Approximation algorithms for optimization problems in graphs with superlogarithmic treewidth

Artur Czumaj*† Magnús M. Halldórsson‡ Andrzej Lingas§

Johan Nilsson¶

Abstract

We present a generic scheme for approximating $\mathcal{NP}$-hard problems on graphs of treewidth $k = \omega(\log n)$. When a tree-decomposition of width ℓ is given, the scheme typically yields an $\ell/\lg n$-approximation factor; otherwise, an extra $\log k$ factor is incurred. Our method applies to several basic subgraph and partitioning problems, including the *maximum independent set* problem.

Keywords: Approximation algorithms, $\mathcal{NP}$-hard problems, partial k-trees, bounded treewidth

*Dept. Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA. czumaj@cis.njit.edu
†Research supported in part by NSF grants CCR-0105701 and CCR-0313219.
‡Dept. Computer Science, University of Iceland, Reykjavik, Iceland. mmh@hi.is
§**Contact author**. Dept. Computer Science, Lund University, 22100 Lund, Sweden. Andrzej.Lingas@cs.lth.se
¶Dept. Computer Science, Lund University, 22100 Lund, Sweden. f98jn@efd.lth.se

1 Introduction

One of the most successful parameterization of graphs is that of *treewidth*. While the formal definition is deferred to the next section, graphs of treewidth k, also known as *partial k-trees*, are graphs that admit a tree-like structure, known as their *tree-decomposition* of *width* k.

A wide variety of $\mathcal{NP}$-hard graph problems have been shown to be solvable in polynomial time, or even linear time, when constrained to partial k-trees [2, 12]. For some of these problems polynomial time solutions are possible for graphs of treewidth $O(\log n)$ or $O(\log n/\log\log n)$ [12].

A standard example of a problem solvable in graphs of treewidth $O(\log n)$ is the maximum independent set (MIS) problem [2], which is that of finding a maximum collection of pairwise non-adjacent vertices. In the weighted version of the problem, vertices are given with weights and we seek an independent set of maximum total weight. For general graphs, the best polynomial-time approximation ratios known for MIS is $n\,(\log\log n)^2/\log^3 n$ [6]. On the other hand, it is known that unless $\mathcal{NP} \subseteq$ ZPTIME$(2^{(\log n)^{O(1)}})$, no polynomial-time algorithm can achieve an approximation guarantee of $n^{1-O(1/(\log n)^{\gamma})}$ for some constant γ [9].

In this paper, we investigate the approximability status of some of the aforementioned $\mathcal{NP}$-hard problems, where our main interest is in graphs of treewidth $k = \omega(\log n)$. We focus our study on MIS, deriving further applications of our method by extensions of that given for MIS.

Better approximation bounds for MIS are achievable for special classes of graphs. For the purposes of this paper, a class that properly contains partial k-trees is that of *k-inductive graphs*. A graph is said to be k-inductive if there is an ordering of its vertices so that each vertex has at most k higher-numbered neighbors. If such an ordering exists, it can be found by iteratively choosing and removing a vertex of minimum degree in the remaining graph. ¿From this definition, it is clear that k-trees, and thus also partial k-trees, are k-inductive. A k-inductive graph is easily $k+1$-colored by processing the vertices in their reverse inductive order, assigning each vertex one of the colors not used by its at most k previously colored neighbors. This implies that the largest weight color class approximates the weighted MIS within a factor of $k + 1$. The best approximation known for MIS (and weighted MIS) in k-inductive graphs is $O(k \log \log k / \log k)$ [8].

1.1 New contribution

We present a novel generic scheme for approximation algorithms for maximum independent set and other $\mathcal{NP}$-hard graph optimization problems constrained to graphs of treewidth $k = \Omega(\log n)$. Our scheme leads to deterministic polynomial-time algorithms that typically achieve an approximation ratio of $\ell / \log n$ for $\ell = \Omega(\log n)$ when a tree-decomposition of width ℓ is given.

Our scheme can be applied to any problem of finding a maximum induced subgraph with hereditary property Π and any problem of finding a minimum partition into induced subgraphs with hereditary property Π provided that for graphs with given tree-decomposition of logarithmic or near logarithmic width can be solved exactly in polynomial time. All these approximation factors achievable in polynomial time are the best known for the aforementioned problems for graphs of superlogarithmic treewidth.

In case a tree-decomposition of width $\ell = k$ is not given, the approximation achieved by our method increases by a factor of $O(\log k)$.

2 Preliminaries

The notion of *treewidth* of a graph was originally introduced by Robertson and Seymour [11] in their seminal graph minors project. It has turned out to be equivalent to several other interesting graph theoretic notions, e.g., that of partial k-trees.

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 three conditions are fulfilled: (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* 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.*

It is well known that a graph G is a partial k-tree iff the treewidth of G is at most k [2]. For a graph with n vertices and treewidth k, a tree decomposition of width k can be found in time $O(n\, 2^{O(k^3)})$ [5], whereas a tree decomposition of width $O(k \log k)$ and size $O(n)$ can be found in time polynomial in n [1].

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

Definition 2 *A tree-decomposition* $T = (I,F)$ *of a graph* G *is* nice *if (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 3 [10] *A tree-decomposition* $T = (I,F)$ *of a graph* G *can be transformed into a nice tree-decomposition in time polynomial in* $|I|$ *and the size of* T*, without increasing its width. The size of the resulting nice decomposition is* $O(\ell \cdot |I|)$*, where* ℓ *is the width of* T*.*

3 Approximation of maximum independent set

In this section we present a deterministic approximation algorithm for finding maximum independent set in graphs with given tree-decomposition of width ℓ. We begin with the following general partition lemma of independent interest.

Lemma 4 *Let t be a positive integer and let G be a graph given with a tree decomposition T of width ℓ. Then, the vertex set of G can be partitioned in polynomial time into classes $V_1, \ldots, V_{\lceil \ell/t \rceil}$ so that each bag of T contains at most t vertices from each class.*

Proof. By Fact 3, we may assume w.l.o.g. that the given tree decomposition T is nice.

To obtain the intended partition, we proceed top-down, arbitrarily assigning the vertices in the root-bag of T into classes with at most t vertices each. Inductively, for a node v with a child u in T, the partition of the bag of u is consistent with that of the bag of v and the upper bound of t on the size of each class V_i within each bag. Namely, if v is a join node, the bag of u gets the same partition as that of v; if v is an introduce node, the partition is also the same (with one fewer vertices); if v is a forget node, then the additional vertex in u is placed into some class that has fewer than t vertices from the bag of u. $\square$

Lemma 4 yields the aforementioned approximation algorithm for maximum independent set.

Theorem 5 *Let c be a positive constant. For a graph G on n vertices given with its tree-decomposition of width $\ell \geq c \log n$ and of polynomial size, the maximum weighted independent set problem admits a $\lceil \ell/(c \log n) \rceil$-approximation algorithm running in polynomial time.*

Proof. Apply Lemma 4 to T with $t = c \log n$ to obtain a vertex partition $V_1, \ldots, V_{\lceil \ell/(c \log n) \rceil}$.

For any i, $1 \leq i \leq \lceil \ell/(c \log n) \rceil$, let G_i be the subgraph of G induced by the vertex set V_i and let T_i be the tree-decomposition of G_i obtained by constraining the bags of T to the vertices in V_i. By the properties of the classes V_i, each T_i has width $\log n$.

For each i, $1 \leq i \leq \lceil \ell/(c \log n) \rceil$, we can find a maximum independent set in G_i by using the standard dynamic programming method on T_i [3]. By the pigeon hole principle, at least one of these maximum weight independent sets is of weight not less than that of a maximum weight independent set of G divided by $\lceil \ell/(c \log n) \rceil$, which yields the theorem. $\square$

4 Extensions of the approximation method

We can generalize Theorem 5 to include the problem of *maximum weight induced subgraph with hereditary property* Π provided that the problem constrained to graphs of treewidth $O(\log n)$ can be solved exactly in polynomial time. For a graph with vertex weights, the problem of maximum weight induced subgraph with property Π is to find a maximum weight subset of vertices of the input graph which induces a subgraph having the property Π. If Π holds for arbitrarily large graphs, does not hold for all graphs, and is *hereditary* (holds for all induced subgraphs of a graph whenever it holds for the graph) then the problem of finding a maximum weight induced subgraph with the property Π is $\mathcal{NP}$-hard (see GT21 in [7]). Examples of such properties Π are "being an independent set," "being m-colorable," and "being a planar graph."

Theorem 6 *For a graph G given with tree-decomposition of width $\ell \geq t > 0$ and of polynomial size, the problem of finding a maximum weight induced subgraph with hereditary property Π admits a $\lceil \ell/t \rceil$-approximation, provided that for graphs of treewidth $\ell = t$ the problem can be solved exactly in polynomial time.*

Proof. Replace the solution of the weighted MIS problem on G_i in the proof of Theorem 5 with the solution of the problem of maximum weight induced subgraph with property Π on G_i, and the theorem follows. $\square$

Since for every clique W in a graph G and every tree decomposition $(\{X_i | i \in I\},\ T)$ of G, there is an $i \in I$ with $W \subseteq X_i$, it suffices to check each subset in each bag of the given tree decomposition in order to find a maximum weight clique. Hence, we can approximate the clique problem by applying any clique-approximation algorithm on the graphs induced by each X_i. Combined with Theorem 6, we obtain the following corollary.

Corollary 7 *Let c be any positive constant. For a graph G on n vertices given with its tree-decomposition of width $\ell \geq c \log n$ and of polynomial size, the problem of maximum weighted clique admits a $\min(\ell/(c \log n), \ell(\log\log\ell)^2/\log^3\ell)$-approximation.*

The problem of *minimum partition into induced subgraphs with property* Π is to find a minimum cardinality partition of vertices of the input graph into subsets inducing subgraphs having the property Π. E.g., if Π is "being independent," we get the minimum coloring problem.

Theorem 8 *For a graph G on n vertices given with its tree-decomposition of width $\ell \geq t > 0$ and of polynomial size, the problem of minimum partition into induced subgraphs with hereditary property Π admits a $\lceil \ell/t \rceil$-approximation polynomial-time algorithm, provided that for $\ell = t$ the problem can be solved exactly in polynomial time.*

Proof. Produce the subgraphs G_i, $1 \leq i \leq \lceil \ell/\log n \rceil$, as in the proof of Theorem 5. For each G_i find a minimum number partition P_i into induced subgraphs with hereditary property Π and output the union of P_i as the approximate solution. □

Since the minimum vertex coloring problem can be solved exactly in polynomial time for graphs with given tree-decomposition of width $O(\log n/\log\log n)$ [12], we obtain the following.

Corollary 9 *Let c be any positive constant. For a graph G on n vertices given with its tree-decomposition of width $\ell \geq c\log n/\log\log n$ and of polynomial size, the minimum vertex coloring problem admits $\lceil \ell \log\log n/(c\log n) \rceil$-approximation.*

Since a tree decomposition of width $O(k\log k)$ and size $O(n)$ can be found in time polynomial in n [1], we obtain the following variants of Theorems 6 and 8 for graphs of treewidth k.

Theorem 10 *Let $k \geq t > 0$ and let G be a graph with treewidth k. The problems of maximum weight induced subgraph with hereditary property Π and the problems of minimum partition into induced subgraphs with hereditary property Π admit $\log k \lceil k/t \rceil$-approximation algorithm running in polynomial time provided that for a graph of treewidth t they can be solved in polynomial time.*

In [12], classes of vertex partitioning problems that can be solved in polynomial time on graphs of $O(\log n)$ or $O(\log n / \log\log n)$ treewidth are given. Thus, for problems in these classes, Theorems 6, 8, and 10 can be used.

References

[1] E. Amir. Efficient approximation for triangulation of minimum treewidth. *Proc. 17th Conference in Uncertainty in Artificial Intelligence (UAI'01)*, pages 7–15, 2001.

[2] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[3] H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. Technical Report RUU-CS-87-22, Utrecht University, 1987.

[4] H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernet.*, 11(1-2):1–22, 1993.

[5] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[6] U. Feige. Approximating maximum clique by removing subgraphs. Manuscript, 2002.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York, 1979.

[8] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms and Applications*, 4(1):1–16, 2000.

[9] S. Khot. Improved inapproximability results for MaxClique, chromatic number, and approximate graph coloring. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 600–609, 2001.

[10] T. Kloks. *Treewidth: Computations and Approximations*. Volume 842 of *Lecture Notes in Computer Science*. Springer-Verlag, Heidelberg, 1994.

[11] N. Robertson and P. Seymour. Graph minors II: Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[12] J. A. Telle and A. Proskurowski. *Algorithms for Vertex Partitioning Problems on Partial k-trees*. *SIAM J. Discrete Math.* Vol. 10, No 4, pages 529-550.