at the www.projectmanagement.ugent.be website or stare at the nice pictures taken after three research meetings in Ghent. Figure A.1 shows the members of my team who won the *IPMA Outstanding Research Contribution award* for the research on "data-driven project management," much of which is discussed in this book. You see from left to right Gaëtane Beernaert (my one and only), José Coelho, Annelies Martens, Mario Vanhoucke, Tom Servranckx, Jordy Batselier, and Sokko.

Figure A.2 shows the people at that time in the research group, and you see, from left to right, José Coelho, Weikang Guo, Jingyu Luo, Xi Wu, Jakob Snauwaert, Wanjun Liu, Izel Unsal-Altuncan, Annelies Martens, Tom Servranckx, Fangfang Cao, Mario Vanhoucke, Forough Vaseghi, Dries Bredael, and Rojin Nekoueian. Not on the picture are Xin Guan and Jie Song, who also belong to our OR&S team but lived in China at that time (and still do). Rob Van Eynde is also not in the photo, as he had moved to Spain a few weeks before this photo was taken. Notice the menacing clouds in the sky on the second picture. This is what we call *good weather*. Now you know why I love Lisbon so much!

Figure A.3 shows an updated version of the team in 2023, and you see, from left to right, Mario Vanhoucke, Guillaume Vermeire, Weikang Guo, Nathan Steyaert, Annelies Martens, Jakob Snauwaert, Tom Servranckx, Dries Bredael, Rojin Nekoueian, Forough Vaseghi, Ziyang Tang, Fangfang Cao, Wanjun Liu, Jingyu Luo and Yuxuan Song. Xin Guan and Jie Song are again not on the picture, but they are still OR&S members. Izel Ünsal Altuncan is neither on the picture as she could not join us on that beautiful day in Ghent. I am sure you will find much more research in the (near) future coming from these fantastic young (and less young) people. Stay tuned!

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

Fig. A.1 The OR&S members who received the IPMA research award (2020)

Fig. A.2 OR&S team picture (2022)

Fig. A.3 OR&S team picture (2023)

Appendix B
Earned Value Management (Glossary)

In this book, the *Earned Value Management* (EVM) methodology is used as a project control methodology in various ways. The provided metrics in the EVM system are used for project time forecasting (Chaps. 4 and 6), constructing control limits (Chap. 5) and are implemented in a so-called *top-down project control* system (Chap. 7) using *statistical* or *analytical* control systems (Chap. 8). Giving a full overview of all the metrics used in an EVM system would lead us too far from the central theme of this book, and this appendix gives a basic summary of most of the EVM concepts. The summary shows the main components of an EVM analysis, divided into four different layers. It is taken from a short article *"Earned Value Management: The EVM formulary"* that is available on **PM Knowledge Center** (www.pmknowledgecenter.com) and in my book that I discussed in Chap. 2 (Vanhoucke, 2016b). PM Knowledge Center is an online learning tool on Project Management and Dynamic Scheduling and includes technical articles on *baseline scheduling*, *schedule risk analysis*, and *project control*. The book contains several questions (and answers) to test your knowledge on these topics. I advise the readers to explore the short and to-the-point articles on this free website to get acquainted with the basic and advanced concepts of EVM. The summary below refers for each concept to the article title of this website.

Key Parameters

- **S-curve**: This graph displays the Planned Value (PV), the Actual Cost (AC), and Earned Value (EV) along the life of the project.
  (Article: *"Earned Value Management: The three key metrics"*)
- **PV curve**: This graph displays the Planned Value (PV) as shown in the S-curve. Since the Planned Value curve is available at the construction of the baseline schedule (before the EVM tracking), this graph is accessible separately from the S-curve.
  (Article: *"Earned Value Management: The project baseline schedule's planned value"*)

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

- **Earned Schedule** (ES): This graph displays the Earned Schedule (ES) calculated from the Earned Value and Planned Value graph along the life of the project.
  (Article: *"Measuring Time: Earned value or earned schedule?"*)

Project Performance
- **EVM Performance Dashboard**: This graph displays both the time and cost performance and divides the project performance into four regions showing time and cost performance.
- **Cost Variance** (CV): This graph displays the Cost Variance (CV = EV − AC) along the life of the project.
  (Article: *"Earned Value Management: Measuring a project's performance"*)
- **Cost Performance** (CPI): This graph displays the Cost Performance Index (CPI = EV / AC) along the life of the project.
  (Article: *"Earned Value Management: Measuring a project's performance"*)
- **Schedule Variance** (SV and SV(t)): This graph displays the Schedule Variance (SV or SV(t)) along the life of the project. Formulas used: SV = EV − PV and SV(t) = ES − AT.
  (Article: *"Earned Value Management: Reliable time performance measurement"*)
- **Schedule Performance** (SPI and SPI(t)): This graph displays the Schedule Performance Index (SPI or SPI(t)) along the life of the project. Formulas used: SPI = EV / PV and SPI(t) = ES / AT.
  (Article: *"Earned Value Management: Reliable time performance measurement"*)
- **Schedule Adherence** (p-factor): This graph displays how good the project progress follows the baseline schedule philosophy. This is known as schedule adherence and measured by the p-factor. Tip: p-factor = % schedule adherence (100% = perfect adherence).
  (Article: *"Earned Value Management: Measuring schedule adherence"*)

Project Forecasting
- **Cost Estimate At Completion** (EAC): This graph displays the estimated final cost at project completion (EAC) predicted along the life of the project. Eight forecasting versions are used, in line with research from literature.
  (Article: *"Earned Value Management: Forecasting cost"*)
- **Time Estimate At Completion** (EAC(t)): This graph displays the estimated final duration at project completion (EAC(t)) predicted along the life of the project. Three methods are used (PVM, EDM, and ESM), each using three variants.
  (Article: *"Earned Value Management: Forecasting time"*)

Forecast Accuracy
- **MAPE**: This graph displays the Mean Absolute Percentage Error as a measure of the forecast accuracy of time or cost predictions.
  (Article: *"Predicting project performance: Evaluating the forecasting accuracy"*)
- **MPE**: This graph displays the Mean Percentage Error as a measure of the forecast accuracy of time or cost predictions.
  (Article: *"Predicting project performance: Evaluating the forecasting accuracy"*)

Appendix C
Properties of Similarity

In Chap. 9, only 6 (out of 60) properties of similarity are taken into account to design the reference class forecasting system. The summary below provides a full summary of these 60 properties to give the readers an idea of possible criteria to define project similarity. The selected properties are highlighted in bold and summarised in Table 9.3 of Chap. 9.

Class 1. Basic Info
- Number of activities: The amount of activities a project contains.
- Planned duration: The total time estimated to complete the project.
- Budget at completion: The total anticipated spending to complete the project.
- Number of resources: The concrete amount of renewable or non-renewable resources that are needed to complete the activities of the project. Examples of renewable resources include: machines, workers, tools, etc. Examples of non-renewable resources include: money, raw materials, energy, etc.
- Network structure: The activities of the project can be executed in a way that is more serial or more parallel. Serial means that the activities have to be executed sequentially. Parallel means that some activities can be executed simultaneously. A combination of both is possible too.

Class 2. Risk Analysis
- Risk analysis method: There are many methods available to analyse and simulate the probability that a certain event will occur. One of the most known methods is the Monte Carlo Simulation. It is also possible to analyse the risk by making use of intuition and experience.
- Probability of event: The chance that certain events will occur.
- Impact of event: In case a certain event occurs, the negative consequences can be high or low.

Class 3. Project Control
- Project control method: There are many methods available to control a project once it is being executed. Two examples of this are the bottom-up approach and

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

the top-down approach. Bottom-up approach: the level of performance is checked when there are troubles with an activity. Top-down approach: the performance of the project is monitored continuously. When the performance is decreasing, the responsible activity has to be determined.
- Project structure: The way in which different phases of the project are structured and planned.

Class 4. Specifications
- Objective: Every project is optimised according to the objective. The objective can be time within a certain period, cost within a certain budget, net-present value to achieve the highest actual value, levelling an equal distribution of the resources, etc.
- Frequency: The number of times a certain kind of project is executed.
- **Project definition**: The project can be a straight redo, an expansion of an earlier executed project or a totally new kind of project.
- **Project complexity**: The overall complexity of a project, independently of how often it is executed.
- Number of subcontractors: The number of other subcontractors that are also working on the same project.
- Materials used: The kind of material that is needed to execute the project.
- Type of tender: There is a fixed total price or the client pays per hour for workforce or materials used. A combination of both is possible too.
- New technologies: The amount of new or recent technologies used in a project.
- **Type of deliverable**: The project can be a development of a product, a service, or a combination.
- Type of activities: Similarity in the scope of the activities to be completed in the project.
- Type of technology: The use of a specific technology, even if they have a different scope.
- Internal/external project: The project can be executed for an external client or within the company itself.
- Priority of the project: The relative importance of a project: for the client, project A could be more important than project B. In a similar way, for the executing company, a project for client 1 could be more important than a project for client 2.
- Type of financing: There are different arrangements as to how the project is financed. The financial resources could be originating straight from the client, but an intermediary bank, institution or government could also be involved. The client for whom the project is executed is not always the one who is paying the compensation.
- Automation of processes: The way processes are executed depends on the grade of automation. The process can be executed by people, machines or a combination of both.
- Profitability: How profitable is a project and how much profit margin is taken into account.

Class 5. Client

- Elation to client: How well the relationship with the client is established.
- Frequency of co-operations: Co-operations with a certain client can be frequent, occasionally or only once.
- Industry of client: The sector in which the company of the client is located.
- Cultural gap with client: A difference in geographical location between the executing company and the client could also imply a cultural difference.
- Experience of client: It is possible that the client is new to the business industry or already has a well-structured system.
- Interference of decision: The number and significance of different subjects that have to be agreed on in co-operation with the client during the execution of the project.
- Activities executed by client: The number of activities that must be completed by the client themselves during the execution of the project.
- Type of client: The client can be a private company or an institution government, school, etc.
- Level of client within company: The person or unit within the clients company for whom the project is executed could be specified. It is possible that the project is executed for a lower business unit within the company or for the highest level of management.
- Award criteria: Award criteria are qualitative criteria that serve to assess the entries. If candidates meet the selection criteria financial situation and technical competencies of a public tender, the client will review the award criteria.

Class 6. Conditions

- Location: The distance between your company and the location where the project has to be executed.
- Weather: The forecast weather conditions during the execution of the project.
- **Governmental law**: The government can impose certain rules which can affect the project duration, cost or other factors in the project.
- Conjuncture: Does the conjuncture have an influence on the type of projects in the company? It could be that companies invest a lot during times of economic expansion, which can lead to the execution of similar projects.
- Number of competitors: When there are many competitors in a particular industry, other types of projects can be executed more easily than in an industry where you have a mono- or oligopoly.

Class 7. Team

- Age project manager: The age of the person who is leading the project.
- Experience project manager: The experience of the person who is leading the project.
- Education project manager: The received education of the person who is leading the project.
- Attitude project manager: The overall attitude of the person who is leading the project. Examples include: optimistic, realistic, pessimistic, organised, leading, etc.

- Geographical location team: The distance between the people that are working on the project and the way they communicate.
- Team composition: It is possible that more than one team is working on the same project. The teams can be composed in different ways. The teams can be variable or fixed during the execution of the project.
- Background of project team: Teams can be formed by individuals with similar experience and education or by members with a completely different background.
- Independence of project team: Is the project team capable of doing things on their own without having to be told what to do step by step? Some teams need a lot of control, others can be given more free reign.
- Drive involved team members: Are the team members willing to go the extra mile to finish a project on time or are they just executing their nine-to-five job?

Class 8. Company

- Industry: The sector in which the executing company is located.
- **Experience of company**: The experience the executing company has in performing a certain kind of project.
- Country: The country in which the executing company is located.
- Reason of assignment: The reason a client selects your company to execute the project can differ. Examples of reasons are: your company offers the lowest price, your company provides the best quality, your company has the most experience, etc.

Class 9. Stakeholders

- Relationship to suppliers: How well the relationship with the supplier is established. In some cases, the completion of a project can depend on suppliers. They are needed for providing information, logistics, workforce, tools, etc.
- Complexity of contractual chain: The complexity depends on the number of links in the chain. Complexity can increase if there are multiple clients/contractors for the same output/input. It is possible that the same project has to deliver two or more outcomes that have different requirements, timings and priorities.
- Variety of stakeholders: The variety of stakeholders involved in the project.
- Number of stakeholders: The number of stakeholders involved in the project.
- Activities by contractors: The number of activities that must be completed by other contractors during the execution of the project.
- **Impact on employees**: The project can be mentally exhausting when there is a lot of pressure on employees.

Appendix D
Patterson Format

The Patterson format discussed in Chap. 11 is a simple text file and its structure is explained on the illustrative project network of Fig. D.1. Each number above the node is assumed to be the activity duration. The network has two dummy activities, i.e., dummy start node 1 and dummy end node 14, and hence, the network contains 14 activities in total, dummies included. The Patterson format also makes use of start and end dummy nodes and is structured as follows:

Line 1 shown two numbers:

- Number of activities (starting with node 1 and two dummy nodes inclusive)
- Number of renewable resources

Line 2 shows one number for each resource:

- Availability for each renewable resource

The next lines show one line for each activity, starting with a dummy start activity and ending with a dummy end activity:

- Activity duration
- Resource requirements for each resource type
- Number of successors
- Activity ID for each successor

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

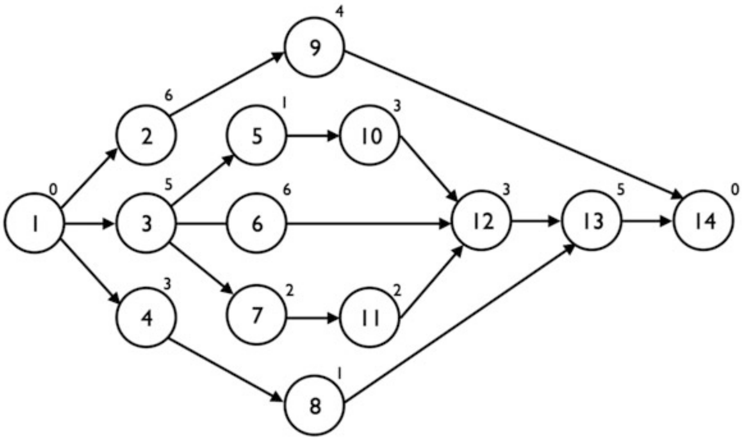


Fig. D.1 An illustrative activity network (Source: Vanhoucke, 2012b)

Suppose that the project network of Fig. D.1 needs four types of so-called renewable resources. Consequently, the Patterson text file for the network of the figure is as follows:

```
14   4
10   20   8    10
0    0    0    0    0    1    2    3    4
6    7    15   2    6    1    9
5    1    8    4    8    3    5    6    7
3    5    8    3    3    1    8
1    6    15   2    6    1    10
3    1    13   0    3    1    12
2    2    16   2    0    1    11
1    2    9    4    4    1    13
4    8    12   5    5    1    14
3    6    17   5    0    1    12
1    2    10   2    5    1    12
3    6    5    5    4    1    13
5    8    10   3    7    1    14
0    0    0    0    0    0
```

As an example, activity 2 of Fig. D.1 needs 7, 15, 2, and 6 units of resource 1, 2, 3, and 4, respectively. The availability of these resources is set at maximum 10, 20, 8, and 10 units.

Appendix E
Network and Resource Indicators

Although most of the scheduling algorithms used in this book make use of the *critical path scheduling* method (and ignore the presence of limited resources), some chapters extend this scheduling problem to the *resource-constrained project scheduling problem* (RCPSP, Chaps. 6 and 11). This problem is well known in the academic literature and is denoted as $m, 1|cpm|C_{max}$ using the classification scheme of Herroelen et al. (1999) or $PS|prec|C_{max}$ using the scheme of Brucker et al. (1999). The problem is defined by a set of activities N that must be scheduled without pre-emption on a set R of renewable resource types. The set N consists of $n = |N|$ non-dummy activities, starting at activity 1 to activity n. Traditionally, the network is extended with two dummy activities 0 and $n + 1$.[1] Each activity i has a deterministic duration d_i and requires r_{ik} units of resource type $k = 1, \ldots R$, which has a constant availability a_k throughout the project horizon. We assume that $r_{ik} \leq a_k (i \in N, k \in R)$. The dummy start and end activities 0 and $n + 1$ have zero duration and do not make use of the renewable resources, while the other activities have a non-zero duration and a non-negative resource requirement. The set A is used to refer to the set of pairs of activities between which a finish-start precedence relationship with time lag 0 exists. We assume graph $G(N, A)$ to be acyclic. A schedule S is defined by an $(n + 2)$-vector of start times $s = (s_0, ..., s_{n+1})$, which implies an $(n + 2)$-vector of finish times f $(f_i = s_i + d_i)$. A schedule is said to be feasible if the precedence and resource constraints are satisfied. The objective of the RCPSP is to find a feasible schedule such that the schedule makespan is minimised.

In the next paragraphs, the calculations for the network topology and resource parameter indicators are explained in detail. Each indicator only takes the non-dummy activities $i = \{1, \ldots, n\}$ into account and ignores the dummy activities 0 and $n + 1$.

[1] Note that the dummy start activity is sometimes activity 1 (instead of activity 0) as is, e.g., the case in Fig. D.1 of Appendix D. However, it is now assumed—without loss of generality—that 0 and $n + 1$ are used for the start and end dummy nodes.

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

Network Topology Parameters

The network structure has a major impact on the performance of scheduling algorithms, the analysis of risk metrics and the performance of project control methods, as discussed in most chapters of this book. Therefore, measuring the network topology should be done with care and different network topology indicators have been proposed in the literature.

The ***coefficient of network complexity*** (CNC) is defined as the number of direct arcs divided by the number of nodes in the network.

The ***order strength*** (OS) is measured by the number of precedence relations (including direct and indirect predecessors) divided by the theoretical maximum number of precedence relations (Mastor, 1970).

The next network topology indicators have been originally proposed in Tavares et al. (2002)[2] and later adapted by Vanhoucke et al. (2008) as the I_1 to I_6 indicators. To avoid confusion, four of these indicators (I_2, I_3, I_4 and I_6) have been renamed in my book *Measuring Time* as proposed in Table 11.1 of this book. The calculations of these indicators rest on a number of straightforward definitions that characterise the project network. The progressive and regressive level of activities in a project network have been defined by Elmaghraby (1977). The *progressive level* of an activity i in a project network is defined as:

$$PL_i = \begin{cases} 1 & \text{if } P_i = \emptyset \\ \max_{j \in P_i} PL_j + 1 & \text{if } P_i \neq \emptyset \end{cases} \tag{E.1}$$

Similarly, the *regressive level* of an activity i in a project network is defined as:

$$RL_i = \begin{cases} m & \text{if } S_i = \emptyset \\ \min_{j \in S_i} RL_j - 1 & \text{if } S_i \neq \emptyset \end{cases} \tag{E.2}$$

with m the maximal progressive level, i.e., $m = max_{i \in N} PL_i$.

Based on the definition of the progressive and regressive level in an activity network, the following definitions can be used:

- *Width:* The width w_a of each progressive level $a = \{1, \ldots, m\}$ is defined as the number of activities at that level.
- *Length of an arc:* The length l of an arc (i, j) is equal to the difference between the progressive level of the end node j and the start node i, i.e., $PL_j - PL_i$.
- *Topological float:* The topological float of an activity i is equal to the difference between the regressive level and the progressive level of activity i, i.e., $RL_i - PL_i$.

[2] Luis Valadares Tavares was the advisor of José Coelho and we have met each other in 2001 for the very first time when José was working on his RiskNet generator. It was the start of the fruitful collaboration that I discussed in several pages of this book.

The size of the project network is measured by the ***number of activities*** in the project, which corresponds to the number of non-dummy nodes in the network (n or $|N|$ or I_1).

The ***serial/parallel indicator*** (SP or I_2) measures the closeness of a network to a serial or parallel network. When SP = 0 then all activities are in parallel, and when SP = 1 then the project network is completely serial. Between these two extreme values, networks can have any topology close to either a serial or parallel network. The SP indicator determines the maximal number of levels of the network, defined as the longest chain (in terms of the number of serial activities) in the network. Its formula is given by the following equation:

$$SP = \begin{cases} 1 & \text{if } n = 1 \\ \frac{m-1}{n-1} & \text{if } n > 1 \end{cases} \tag{E.3}$$

The ***activity distribution indicator*** (AD or I_3) measures the distribution of project activities along the levels of the project by taking the width of each progressive level into account. When AD = 0, all levels contain a similar number of activities, and hence, the number of activities is uniformly distributed over all levels. When AD = 1, there is one level with a maximal number of activities, and all other levels contain a single activity. The indicator can be defined as follows:

$$AD = \begin{cases} 0 & \text{if } m \in \{1, n\} \\ \frac{\alpha_w}{\alpha_{max}} = \frac{\sum_{a=1}^{m} |w_a - \bar{w}|}{2(m-1)(\bar{w}-1)} & \text{if } m \notin \{1, n\} \end{cases} \tag{E.4}$$

Consequently, this indicator measures the distribution of the activities over the progressive levels by calculating the total absolute deviations α_w and α_{max}. α_w measures the total absolute deviation of the activity distribution $w = (w_1, w_2, \ldots, w_m)$ from the average deviation $\bar{w} = n/m$ as $\alpha_w = \sum_{a=1}^{m} |w_a - \bar{w}|$. α_{max} determines the maximal value of α_w for a network with n activities and m progressive levels. α_{max} corresponds to a network for which $m - 1$ progressive levels have a width w_a equal 1, and one progressive level has a width w_a equal $n - (m - 1)$. The value of α_{max} can be calculated as $\alpha_{max} = (m - 1)(\bar{w} - 1) + (n - m + 1 - \bar{w})$. The first term calculates the absolute deviation between $w_a = 1$ and the average width $\bar{w}$ for $m - 1$ progressive levels. The second term calculates the difference between $w_a = n-(m-1)$ and $\bar{w}$ for the remaining progressive levels. The formula for $m \neq 1$ can be simplified to $\alpha_{max} = 2(m - 1)(\bar{w} - 1)$, resulting in the AD indicator defined above. This indicator equals 1 when $\alpha_w = \alpha_{max}$. At the other extreme, the indicator has a value of 0 when the activities are uniformly distributed over the progressive levels, i.e., $w_a = \bar{w} = n/m$ (for $a = \{1, \ldots, m\}$).

The ***length of arcs indicator*** (LA or I_4) measures the length of each precedence relation (i, j) in the network as the difference between the level of the end activity j and the level of the start activity i. When LA equals 0, the network has many precedence relations between two activities on levels far from each other. Hence, the activity can be shifted further in the network. When LA equals 1, many precedence relations have a length of one, resulting in activities with immediate successors on the next level of the network, and hence little freedom to shift. In order to define the LA indicator, a parameter n'_l is defined as the number of arcs in the network with length l (note that the length l of an arc can vary between 1 (short) and $m-1$ (long)). Based on this parameter, the LA $\in$ [0, 1] indicator measures the presence of short (i.e., with a length $l=1$) immediate precedence relations and can be defined as follows:

$$LA = \begin{cases} 1 & \text{if } D = n - w_1 \\ \frac{n'_1 - n + w_1}{D - n + w_1} & \text{if } D > n - w_1, \end{cases} \tag{E.5}$$

where D stands for the maximal number of short ($l=1$) precedence relations in a network, given the width of each level, i.e., $D = \sum_{a=1}^{m-1} w_a \times w_{a+1}$.

The next indicator does not occur in Table 11.1 and has not received a name other than the I_5 since it is very similar to the LA indicator but measures the presence of long arcs (instead of the short arcs) with a difference between the progressive level of the end node and the start node of each arc bigger than 1 (it should therefore be called the **long arcs indicator**). I_5 is equal to 0 for a network with $n - w_1$ arcs with a length of 1 and all other arcs a maximum length of $m-1$, while $I_5 = 1$ for networks where all arcs have a length equal to 1. It is defined as:

$$I_5 = \begin{cases} 1 & \text{if } |A| = n - w_1 \\ \frac{\sum_{l=2}^{m-1} n'_l \frac{m-l-1}{m-2} n'_1 - n + w_1}{|A| - n + w_1} & \text{if } |A| > n - w_1 \end{cases} \tag{E.6}$$

with $|A|$ the number of direct precedence relations between the activities (excluding the relations with the dummy activities).

The ***topological float indicator*** (TF or I_6) measures the topological float of a precedence relation as the number of levels each activity can shift without violating the maximal level of the network (as defined by SP). Hence, TF = 0 when the network structure is 100% dense and no activities can be shifted within its structure with a given SP value. A network with TF = 1 consists of one chain of activities without topological float (they define the maximal level and hence, the SP value), while the remaining activities have a maximal float value (which equals the maximal level, defined by SP, minus 1).

$$TF = \begin{cases} 0 & \text{if } m \in \{1, n\} \\ \frac{\sum_{i=1}^{n} RL_i - PL_i}{(m-1)(n-m)} & \text{if } m \notin \{1, n\}. \end{cases} \tag{E.7}$$

Resource Parameters

The first and most obvious resource parameter to measure the availability of resources in a project is the ***number of resources*** ($|R|$) in the project. Each resource $k = \{1, \ldots, |R|\}$ has a certain availability a_k and each activity i of the project makes use of an integer number $0 \leq r_{ik} \leq a_k$ for each resource type k. The relation between the resource availability (a_k) and the resource requirements (r_{ik}) is measured by four well-known resource parameters that have been implemented in the *RanGen1* (Demeulemeester et al., 2003) and *RanGen2* (Vanhoucke et al., 2008) network generators. They will be reviewed along the following lines.

The ***resource factor*** (RF) measures the average portion of resource types requested per activity. This indicator is introduced by Pascoe (1966) and in later studies by Cooper (1976) and Alvarez-Valdes and Tamarit (1989) defined as:

$$RF = \frac{1}{nK} \sum_{i=1}^{n} \sum_{k=1}^{|R|} \begin{cases} 1 & \text{if } r_{ik} > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{E.8}$$

where—as defined earlier—n denotes the number of activities (excluding dummy activities), $|R|$ denotes the number of resource types and r_{ik} denotes the amount of resource type k required by activity i. Since the resource factor reflects the average portion of resource types requested per activity, it measures the density of the matrix r_{ik}.

An alternative metric to measure this density is given by the ***resource use*** (RU), which measures for each activity the number of resource types. It is introduced by Demeulemeester et al. (2003) and varies between zero and the number of resource types available, defined as:

$$RU_i = \sum_{k=1}^{|R|} \begin{cases} 1 & \text{if } r_{ik} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{E.9}$$

In most project networks available in the literature, $RU_i = RU$ to assure that each activity uses the same number of resources.

The ***resource strength*** (RS) is a combined indicator since it uses both network topology information and resource demand in its calculations. It was first introduced by Cooper (1976) and later used by Alvarez-Valdes and Tamarit (1989). Inspired by their work, Kolisch et al. (1995) redefined the RS indicator as follows:

$$RS_k = \frac{a_k - r_k^{min}}{r_k^{max} - r_k^{min}}, \tag{E.10}$$

where a_k denotes the total availability of renewable resource type k, r_k^{min} equals $\max_{i \in N} r_{ik}$ and r_k^{max} denotes the peak demand of resource type k in the precedence preserving earliest start schedule. Since this indicator does not always lie in the interval between 0 and 1, Demeulemeester et al. (2003) has extended the indicator with an extra condition and is calculated as:

$$RS = \begin{cases} 1 & \text{if } a_k \geq r_k^{max} \text{ or } r_k^{max} = r_k^{min} \\ \frac{a_k - r_k^{min}}{r_k^{max} - r_k^{min}} & \text{otherwise} \end{cases} \tag{E.11}$$

The ***resource constrainedness*** (RC) measures the average resource requirement for all activities for a particular resource divided by the availability of that resource. It has been introduced by Patterson (1976) as follows:

$$RC_k = \frac{\overline{r}_k}{a_k}, \tag{E.12}$$

where a_k is defined as above and $\overline{r}_k$ denotes the average quantity of resource type k demanded when required by an activity, calculated as:

$$\overline{r}_k = \frac{\sum_{i=1}^{n} r_{ik}}{\sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ik} > 0 \\ 0 & \text{otherwise.} \end{cases}} \tag{E.13}$$

Appendix F
Network × Resources = NetRes

This appendix briefly describes the settings for the new NetRes (i.e., network and resources) dataset, containing two separate sets. The first network set is proposed in Vanhoucke et al. (2008) and has been called the MT set in Vanhoucke et al. (2016). It contains four subsets with diverse settings for the network structure. The second set called ResSet is completely new and contains four subsets with only resource information. These ResSet instances should be combined with the MT set to construct the so-called NetRes data instances that can be used for solving resource-constrained project scheduling problems.

Network Data

In Vanhoucke et al. (2016), it has been shown that the MT set contains four subsets to diversify the network structure as much as possible. The set does not contain resources and has been used for project control studies in which the use of resources is completely ignored. The four subsets contain in total 4100 networks for which the network indicator settings are summarised along the following lines. Each network contains exactly 30 activities.

Set 1: Network indicator: serial or parallel network (SP)

SP = 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9 and AD, LA and TF random from interval [0, 1]

→ Using 100 instances for each setting, 900 project network instances have been generated.

Set 2: Activity indicator: activity distribution (AD)

Set 2.1: AD = 0.2; 0.4; 0.6; 0.8, SP = 0.2 and LA and TF random from interval [0, 1]

Set 2.2: AD = 0.2; 0.4; 0.6; 0.8, SP = 0.5 and LA and TF random from interval [0, 1]

→ Using 100 instances for each setting, 2 * 400 = 800 project network instances have been generated for this subset.

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

Set 3: Precedence relations indicator: length of arcs (LA)

Set 3.1: LA = 0.2; 0.4; 0.6; 0.8, SP = 0.2 and AD and TF random from interval [0, 1]
Set 3.2: LA = 0.2; 0.4; 0.6; 0.8, SP = 0.5 and AD and TF random from interval [0, 1]
Set 3.3: LA = 0.2; 0.4; 0.6; 0.8, SP = 0.8 and AD and TF random from interval [0, 1]

→ Using 100 instances for each setting, 3 * 400 = 1200 project network instances have been generated for this subset.

Set 4: Float indicator: topological float (TF)

Set 4.1: TF = 0.2; 0.4; 0.6; 0.8, SP = 0.2 and AD and LA random from interval [0, 1]
Set 4.2: TF = 0.2; 0.4; 0.6; 0.8, SP = 0.5 and AD and LA random from interval [0, 1]
Set 4.3: TF = 0.2; 0.4; 0.6; 0.8, SP = 0.8 and AD and LA random from interval [0, 1]

→ Using 100 instances for each setting, 3 * 400 = 1200 project network instances have been generated for this subset.

Resource Data

The generation of resources to create the ResSet dataset has been done in such a way that enough data instances are available that can be combined with the network instances to test the impact of resources in detail. The set contains four subsets. The resource availability has been generated for a varying number of resources (NR). The resource demand has been generated for each activity, with a total of 30 activities such that the resource data can be combined with the network instances of the MT set (which has 30 activities too). The demand has been generated to satisfy varying values for the resource use (RU) and the resource constrainedness (RC). The RC values are fixed values for each resource, except for the fourth set where the RC values differ for each resource (but the average RC value is controlled).

Set 1: Basic R4 Set

NR = 4
RU = 2; 4
RC = 0.25; 0.50; 0.75 (fixed values for each resource)

→ Using 100 instances for each setting, 600 resource files have been generated.

Set 2: R4 Set with extended RC values

NR = 4
RU = 2; 4
RC = 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9 (fixed values for each resource)

→ Using 100 instances for each setting, 1800 resource files have been generated.

Set 3: R10 Set with extended RU values

NR = 10
RU = 2; 6; 10
RC = 0.25; 0.50; 0.75 (fixed values for each resource)

→ Using 100 instances for each setting, 900 resource files have been generated.

Set 4: R10 Set with variable RC values

NR = 10
RU = 2; 6; 10
RC = 0.25; 0.50; 0.75, varying for each resource in a predefined interval, as follows:

Average RC = 0.25: [0.12, 0.63] or [0.10, 0.90]
Average RC = 0.50: [0.25, 0.75] or [0.10, 0.90]
Average RC = 0.75: [0.37, 0.88] or [0.10, 0.90]

→ Using 100 instances for each setting, 1800 resource files have been generated.

Appendix G
Example Project Card

In Chap. 13, an empirical project dataset of 181 real projects has been presented. With the aim of obtaining a clear overview of the projects in the database, the so-called project cards were introduced. A project card summarises the whole of the project data (i.e., general characteristics, risk information, and tracking data) for a certain project in an orderly and structured manner. Its main structure originates from the three dimensions of dynamic scheduling (baseline scheduling, risk analysis, and project control) as presented at different places in this book. The presentation of the project cards is logically constructed according to the three dynamic scheduling dimensions and preceded by an introductory subsection which describes the header of a project card. For illustration, an example project card is included in the next pages in Figs. G.1, G.2, and G.3. Moreover, the project cards for most projects in the database are available on the supporting website (www.projectmanagement.ugent.be/research/data).

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

OR-AS Operations Research Applications and Solutions	Case Name: **Pumping Station Jabbeke**	Sector	Construction (Civil)	
	OR-AS Operations Research - Applications and Solutions www.or-as.be info@or-as.be	Baseline Schedule	Schedule with resources	
			Schedule with costs	
		Risk Analysis	Random simulation	
			One of nine std. scenarios	
Submitted by	Niels Crommen		User defined distributions	
Date	December 20, 2012	Project Control	Automatic tracking	
File Name	C2012-13 Pumping Station Jabbeke.p2x		Tracking based on user input	

1. Project description

Project authenticity

The renovation of three pumps used to prevent creeks from overflowing by removing the water from the polder and pumping it into the nearby canal. All main activities are controlled by a general low voltage panel, which also has to be installed. The pumping station is situated in Jabbeke (Belgium).

The project consists of activity, resource and cost data that were obtained directly from the actual project owner.

2. Project properties

2.1. Baseline Schedule

General	
# Activities	74
Planned Duration (PD)	125 days*
Budget At Completion (BAC)	366,410 €
Renewable Resources	2
Consumable Resources	-

* standard eight-hour working days

Network topology	
Serial/Parallel (SP)	64%
Activity Distribution (AD)	59%
Length of Arcs (LA)	3%
Topological Float (TF)	27%

2.2. Risk Analysis

Random simulation by ProTrack was performed using the default symmetric triangular risk distribution profiles.

	Cost sensitivity		
	avg [%]	std dev [%]	skew [-]
CRI-r	10.2	10.6	2.8
CRI-rho	15.2	14.9	1.6
CRI-tau	32.4	24.8	1.3

	Resource sensitivity		
	avg [%]	std dev [%]	skew [-]
CRI-r	54.0	44.0	N/A
CRI-rho	55.0	43.0	N/A
CRI-tau	43.0	43.0	N/A

	Time sensitivity		
	avg [%]	std dev [%]	skew [-]
CI	58.5	48.6	-0.3
SI	64.1	42.6	-0.5
SSI	4.5	10.7	6.0
CRI-r	10.1	12.4	3.8
CRI-rho	14.1	16.1	2.2
CRI-tau	34.7	24.2	1.3

Fig. G.1 Example project card (Part 1)

2.3. Project Control

2.3.1. Simulated forecasting accuracy

The accuracy of time and cost forecasting methods has been evaluated based on Monte Carlo simulation runs using the risk profiles described in section "2.2. Risk Analysis". Based on these risk profiles, the Mean Absolute Percentage Error (MAPE) and Mean Percentage Error (MPE) has been calculated to evaluate the expected accuracy of the time and cost predictions, EAC(t) and EAC, respectively.

Simulated EAC(t) accuracy		
method - PF	MAPE [%]	MPE [%]
PV - 1	14.3	-14.3
PV - SPI	21.2	4.0
PV - SCI	21.2	4.3
ED - 1	21.2	-21.2
ED - SPI	21.2	4.0
ED - SCI	21.2	4.1
ES - 1	12.4	-11.4
ES - SPI(t)	18.6	15.4
ES - SCI(t)	18.7	15.5

Simulated EAC accuracy		
method (PF)	MAPE [%]	MPE [%]
1	0.3	-0.2
CPI	0.3	0.0
SPI	12.2	12.2
SPI(t)	14.6	14.5
SCI	12.3	12.2
SCI(t)	14.6	14.6
0.8 CPI + 0.2 SPI	4.0	3.9
0.8 CPI + 0.2 SPI(t)	5.5	5.5

According to the MAPE values[1] the best performance for time forecasting can be expected from the unweighted Earned Schedule method. For cost forecasting the unweighted and CPI-weighted methods should yield the best results.

2.3.2. Tracking description

Tracking authenticity

Manual tracking was performed over 28 tracking periods with a length of approximately one week. The Real Duration and Real Cost mentioned in section "2.3.3. Earned Value Management" are based on manual user input.

The tracking information obtained from the project owner and introduced in ProTrack includes actual activity start dates, durations and costs.

[1] The MAPE gives the best indication for the forecast accuracy (the lower the MAPE, the more accurate the method) since all deviations from the targeted real duration (real cost) are cumulated, whereas for the MPE underestimates can be compensated by overestimates and vice versa, possibly leading to an overly positive evaluation of a certain method. However, the MPE can provide useful information about the nature of the deviations, i.e. does the method rather underestimate or overestimate the real duration (real cost)?

Fig. G.2 Example project card (Part 2)

2.3.3. Earned Value Management

2.3.3.1. Performance metrics

	CV [€]	SV [€]	SV(t) [d]	CPI [-]	SPI [-]	SPI(t) [-]	p-factor [-]
avg	-5,717	6,471	-6.01	0.98	1.00	0.97	0.98
std dev	5,381	17,755	8.89	0.02	0.10	0.12	0.04
final	-14,101	0	-15.00	0.96	1.00	0.89	1.00

2.3.3.2. Time forecasting

PD	125 days	Real Duration	140 days	Late	12.00%

EAC(t)			Real Accuracy	
method - PF	avg [d]	std dev [d]	MAPE [%]	MPE [%]
PV - 1	127.41	6.60	9.0	-9.0
PV - SPI	125.72	11.14	10.5	-10.2
PV - SCI	128.47	13.46	9.6	-8.2
ED - 1	128.50	7.48	8.5	-8.2
ED - SPI	126.83	12.03	10.0	-9.4
ED - SCI	127.46	12.59	9.7	-9.0
ES - 1	131.01	8.89	7.7	-6.4
ES - SPI(t)	130.16	15.09	10.2	-7.0
ES - SCI(t)	130.90	15.73	10.3	-6.5

2.3.3.3. Cost forecasting

BAC	336,410 €	Real Cost	350,511 €	Over Budget	4.19%

EAC			Real Accuracy	
method (PF)	avg [€]	std dev [€]	MAPE [%]	MPE [%]
1	342,127	5,381	2.4	-2.4
CPI	343,266	6,632	2.1	-2.1
SPI	337,585	18,034	4.1	-3.7
SPI(t)	338,319	21,039	4.7	-3.5
SCI	338,860	19,596	4.1	-3.3
SCI(t)	339,646	22,476	5.0	-3.1
0.8 CPI + 0.2 SPI	341,966	9,025	2.5	2.4
0.8 CPI + 0.2 SPI(t)	342,046	9,433	2.5	-2.4

Fig. G.3 Example project card (Part 3)

Appendix H
OR&S Project Datasets

This appendix provides some details for the different project data libraries presented in Chap. 15. All libraries can be downloaded from www.projectmanagement.ugent.be/research/data.

Problem Type: RCPSP: 1 instance = 1 project, 1 activity mode
Each dataset contains a number of projects (# instances), and for each project instance, the number of activities per project (# activities) is given in Table H.1.

Problem Type: MMRCPSP: 1 instance = 1 project, multiple activity modes
Each dataset contains a number of project instances (# instances). For each project instance, the number of activities is given (# activities). For each activity, the average number of modes (# modes) is also given in Table H.2.

Problem Type: RCPSP-AS: 1 instance = 1 project, multiple network subgraphs
Each dataset contains a number of instances (# instances). Each instance is a project and consists of a number of alternative subgraphs in the network under different settings (Subgraphs). Each alternative branch of the project contains a number of activities (# activities / branch). Since each project has multiple alternative branches (from which one or more must be selected), the total project can contain up to +700 activities (but not all of them will be selected). Details are given in Table H.3.

Problem Type: MSRCPSP: 1 instance = 1 project, resource skills, 1 activity mode
Each dataset contains a number of projects (# instances) and the number of activities per project (# activities) and some typical characteristics are given for each project instance, as shown in Table H.4 for both the multi-skill problem and the software scheduling problem.

Problem Type: RCMPSP: 1 instance = 1 portfolio, multiple projects, 1 activity mode

M. Vanhoucke, *The Illusion of Control*, Management for Professionals,
https://doi.org/10.1007/978-3-031-31785-9

Table H.1 Various RCPSP libraries (RCLIB)

	# instances	# activities	Remark
RG30	1800	30	
RG300	480	300	
DC1	1800	10–50	
DC2	720	25–100	
MT	4100	30	These projects contain no relevant resource data
ResSet	–	–	4 files with only resource data
1kNetRes	3810	30	1kNetRes is a subset of the full NetRes dataset. You can create the full NetRes dataset as NetRes = MT $\times$ ResSet (3,810,000 instances)
CV	623	$\leq$ 30	This dataset contains small but very hard instances
sD	390	50	This dataset contains instances with sigma-distance values from 3 to 16

Table H.2 Multi mode library (MMLIB)

	# instances	# activities	# modes
MMLIB50	540	50	3
MMLIB100	540	100	3
MMLIB+	3240	50 or 100	3, 6 or 9

Table H.3 Alternative subgraph library (ASLIB)

	# instances	# activities / branch	Subgraphs
ASLIB0	72,000	10	Nested/linked
ASLIB1	720	10 or 50	Basic problem
ASLIB2	6480	10 or 50	Nested/linked
ASLIB3	5760	10 or 50	Multiple/split
ASLIB4	11,520	10 or 50	Caused/closed
ASLIB5	19,440	10 or 50	All settings
ASLIB6	14	[2, 18]	Empirical data

Table H.4 Multi-skilled libraries (MSLIB and SSLIB)

	# instances	# activities	Characteristic
MSLIB1	6600	30	Basic workforce data
MSLIB2	9000	30, 60, 90	Small and large projects
MSLIB3	11,880	30	Extended workforce data
MSLIB4	5000	30	Hard set
MSLIB5	19	[13,95]	Empirical data
SSLIB1	6.6	30	Basic workforce data
SSLIB2	9	30,60,90	Small and large projects
SSLIB3	11.88	30	Extended workforce data
SSLIB4	5	30	Hard set of instances

Each dataset contains a number of instances (# instances). Each instance is now a project portfolio and consists of a number of projects (# projects). Each project in the portfolio contains a number of activities (# activities), shown in Table H.5.

Table H.5 Multi-project library (MPLIB)

	# instances	# projects	# activities
MPLIB1	4547	6, 12 or 24	60
MPLIB2	35,085	10, 20 or 30	50

Table H.6 Empirical database (DSLIB)

	# instances	# activities	# resources
Construction	122	7–1796	0–27
Engineering	11	7–31	2–4
Mobility	6	19–41	2–8
Event management	13	19–75	1–15
IT	27	11–279	1–24
Education	2	112–134	1–3

Problem Type: Empirical projects: 1 instance = 1 project
Each dataset contains a number of projects (# instances) from a certain sector, and for each project instance, the number of activities per project (# activities) and the number of renewable resources (# resources) is given in Table H.6

References

Alvarez-Valdes, R., & Tamarit, J. (1989). Heuristic algorithms for resource-constrained project scheduling: A review and empirical analysis. In R. Slowinski, & J. Weglarz (Eds.), *Advances in project scheduling* (pp. 1–26). Amsterdam: Elsevier.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research, 112*, 3–41.

Cooper, D. (1976). Heuristics for scheduling resource-constrained projects: An experimental investigation. *Management Science, 22*, 1186–1194.

Demeulemeester, E., Vanhoucke, M., & Herroelen, W. (2003). RanGen: A random network generator for activity-on-the-node networks. *Journal of Scheduling, 6*, 17–38.

Elmaghraby, S. (1977). *Activity networks: Project planning and control by network models*. New York: Wiley.

Herroelen, W., Demeulemeester, E., & De Reyck, B. (1999). A classification scheme for project scheduling problems. In J. Weglarz (Ed.), *Project scheduling - recent models, algorithms and applications* (pp. 1–26). Dortrecht: Kluwer Academic Publishers.

Kolisch, R., Sprecher, A., and Drexl, A. (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science, 41*, 1693–1703.

Mastor, A. (1970). An experimental and comparative evaluation of production line balancing techniques. *Management Science, 16*, 728–746.

Pascoe, T. (1966). Allocation of resources—CPM. *Revue Française de Recherche Opérationnelle, 38*, 31–38.

Patterson, J. (1976). Project scheduling: The effects of problem structure on heuristic scheduling. *Naval Research Logistics, 23*, 95–123.

Tavares, L., Ferreira, J., & Coelho, J. (2002). A comparative morphologic analysis of benchmark sets of project networks. *International Journal of Project Management, 20*, 475–485.

Vanhoucke, M. (2012b). *Project management with dynamic scheduling: Baseline scheduling, risk analysis and project control* (Vol. XVIII). Springer.

Vanhoucke, M. (2016b). *Project management knowledge sourcebook: The technical guide to project scheduling, risk and control*. Springer.

Vanhoucke, M., Coelho, J., & Batselier, J. (2016). An overview of project data for integrated project management and control. *Journal of Modern Project Management, 3*(2), 6–21.

Vanhoucke, M., Coelho, J., Debels, D., Maenhout, B., & Tavares, L. (2008). An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research, 187*, 511–524.