

Improved Approximation Algorithms for Optimization Problems in Graphs with Superlogarithmic Treewidth

Artur Czumaj¹, Andrzej Lingas², and Johan Nilsson²

¹ Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA. czumaj@cis.njit.edu.

² Department of Computer Science, Lund University, 22100 Lund, Sweden. Andrzej.Lingas@cs.lth.se, f98jn@efd.lth.se.

Abstract. In this paper we present two novel generic schemes for approximation algorithms for optimization $\mathcal{NP}$ -hard graph problems constrained to partial k -trees. Our first scheme yields *deterministic polynomial-time algorithms* achieving typically an approximation factor of $k/\log^{1-\epsilon} n$, where $k = \text{polylog}(n)$. The second scheme yields *randomized polynomial-time algorithms* achieving an approximation factor of $k/\log n$ for $k = \Omega(\log n)$. Both our approximation methods lead to the best known approximation guarantees for some basic optimization problems. In particular, we obtain best known polynomial-time approximation guarantees for the classical *maximum independent set* problem in partial trees.

1 Introduction

In this paper we investigate approximation algorithms for several optimization graph problems constrained to partial k -trees. For a natural k , a *partial k -tree* is a subgraph of a k -tree. A k -tree is either a k -clique, i.e., a complete graph on k vertices, or a graph which can be obtained from another k -tree by addition of a vertex with k edges connecting it to k pairwise adjacent vertices in the other k -tree [4]. Each partial k -tree admits the so called tree-decomposition of width k and *vice versa*, each graph which has a tree-decomposition of width k is a partial k -tree (see [3,4,8] and Preliminaries).

Shortly after introducing the notion of partial trees, it has been shown that many $\mathcal{NP}$ -hard graph problems constrained to partial $O(1)$ -trees admit polynomial-time or even linear-time exact algorithmic solutions [3,4,8] (e.g., these definable in extended weak monadic second-order logic [4]). For some of these $\mathcal{NP}$ -hard problems (e.g., these in the so called C -ECC class or those in the so called C -LCC class having $O(1)$ maximum degree) analogous, polynomial-time solutions are possible for partial $O(\log n)$ -trees [8]; a standard example here is the maximum independent set problem [8].

In this paper, we investigate the approximability status of some of the aforementioned $\mathcal{NP}$ -hard problems, where our main interest is on partial k -trees with

$k = \Omega(\log n)$. We focus our study on the maximum independent set and maximum clique problems and further applications of our methods are obtained by extensions of our algorithms for maximum independent set.

The *maximum independent set* problem (or equivalently, the *maximum clique* problem) is a classical $\mathcal{NP}$ -hard optimization problem studied intensively for a three decades (these problems were proven to be $\mathcal{NP}$ -hard in Karp's original paper on $\mathcal{NP}$ -completeness [15]). For general graphs, the best known polynomial-time approximation algorithm for maximum independent set (or, equivalently, maximum clique) achieves solely $n \log \log^2 n / \log^3 n$ factor [12]. On the other hand, it is known that unless $\mathcal{NP} \subseteq \text{ZPTIME}(2^{(\log n)^{O(1)}})$, no polynomial-time algorithm can achieve the approximation guarantee of $n^{1-O(1/(\log n)^\gamma)}$ for some constant γ [16] (see also [11,14] for other results in this theme). Better approximation bounds are achievable for special classes of graphs. For partial trees, one can fairly easily color any partial k -trees with $k+1$ colors (see Lemma 1 for details) what immediately implies a polynomial-time $(k+1)$ -approximation algorithm for maximum independent set. For $k = O(\log n)$, Bodlaender [8] presented a polynomial-time algorithm that returns an optimal solution. Another known approach, developed for k -clique-free graphs in [1,19], does not yield any non-trivial approximation bounds when $k = \Omega(\log n)$. A partial k -tree is $(k+2)$ -clique-free graph by definition. The technique by Ajtai, Komlós, and Szemerédi [1] treated by Shearer [19] as a *randomized* greedy algorithm can be made deterministic to find an independent set in k -clique-free graphs of size $\Omega(n^{1/(k-1)} \cdot (\log n)^{(k-2)/(k-1)})$ in polynomial time. The latter bound for $k = \Omega(\log n)$ yields merely $O(\log n)$ size independent set for a partial k -tree and therefore it implies only a trivial approximation factor in this case.

1.1 New Contribution

We present two novel generic schemes for approximation algorithms for maximum independent set and other optimization $\mathcal{NP}$ -hard graph problems constrained to partial k -trees. Our first scheme leads to *deterministic* polynomial-time algorithms that achieve typically an approximation factor guarantee of $k / \log^{1-\epsilon} n$ for all $k = \text{polylog}(n)$. The second scheme yields *randomized* polynomial-time algorithms achieving an approximation factor of $k / \log n$ for $k = \Omega(\log n)$.

These two generic schemes lead to significantly improved approximation bounds for maximum independent set. Our deterministic algorithm improves the previous approximation guarantees for all $k = \omega(\log n)$ and $k \leq \text{polylog}(n)$, and our randomized algorithm does so for all $k = \omega(\log n)$.

Besides the maximum independent set problem, the first scheme can be also adapted to the maximum induced m -colorable subgraph problem, and simplified to include the maximum clique problem (see Section 4). The second scheme can be applied to induced subgraph with hereditary property Π problems that can be solved exactly in polynomial time on graphs with given tree-decomposition of logarithmic width (e.g., members of the C -ECC class or the C -LCC class

having $O(1)$ maximum degree [8]), in particular the so called d -separable problems. All these approximation factors achievable in polynomial time are the best known for the aforementioned problems constrained to partial k -trees where k is polylogarithmic or superlogarithmic, respectively. In case a tree-decomposition of width k is not given, the approximation factors achievable by our methods increase by $O(\log k)$.

1.2 Paper Organization

In Preliminaries, we introduce the notion of tree-decomposition, treewidth, nice tree-decomposition and present several known facts on tree-decomposition construction. We describe also a simple method of approximating maximum independent set in a partial k -tree via graph coloring. In Section 3, we present our first deterministic method by applying it to maximum independent set. In the next section, Section 4, we adapt the method to the maximum induced m -colorable subgraph problem and simplify it to include the maximum clique problem. In Section 5, we exemplify our second randomized method by applying it to maximum independent set. Because of the space considerations, for the extensions of the randomized method to include maximum induced subgraph with hereditary property Π problems (that can be solved exactly in polynomial time on graphs with given tree-decomposition of logarithmic width) the reader is referred to the full version. In the final Section 6, we provide concluding remarks.

2 Preliminaries

The notion of *treewidth* of a graph was originally introduced by Robertson and Seymour [18] as one of the main contributions in their seminal graph minor project. It has turned out to be equivalent to several other interesting graph theoretic notions, e.g., the aforementioned notion of partial k -trees (see Introduction and [3,5]).

Definition 1. A tree-decomposition of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T = (I, F))$, where $\{X_i \mid i \in I\}$ is a collection of subsets of V , and $T = (I, F)$ is a tree, such that the following conditions hold:

1. $\bigcup_{i \in I} X_i = V$,
2. for all edges $(v, w) \in E$, there exists a node $i \in I$, with $v, w \in X_i$, and
3. for every vertex $v \in V$, the subgraph of T , induced by the nodes $\{i \in I \mid v \in X_i\}$ is connected.

The size of T is the number of nodes in T , that is, $|I|$. Each set X_i , $i \in I$, is called the bag associated with the i th node of the decomposition tree T . The width of a tree-decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph is the minimum width of its tree-decomposition taken over all possible tree-decompositions of the graph. A graph which has a tree-decomposition of width $O(1)$ is called a bounded treewidth graph.

Fact 1 [4] *A graph G is a partial k -tree if and only if the treewidth of G is at most k .*

Fact 2 [6] *For a partial k -tree on n vertices, a tree decomposition of width k can be found in time $O(n 2^{O(k^3)})$.*

Fact 3 [9] *For a partial k -tree on n vertices, a tree decomposition of width $O(k \log n)$ and size $O(n)$ can be found in time polynomial in n .*

Fact 3 has been refined as follows.

Fact 4 [2] *For a partial k -tree on n vertices, a tree decomposition of width $O(k \log k)$ and size $O(n)$ can be found in time polynomial in n .*

For technical reasons, it will be more convenient to use a special form of tree-decomposition termed as a *nice tree-decomposition*.

Definition 2. *A tree-decomposition $T = (I, F)$ of a graph G is nice if it fulfills the following conditions:*

1. *T is a binary rooted tree,*
2. *if a node $i \in I$ has two children j_1 and j_2 , then $X_i = X_{j_1} = X_{j_2}$ (i is called a join node),*
3. *if a node $i \in I$ has one child j , then either $X_j \subset X_i$ and $|X_i - X_j| = 1$, or $X_i \subset X_j$ and $|X_j - X_i| = 1$ (i is called an introduce or a forget node, respectively).*

Fact 5 [17] *A tree-decomposition $T = (I, F)$ of a graph G can be transformed without increasing its width into a nice tree-decomposition in time polynomial in $|I|$ and the size of G . The size of the resulting nice decomposition is $O(\ell \cdot |I|)$, where ℓ is the width of the tree-decomposition.*

Note that we may assume w.l.o.g that the leaf bags in the resulting nice tree-decomposition are singletons. For technical reasons, we shall keep this assumption for nice tree-decompositions throughout the paper.

2.1 Simple Algorithm for Coloring Partial Trees

The following simple lemma gives a simple polynomial-time $(k + 1)$ -approximation algorithm for maximum independent set in partial k -trees mentioned in Introduction.

Lemma 1. *Any partial k -tree can be colored with $k + 1$ colors in polynomial time. Consequently, in any partial k -tree, an independent set of size at least $\frac{n}{k+1}$ can be found in polynomial time too.*

Proof. It is well known that any partial k -tree and any its subgraph (as a partial ℓ -tree with $\ell \leq k$) has a vertex of degree at most k . Therefore, it is sufficient to pick such a vertex v , remove it from the graph, color the resulting subgraph inductively with $k + 1$ colors, and then color v with one of the $k + 1$ colors different from those of its neighbors. $\square$

3 Deterministic Approximation of Maximum Independent Set

In this section we present a deterministic approximation algorithm for finding maximum independent set in k -partial trees with given nice decomposition-tree of width ℓ . The algorithm traverses the nice decomposition-tree of the input graph bottom-up to collect information about independent sets in the subtrees of the tree. For each node v in the tree with the bag X_v , we would like to collect information about all independent sets induced by the vertices of the input graph that are in the subtree of the decomposition tree rooted at v . Unfortunately, in general, the number of these sets might be too large. Therefore, we consider only the sets having at most $M(n)$ nodes in each bag, where $M = \Theta(\log n / \log \log n)$. For this, with each node v we associate a set S_v containing pairs (b, r) such that $b \subseteq X_v$ and there is an independent set I on the vertices of the subtree rooted at v of size r with $I \cap X_v = b$.

Algorithm 1:

Input: A nice tree-decomposition T of a partial tree G and a parameter M .

Output: An independent set in G .

for each node v of T **do**

if v is a leaf **then** $S_v = \{(b, 1), (\emptyset, 0)\}$ where b is the singleton bag of v ($X_v = \{b\}$)
else $S_v = \{(\emptyset, 0)\}$

{Invariant: $(b, r) \in S_v$ means that there is an independent set I of size r with $X_v \cap I = b$ that contains only the vertices from the bags in the subtree rooted at v }

Traverse the tree T bottom-up

for a join node v of T with children u and w **do**
for each $(b_u, n_u) \in S_u$ and $(b_w, n_w) \in S_w$, where $b_u = b_w$ **do**
 $S_v = S_v \cup \{(b_u, n_u + n_w - |b_u|)\}$

for an introduce node v of T with one child u **do**

begin

$q \leftarrow$ the vertex in $X_v \setminus X_u$

for each $(b_u, n_u) \in S_u$ **do**

$S_v = S_v \cup \{(b_u, n_u)\}$

if $b_u \cup \{q\}$ is independent and $|b_u| + 1 \leq M$ **then**

$S_v = S_v \cup \{(b_u \cup \{q\}, n_u + 1)\}$

end

for a forget node v of T with one child u **do**

begin

$q \leftarrow$ the vertex in $X_u \setminus X_v$

for each $(b_u, n_u) \in S_u$ **do**

$S_v = S_v \cup \{(b_u \setminus \{q\}, n_u)\}$

end

Let r be the root of T .

Find a pair (b, m) in S_r that maximizes m .

By backtracking, produce an independent set in G on m vertices that includes b .

Theorem 1. *If T has width poly-logarithmic in n and size polynomial in n , and $M = O(\log n / \log \log n)$ then Algorithm 1 runs in polynomial time.*

Proof. Let ℓ be the width of T . The running time of the algorithm is equal to the running time required to traverse the entire nice tree-decomposition T , and thus, it is the sum of the running times the algorithm spends in each node of T . Since the number of nodes in the nice tree-decomposition in T is polynomial, it is enough to show that the time spent in each node is polynomial too. The time spent on a node v with two children u and w is $O(|S_u| \cdot |S_w|)$, and the time spent on a node v with one child u is $O(|S_u|)$. Therefore, to complete the proof, we must show that the size of each S_v is polynomial. This follows from the fact that each element in S_v is of the form (b, r) , with $0 \leq r \leq n$, $b \subseteq X_v$ with $|b| \leq M = O(\log n / \log \log n)$. Therefore, since $|X_v| \leq \ell \leq \log^{O(1)} n$, we have at most $(n+1) \cdot \binom{M}{\ell} \leq (n+1)\ell^M \leq n^{O(1)}$ sets b associated with S_v and hence the size of each S_v is polynomial in n . $\square$

Theorem 2. *Suppose that the nice tree-decomposition T of a partial k -tree has width ℓ and the size polynomial in n . Let ϵ be any positive constant satisfying $\ell \geq \log^{1-\epsilon} n$. There is a constant c such that if $M \geq \frac{c \log n}{\epsilon \log \log n}$ then Algorithm 1 returns an independent that is an $(\ell / \log^{1-\epsilon} n)$ -approximation of maximum independent set of G .*

Proof. We set the constant c such that T has size upper bounded by n^{c-1} .

Let I be a maximum independent set in G . Let $s = \log^{1-\epsilon} n$ and $r = \ell / \log^{1-\epsilon} n$. It is easy to verify that the invariant in Algorithm 1 is satisfied throughout the entire algorithm and therefore the output is a correct independent set. Furthermore, we observe that Algorithm 1 returns an independent set of G whose size is not less than that of any independent set in G whose intersection with any bag of T is of cardinality not exceeding M . Therefore, to conclude the proof we only must show the *existence* of an independent set whose size is at least $|I|/r$ and whose intersection with any bag of T has at most M vertices.

Let $\lambda = c/\epsilon$. Consider a randomized construction of a smaller independent set $I' \subseteq I$ that is obtained by taking each element from I to I' independently at random with probability $\frac{1}{r}$. We show first that with high probability the intersection of I' with any bag in the nice tree-decomposition T does not exceed M .

For any node v in the tree-decomposition T , let $\mathfrak{Y}_v$ be the random variable that denotes the size of the intersection of I' with the bag X_v of v . Let $q_v = |I \cap X_v|$. Clearly, the random variable $\mathfrak{Y}_v$ has the binomial distribution with parameters q_v and $\frac{1}{r}$. Therefore, $\mathbf{E}[\mathfrak{Y}_v] = \frac{q_v}{r}$. Since $q_v \leq |X_v| \leq \ell = r \cdot s$, we have $\mathbf{E}[\mathfrak{Y}_v] \leq s$. Furthermore, by the Chernoff bound (see, e.g., [10]), for every $t > \mathbf{E}[\mathfrak{Y}_v]$ we have $\Pr[\mathfrak{Y}_v \geq t] \leq (e \cdot \mathbf{E}[\mathfrak{Y}_v]/t)^t \leq (es/t)^t$. If we set $t = M = \lambda \log n / \log \log n$, then we obtain

$$\Pr[\mathfrak{Y}_v \geq M] \leq (es/M)^M = \left(\frac{es}{\lambda \log n / \log \log n} \right)^{\lambda \log n / \log \log n}$$

$$\begin{aligned}
&\leq \left(\frac{s}{\log n} \right)^{\lambda \log n / \log \log n} = \left(\log^{-c/\lambda} n \right)^{\lambda \log n / \log \log n} \\
&= \left(2^{-\frac{c}{\lambda} \cdot \log \log n} \right)^{\lambda \log n / \log \log n} = n^{-c}.
\end{aligned}$$

In the bound above we use the fact that $s = \log^{1-\epsilon} n = \log^{1-\frac{c}{\lambda}} n$.

Since the number of nodes in T is at most n^{c-1} , the union bound yields

$$\Pr[\exists v \mid \mathfrak{Y}_v \geq M] \leq \sum_v \Pr[\mathfrak{Y}_v \geq M] \leq n^{c-1} \cdot n^{-c} \leq n^{-1}.$$

Hence, the intersection of the independent set I' in G with every bag of T is of cardinality not exceeding M with high probability. Furthermore, with a constant probability $|I'| \geq |I|/r$. This implies that in our probabilistic experiment, with probability at least 0.1, the set I' is an independent set of size at least $|I|/r$ that intersects every bag of T in at most M elements. Since this probabilistic experiment makes the random choices at random independently of G and T , the claim above implies that there exists an independent set in G of size at least $|I|/r$ that intersects every bag of T in at most M elements. This yields an r -approximation of a maximum independent set of G . $\square$

Note that Algorithm 1 yields $\log^{1-\epsilon} n$ times better approximation than that implied by Lemma 1.

Theorem 3. *Let $\ell = O(\log^{O(1)} n)$ and let ϵ be any positive constant. For a partial k -tree given with its nice tree-decomposition of width ℓ having a polynomial size, there exists an $M = \Theta(\log n / \epsilon \log \log n)$ such that Algorithm 1 yields an $(\ell / \log^{1-\epsilon} n)$ -approximation of maximum independent set in polynomial time.*

4 Extensions to Other Problems

The deterministic approximation method for independent set presented in the previous section can be adapted to several other $\mathcal{NP}$ -hard optimization graph problems constrained to partial polylog-trees.

The following fact yields a straightforward simplification of the method for a maximum clique in partial polylog-trees.

Fact 6 *Let $(\{X_i \mid i \in I\}, T = (I, F))$ be a tree-decomposition of $G = (V, E)$. Suppose that $W \subseteq V$ forms a clique in G . There is $i \in I$ such that $W \subseteq X_i$. $\square$*

Theorem 4. *Let $\ell = O(\log^{O(1)} n)$. For a partial k -tree given with its tree-decomposition of width ℓ and of polynomial size, an $(\ell \log \log n / \log n)$ -approximation of maximum clique can be found in polynomial time.*

Proof. By Fact 6, a maximum clique is contained in one of the bags of the given decomposition tree. Therefore, by checking each subset of size $O(\log n / \log \log n)$ of each bag for whether or not it induces a clique, we can obtain the desired approximation. By straightforward calculations, the number of subsets to check is polynomial. $\square$

The problem of *maximum induced m -colorable subgraph* for a graph $G = (V, E)$ is to find a maximum cardinality subset of V that induces an m -colorable subgraph of G . The problem is known to be $\mathcal{NP}$ -hard already for $m = 2$ (see [13]).

Theorem 5. *Let $\ell = O(\log^{O(1)} n)$, $m \geq 2$ and let ϵ be an arbitrary small positive constant where $\ell \geq \log^{1-\epsilon} n$. For a partial k -tree given with its tree-decomposition of width ℓ and of polynomial size, the problem of maximum induced m -colorable subgraph admits an $(\ell/\log^{1-\epsilon} n)$ -approximation in polynomial-time.*

Proof. We use a method analogous to Algorithm 1. For the sake of explanation let us assume $m = 2$ first. The main difference now is that the classes S_v contain triplets (V', V'', n'') where:

- V', V'' are disjoint subsets of $O(\log n / \log \log n)$ vertices in the bag corresponding to v , and
- n'' is the maximum cardinality of the union $V_1 \cup V_2$ where V_1 and V_2 are disjoint subsets of vertices in the subgraph of the input graph induced by the bag corresponding to v and the the bags lying under it in the given tree-decomposition, $V' \subset V_1$ and $V'' \subset V_2$, and the subgraph of the input graph induced by $V_1 \cup V_2$ can be colored with two colors by coloring vertices in V_1 with the first color and vertices in V_2 with the second color.

Note that the number of such pairs V', V'' is polynomial since the bag is of size $O(\log^{O(1)} n)$. A straightforward modification of Algorithm 1 yields the thesis. We replace b_u with pairs b'_u, b''_u and the test of $b_u \cup \{q\}$ for independence with the test of whether or not the subgraph induced by $b'_u \cup \{q\} \cup b''_u$ can be colored with two colors such that vertices in $b'_u \cup \{q\}$ are colored with the first color and vertices in b''_u are colored with the second color, or vertices in b'_u are colored with first color and vertices $b''_u \cup \{q\}$ are colored with the second color, respectively. To prove the approximation factor of our method we proceed along the lines of the proof of Theorem 2. We leave the straightforward details and the straightforward generalization to an arbitrary $m = O(1)$ to the reader.

5 Randomized Approximation of Maximum Independent Set

One drawback of Theorem 3 is that it requires that the width ℓ is polylogarithmic. In this section we overcome this obstacle and describe a *randomized* algorithm that works for arbitrary $\ell = \Omega(\log n)$ and achieves a better approximation than in Theorem 3.

Theorem 6. *Let c be any positive constant. For a partial k -tree G on n vertices given with its tree-decomposition of width $\ell \geq \log n/c$ and of polynomial size, the problem of maximum independent set admits a randomized $(c\ell/\log n)$ -approximation algorithm running in polynomial time, with high probability.*

Proof. We assume, without loss of generality and for simplicity of presentation only, that $c\ell/\log n$ and $n \log n/(c\ell)$ are integers.

Let I be an arbitrary maximum independent set in G . Partition the vertices of G into $c\ell/\log n$ sets $V_1, \dots, V_{c\ell/\log n}$ independently and uniformly at random. (That is, for each vertex y of G and for each i , $\Pr[y \in V_i] = \log n/(c\ell)$.) For any i , $1 \leq i \leq c\ell/\log n$, let G_i be the subgraph of G induced by the vertex set V_i .

Let T be a tree-decomposition of G of width ℓ and let the size of T be $O(n^b)$ for a positive constant b . Let T_i be the tree-decomposition T constrained to the vertices in V_i . We show that for each i , the tree-decomposition T_i has width $O(\log n)$ with probability $1 - o(1)$. Indeed, let us first consider any node v in T and let us estimate the probability that the bag of v , X_v , intersects with more than $O(\log n)$ vertices in V_i . Since for any i , for any vertex $y \in X_v$, $\Pr[y \in V_i] = \log n/(c\ell)$, and since $|X_v| \leq \ell$, the size of $X_v \cap V_i$ is (stochastically) upper bounded by a random variable with the binomial distribution with the parameters k and $\log n/(c\ell)$, which we denote by $\mathbb{B}(\ell, \log n/(c\ell))$. That is, for any N , $\Pr[|X_v \cap V_i| \geq N] \leq \Pr[\mathbb{B}(\ell, \log n/(c\ell)) \geq N]$. On the other hand, by the Chernoff bound we have for $\lambda = 2e + c(b+1)$,

$$\begin{aligned} \Pr[\mathbb{B}(\ell, \log n/(c\ell)) \geq \frac{\lambda \log n}{c}] &\leq \left(\frac{e \log n/c}{\lambda \log n/c} \right)^{\lambda \log n/c} = (e/\lambda)^{\lambda \log n/c} \\ &\leq (1/2)^{c(b+1) \log n/c} = n^{-(b+1)}. \end{aligned}$$

Since T has $O(n^b)$ nodes and since there are $c\ell/\log n$ sets V_i , this implies that for each V_i , the tree T_i has width $O(\log n)$ with probability at least $1 - O(n^b) \cdot \frac{c\ell}{\log n} \cdot n^{-(b+1)} = 1 - o(1)$. Therefore, conditioned on that, for each i , $1 \leq i \leq c\ell/\log n$, we can find a maximum independent set in G_i by using the standard dynamic programming method on T_i [8]. By the pigeon hole principle, at least one of these maximum independent sets is of size not less than $|I| \cdot \log n/(c\ell)$, which yields the theorem. The failure probability of the randomized algorithm is $O(\frac{c\ell}{\log n}) = o(1)$. One can easily modify our algorithms and the arguments to amplify the failure probability to less than n^{-c} for any given constant $c > 0$. $\square$

6 Final Remarks

The $\mathcal{NP}$ -hard problem of finding a tree decomposition of a graph having minimum treewidth is known to admit logarithmic-factor approximation polynomial-time algorithms (see Fact 3 and 4). Merely, for graphs of constant treewidth, polynomial-time algorithms for tree decomposition of minimum width are known (see Fact 2). On the other, a tree decomposition of a graph can be transformed into its nice tree decomposition without increasing the width (see Fact 5). Therefore, for a partial k -tree on n vertices, in case a tree decomposition of width k is not given, all our upper bounds on approximation factors achievable in polynomial time established in the previous sections have to be increased by a logarithmic factor. Due to the improvement of Fact 3 to Fact 4, the logarithmic factor is of the form $O(\log k)$. Thus, in particular, if k is polylogarithmic in n , the increase is merely by an $O(\log \log n)$ factor.

References

1. M. Ajtai, J. Komlós, and E. Szemerédi. A note on Ramsey numbers. *Journal of Combinatorial Theory, Series A*, 29: 354–360, 1980.
2. E. Amir. Efficient approximation for triangulation of minimum treewidth. *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI'01)*, pages 7–15, University of Washington, Seattle, WA, USA, August 2–5, 2001.
3. S. Arnborg. Efficient algorithms for combinatorial problems on graphs with bounded decomposability — A survey. *BIT*, 25(1): 2–23, 1985.
4. S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2): 308–340, 1991.
5. H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11(1-2): 1–22, 1993.
6. H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6): 1305–1317, 1996.
7. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2): 1–45, 1998.
8. H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. Technical Report RUU-CS-87-22, Utrecht University 1987.
9. H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, and shortest elimination tree height. *Journal of Algorithms*, 18(2): 238–255, 1995.
10. H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23:493–507, 1952.
11. L. Engebretsen and J. Holmerin. Clique is hard to approximate within $n^{1-o(1)}$. In *Proc. 27th ICALP*, pages 2–12, 2000.
12. U. Feige. Approximating maximum clique by removing subgraphs. Manuscript, March 2002.
13. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York, NY, 1979.
14. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1): 105–142, 1999.
15. R. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, eds., *Complexity of Computer Computations*, pages 85–103, Plenum Press, New York, 1972.
16. S. Khot. Improved inapproximability results for MaxClique, chromatic number, and approximate graph coloring. In *Proc. 42nd FOCS*, pages 600–609, 2001.
17. T. Kloks. Treewidth: Computations and Approximations. *Lecture Notes in Computer Science* 842. Springer-Verlag, Heidelberg, 1994.
18. N. Robertson and P. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3): 309–322, 1986.
19. J.B. Shearer. A note on the independence number of triangle-free graphs. *Discrete Mathematics*, 46: 83–87, 1983.