ELSEVIER

Information Processing Letters 61 (1997) 227-232

Information Processing Letters

Fast partitioning l-apex graphs with applications to approximating maximum induced-subgraph problems

Dimitrios M. Thilikos *,[1], Hans L. Bodlaender

Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands

Received 23 July 1996; revised 15 January 1997
Communicated by T. Lengauer

Abstract

A graph is l-apex if it can be made planar by removing at most l vertices. In this paper we show that the vertex set of any graph not containing an l-apex graph as a minor can be partitioned in linear time into 2^l sets inducing graphs with small treewidth. As a consequence, several maximum induced-subgraph problems when restricted to graph classes not containing some special l-apex graphs as minors, have practical approximation algorithms. © 1997 Elsevier Science B.V.

Keywords: Algorithms; Analysis of algorithms; Approximation algorithms; Combinatorial problems; Graph minors; Treewidth

1. Introduction

Much work in algorithmic graph theory has been done in finding polynomial approximation algorithms (or even NC algorithms) for NP-complete graph problems when restricted to special classes of graphs. A wide class of such problems is defined in terms of hereditary properties (a graph property π is called *hereditary* when, if π is satisfied for some graph G, then π is also satisfied for all induced subgraphs of G). The *maximum induced subgraph problem* for hereditary property π is the following problem: Given a graph $G = (V, E)$, find a maximum subset of V that induces a subgraph satisfying π. We call this problem MISP(π). A wide range of this type of problems has been shown to be NP-complete by Yannakakis [17]. There is a long series of results concerning fast approximation algorithms (serial or parallel) for such problems. An algorithm that, given an instance of MISP(π), always returns a solution that is of size at least a constant factor $1/\alpha$, is called *an approximation algorithm* for MISP(π) with *performance ratio* α. Also, MISP(π) has a *polynomial-time approximation scheme* (PTAS) if, for any fixed $\varepsilon > 0$, there exists a polynomial approximation algorithm with performance ratio $1 + \varepsilon$. Lipton and Tarjan [12] proved that various MISP(π)'s have a PTAS when their instances are restricted to classes of planar graphs. This result has been considerably generalised to any class of graphs with an excluded minor by Alon et al. [1]. Unfortunately, these schemes appear to have only theoretical interest as their running time is a highly exponential function of $1/\varepsilon$ (see [10]). Consequently, the following question appears: for which graph classes do there exist *practical* approximation algorithms for MISP(π)'s? In this direction, Baker [3] gave a prac-

* Corresponding author. E-mail: {sedthilk,hansb}@cs.ruu.nl

[1] This author was supported by the Training and Mobility of Researchers (TMR) Program under EU Contract No. ERBFMBICT950198.

PII S0020-0190(97)00024-0

tical PTAS for several MISP(π) when the input instances are planar. Chen [18] gave a nontrivial generalisation of Baker's result for the classes of graphs that do not contain $K_{3,3}$ (or K_5) as a minor.

In this paper we examine the practical approximability of several MISP(π)'s on some more general classes of graphs. We call a graph H *l-apex* of a planar graph H_0 if it contains at most l vertices whose removal produces H_0. Let $G = (V, E)$ be an H-minor free graph, where H is an l-apex of some planar graph H_0. In Section 3, we give a linear and easy to implement algorithm that outputs a partition of V into 2^l sets, each inducing a graph of bounded treewidth (intuitively, graphs of bounded treewidth are graphs that can be constructed by piecing together graphs of bounded size in a tree-like fashion). Using the fact that a wide range of MISP(π)'s restricted to graphs with bounded treewidth can be solved by linear time algorithms, we can obtain approximation algorithms for these MISP(π)'s with performance ratio 2^l. In Section 4 we describe several l-apex graphs that, when excluded, our approach leads to practical approximation algorithms. Moreover, some interesting corollaries of our results are discussed.

2. Definitions

We consider undirected graphs without multiple edges or self-loops. Given a graph $G = (V, E)$ we denote its vertex set and edge set with $V(G)$ and $E(G)$ respectively. Given two graphs G, H we say that H is a minor of G if H can be obtained by a series of vertex deletions, edge deletions and/or edge contractions (a contraction of an edge $\{u, v\}$ in G is the operation that replaces u and v by a new vertex whose neighbours are the vertices that where adjacent to u and/or v). G is *H-minor free* if G has no minor isomorphic to H. A graph class containing only H-minor free graphs is called *H-minor free*. If $V' \subseteq V(G)$, we call the graph $(V', \{\{v, u\} \in E(G) \mid v, u \in V'\})$ *the subgraph of G induced by* V' and we denote it as $G[V']$.

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 is a tree, such that

- $\bigcup_{i \in I} X_i = V(G)$,
- for each edge $\{v, w\} \in E$, there is an $i \in I$ such that $v, w \in X_i$,
- for each $v \in V$, the set of nodes $\{i \in I \mid v \in X_i\}$ induces a subtree of T.

The width of a tree decomposition $(\{X_i \mid i \in I\}, T = (I, F))$ equals $\max_{i \in I}(|X_i| - 1)$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G.

Robertson and Seymour [14] proved (see also [15]) that for any planar graph H_0 there exists a constant c_H such that any H_0-minor free graph has treewidth at most c_H. Given a planar graph H, we define the *minimum excluding bound* of H, med(H), as the maximum treewidth over all H-minor free graphs.

In [15], it was shown that for all planar graphs H,

$$\operatorname{med}(H) \leqslant 20^{2(2|V(H)|+4|E(H)|)^5}.$$

There are several classes where smaller upper bounds for the minimum excluding bound have been found. Examples of such classes are forests with at most r vertices ($\leqslant r - 2$, see [4]), minors of r-disjoint triangles ($\leqslant 12r^2 - 27r + 6$, see [6]), graphs that are minors of a $2 \times q$ grid and an r-circus graph ($\leqslant 2(q-1)^2(r-1)+1$, see [5]), cycles of length a most r ($\leqslant r - 2$, see [11]) and minors of $K_{2,r}$ ($\leqslant 2r - 2$, see [8]). It is interesting to mention that, according to the results in [4,6,5,11,8], there exist algorithms that, given a graph G with an excluded minor belonging into one of the aforementioned classes, output the corresponding small width tree decompositions in time linear on $|V(G)|$ and polynomial on r and q. This means that when the size of the excluded graphs is small, there are really practical algorithms to build the corresponding small width tree decompositions. We call such classes of graphs, where a tree decomposition of small width can computed, not only linearly, but also fast in practice, *quickly and fast* excluded.

Given a planar graph H_0 we define the *l-apex extension* of H_0, $\mathcal{H}_l$, as the class of graphs containing a set of at most l vertices whose removal produces H_0, i.e.,

$$\mathcal{H}_l = \{G \mid \exists S \subseteq V(G),\ |S| \leqslant l$$
$$G[V(G) - S] \text{ is isomorphic to } H_0\}.$$

Given a class $\mathcal{H}_0$ of planar graphs, we define the *l-apex extension* of $\mathcal{H}_0$ as the union of all the l-apex extensions of the graphs in $\mathcal{H}_0$. We call a graph H l-apex if it is contained in an l-apex extension of some planar graph H_0 (we call such a planar graph an *l-apex root* of H).

A (β,γ)-partition of a graph is a partition $\{V_1,\ldots,V_\beta\}$ of its vertices such that

$$\text{treewidth}(G[V_i]) \leqslant \gamma, \quad 1 \leqslant i \leqslant \beta.$$

3. The splitting algorithm

The main result of this paper is the following. Note that H and H_0 are not necessarily required as part of the input for the algorithm.

Lemma 1. *There exists an algorithm that, when given a graph $G = (V, E)$ and an integer l, such that G is an H-minor free graph for a graph H that is an l-apex of a planar graph H_0, outputs an $(2^l, \text{med}(H_0))$-partition of G in $O(l(|V(G)| + |E(G)|))$ time.*

Proof. First, we remark that we may assume that $2^l \leqslant |V(G)| \leqslant |E(G)|$: if not, then we just output the partition with one vertex per set. Now we claim that the required partition can be computed by algorithm l-SPLIT(G) shown in Fig. 1.

Algorithm l-SPLIT(G) refines a partition of G l times; each time each set is (possibly) split in two. Thus, it outputs at most 2^l sets. Each split is done in the following way (per connected component of the induced graph): a breadth first search is done from some arbitrary vertex v_0^i; in one set, we put all vertices with an even distance to v_0^i, and in the other set, we put all vertices with odd distance to v_0^i.

As H is an l-apex of H_0, there exists a set

$$S_{\text{add}} = \{s_1,\ldots,s_\mu\} \subseteq V(H), \quad \mu \leqslant l,$$

such that H_0 is isomorphic to $G[V(H) - S_{\text{add}}]$. We set $H_i = (V(H_0) \cup \{s_1,\ldots,s_i\},\ E(H_0) \cup \{\{v,u\} \mid v \in V(H_0) \cup \{s_1,\ldots,s_i\},\ u \in \{s_1,\ldots,s_i\}\}), 1 \leqslant i \leqslant \mu$.

Claim. *Let each connected component of $G[V_{1+n}]$ be H_s-minor free, where*

$$n = (m-1)2^{h-1}, \quad 1 \leqslant h \leqslant s, \ 1 \leqslant m \leqslant 2^{h-1}.$$

Then, after the execution of Procedure SPLIT(m,h), *each connected component of the graphs $G[V_{1+n}]$, $G[V_{1+n+2^{l-h}}]$ is H_{s-1}-minor free.*

```
Algorithm $l$-SPLIT$(G)$
Input: An $H$-minor free graph $G$, where $H$ is
       an $l$-apex extension of a planar graph $H_0$.
Output: A $(l, \text{med}(H_0))$-partition $\{V_1,\ldots,V_{2^l}\}$ of $G$.
1   begin
2   set $V_1 = V(G)$, $V_i = \emptyset, 2 \leqslant i \leqslant 2^l$.
3   for $h = 1,\ldots,l$ do
4     for $m = 1$ to $2^{h-1}$ call procedure SPLIT$(m,h)$
5   output $\{V_1,\ldots,V_{2^l}\}$
6   end

Procedure SPLIT$(m,h)$
1   begin
2   set $n = (m-1)2^{h-1}$.
3   Let $G^1,\ldots,G^\sigma$ be the connected components of $G[V_{1+n}]$.
4   for each connected component $G^i, i = 1,\ldots,\sigma$ do
5     begin
6     Choose arbitrarily a vertex $v_0^i \in V(G^i)$.
7     Compute a partition $V_0^i, V_1^i,\ldots,V_{t_i}^i$ of $V(G^i)$ such that
      $V_0^i = \{v_0^i\}$ and any set $V_j^i, 1 \leqslant t_i$, contains the vertices
      of $V(G^i)$ whose distance from $v_0^i$ is exactly $j$.
8     Set $V_{1+n+2^{l-h}} \leftarrow V_{1+n+2^{l-h}} \cup (\bigcup_{j=0,\ldots,\lfloor t_i/2 \rfloor} V_{2j}^i)$
9     end
10  Set $V_{1+n} \leftarrow V_{1+n} - V_{1+n+2^{l-h}}$
11  end
```

Fig. 1. Algorithm l-SPLIT(G).

Proof of Claim. Suppose not. As any connected component is entirely inside one of the sets V_j^i, as computed in Step 7, we can obtain H_s as a minor of G^i: contract all vertices in $V_0^i \cup \cdots \cup V_{j-1}^i$ to one vertex (v_0^i), contract (and remove) the edges (and vertices) in V_j^i to H_{s-1}, and remove all other vertices. As each vertex in V_q^i is adjacent to at least one vertex in V_{q-1}^i for any $q = 2,\ldots,j$, we now have a graph, obtained by adding a vertex to H_{s-1} that is adjacent to all vertices in H_{s-1}: this graph is isomorphic to H_s. A contradiction. □

Using inductively the Claim above, we can conclude that each connected component of each set in the partition, outputted by algorithm l-SPLIT is H_0-minor free and hence has treewidth at most $\text{med}(H_0)$. As the treewidth of a graph is the maximum of the treewidth of its connected components, it follows that the output is a $(2^l, \text{med}(H_0))$-partition of G.

Implementing Step 7 of Procedure SPLIT by the standard breadth first search algorithm, it directly follows that the algorithm uses $O(l(|V(G)| + |E(G)|))$ time. □

We mention that any H-minor free graph G is a sparse graph (i.e. $|E(G)| \leqslant c_H|V(G)|$ for some constant c_H). According to a result of Mader [13], $c_H \leqslant 2^{|V(H)|-3}$ (see [9]) and thus, we can conclude that the time complexity of l-SPLIT(G) is $O(2^{|V(H)|-3}l|V(G)|)$.

4. Conclusions

For a planar graph H_0 and a hereditary property π such that MISP(π) can be solved in linear time when restricted to graphs with bounded treewidth, we let $p_{H_0,\pi}$ be the value, such that MISP(π) can be solved in $\leqslant p_{H_0,\pi} \cdot n$ time when restricted on H_0-minor free graphs with n vertices (as we have already mentioned, such graphs have bounded treewidth, so this value does exist).

Theorem 1. *Let π be a hereditary property. Let $H \in \mathcal{H}$ where $\mathcal{H}$ is an l-apex extension of some planar graph H_0. Then, there exists an approximation algorithm for* MISP(π) *on H-minor free graphs G with performance ratio 2^l, and with running time*

$$\leqslant p_{H_0,\pi} \cdot |V_G| + c\,l(|V(G)| + |E(G)|),$$

where c is a constant not depending on π, l.

Proof. We apply the following steps.

(i) Using SPLIT(G), we find a $(2^l, \mathrm{med}(H_0))$-partition $\{V_1, \ldots, V_{2^l}\}$ of G.

(ii) We find a maximum subset W_i of V_i such that $G[W_i]$ satisfies π, $i = 1, \ldots, 2^l$.

(iii) We output the maximum cardinality set in $\{W_1, \ldots, W_{2^l}\}$. We denote this set as W_{aprx}.

Let W be a solution of MISP(π) and V^* a set in the partition such that $\forall i, 1 \leqslant i \leqslant 2^l$: $|W \cap V^*| \geqslant |W \cap V_i|$. Clearly, $|W \cup V^*| \geqslant |W|/2^l$. Notice that, as π is hereditary, $G[W \cup V^*]$ satisfies π and thus $|W \cap V^*| \leqslant |W_{\mathrm{aprx}}|$. It follows that $|W_{\mathrm{aprx}}| \geqslant |W|/2^l$, thus the performance ratio of the algorithm is 2^l. By Lemma 1 step (i) can be done in $O(l(|V(G)|+|E(G)|))$ time. Also, step (ii) can be done in $\leqslant p_{H_0,\pi}|V(G)|$ time. □

The term $|E(G)|$ can be replaced by a factor $2^{|V(H)|-3}|V(G)|$, by the result of Mader [13], discussed above.

Theorem 1 leads to practical approximation algorithms when $p_{H_0,\pi}$ is a relatively small constant. As in many cases, given a tree decomposition of G with width $\leqslant k$ the time to solve MISP(π) is $O(2^k n)$, an important bottleneck will often be the time needed to construct such a decomposition. Therefore, the size of $p_{H_0,\pi}$ depends heavily on the existence of fast algorithms that, given a graph with a planar graph as an excluded minor, output a tree decomposition with relatively small treewidth. Consequently, we conclude, that $p_{H_0,\pi}$ is often practically small when H_0 is quickly and fast excluded.

We mention that any new result characterising some planar graph as quickly and fast excluded will extent further the collection of graph classes where Theorem 1 leads to practical approximation algorithms.

Corollary 1. *For any hereditary property π such that* MISP(π) *can be solved in $\leqslant t(i)n$ time in graphs with treewidth at most $i = 1, 2$, there exists an*

$$O(t(i)n + (|V(H)| - 2 - i)(|V(G)| + |E(G)|))$$

time approximation algorithm for MISP(π) *on H-minor free graphs G, with performance ratio $2^{|V(H)|-2-i}$.*

Proof. It is enough to observe that if we apply l-SPLIT(G) where $l = |V(H)| - 2 - i$, $i = 1, 2$, the output will be a partition of sets inducing forests (in case $i = 1$) or graphs with treewidth $\leqslant 2$ (in case $i = 2$). □

In fact, we can obtain somewhat better approximation ratios than Corollary 1. If we run algorithm l-SPLIT(G) for $l = |V(H)| - 5$ and with input an H-minor free graph G, we can easily see that the output is a partition $\{V_1, \ldots, V_{2^{|V(H)|-5}}\}$ of $V(G)$ such that $G[V_i]$ is a K_5 free graph for $i = 1, \ldots, 2^{|V(H)|-5}$. Using now the practical PTAS of Chen [18] we can easily conclude that, for several hereditary properties π, given a H-minor free graph G and some $\varepsilon > 0$, there exists a practical approximation algorithm for MISP(π) with performance ratio $2^{|V(H)|-5} + \varepsilon$ (e.g. for K_6-minor free graphs the performance ratio is $2 + \varepsilon$)). We also have the following corollary.

Corollary 2. *Let π be a hereditary property such that* MISP(π) *can be solved in $\leqslant t(r)n$ time for n vertex*

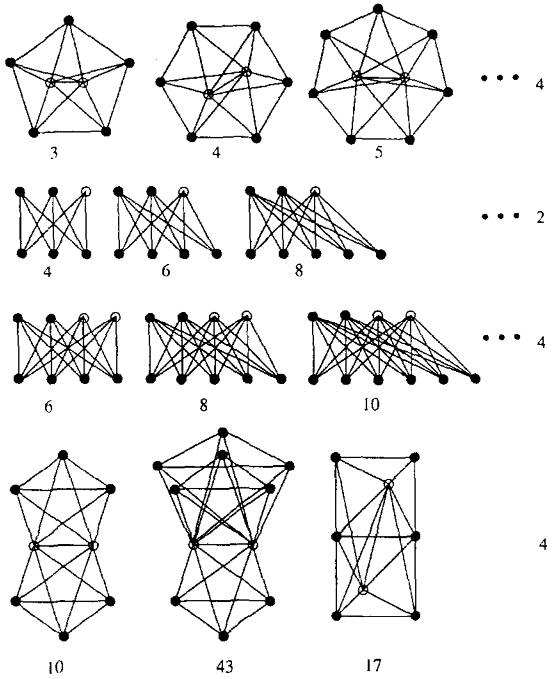

Fig. 2. Examples of some l-apex graphs where $l = 1, 2$. The l-root of each graph is the one induced by the dark vertices. The number below each graph is an upper bound to the minimum excluding bound of its l-apex root.

graphs given with a tree decomposition of width $\leqslant r$. Then, there exists an

$$O(t(2k_2 - 2)|V(G)| \\ + (k_1 - 2)(|V(G)| + |E(G)|))$$

time approximation algorithm with performance ratio 2^{k_1-2} for MISP(π) *on graphs G that are H-minor free, where H is a bipartite graph that is a subgraph of K_{k_1,k_2}, $k_1 \leqslant k_2$.*

Proof. Observe that K_{k_1,k_2}, and hence H is an (k_1-2)-apex extension of K_{2,k_2}. Further we use that, given a $K_{2,r}$-minor free graph G one can find a tree decomposition of G with width $\leqslant 2r-2$ in $O(r|V(G)|)$ time (see [8]). The result now follows from discussions above. □

Some examples of 1- or 2-apexes of quickly and fast excluded graphs along with the performance ratio of the corresponding approximation algorithms are shown in Fig. 2.

Acknowledgements

We thank Babette de Fluiter and Koichi Yamazaki for discussions on this research.

References

[1] N. Alon, P.D. Seymour and R. Thomas, A separator theorem for graphs with an excluded minor and applications, in: *Proceedings 22nd Annual Symposium on Theory of Computing* (ACM Press, New York, 1990); to appear in *J. Amer. Math. Soc.*

[2] S. Arnborg and A. Proskurowski, Characterization and recognition of partial 3-trees, *SIAM J. Algebraic Discrete Methods* 7 (1986) 305-314.

[3] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* 41 (1994) 153-180.

[4] D. Bienstock, N. Robertson, P.D. Seymour and R. Thomas, Quickly excluding a forest, *J. Combin. Theory Ser. B* 52 (1991) 274-283.

[5] H.L. Bodlaender, On linear time minor tests with depth first search, *J. Algorithms* 14 (1993) 1-23.

[6] H.L. Bodlaender, On disjoint cycles, *Internat. J. Found. Computer Science* 5(1) (1994) 59-68.

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

[8] H.L. Bodlaender, R.B. Tan, D.M. Thilikos and J. van Leeuwen, On interval routing schemes and treewidth, in: M. Nagl, ed., *Proceedings 21st International Workshop on Graph Theoretic Concepts in Computer Science* (WG'95), Lecture Notes in Computer Science 1017 (Springer, Berlin, 1995) 181-186.

[9] B. Bollobás, *Extremal Graph Theory with Emphasis on Probabilstic Methods*, Regional Conference Series in Mathematics (American Mathematical Society, Providence, RI, 1986).

[10] N. Chiba, T. Nishizeki and N. Saito, An approximation algorithm for the maximum independent set problem on planar graphs, *SIAM J. Comput.* 11 (1982) 663-675.

[11] M.R. Fellows and M.A. Langston, On search, decision and the efficiency of polynomial-time algorithms, in: *Proceedings 21st Annual Symposium on Theory of Computing* (ACM Press, New York, 1989) 501-512.

[12] R.J. Lipton and R.E. Tarjan, Applications of a planar separator theorem, *SIAM J. Comput.* 9 (1980) 615-627.

[13] W. Mader, Homomorphieeigenshaften und mittlere Kantendichte von Graphen, *Math. Ann.* 174 (1967) 265-268.

[14] N. Robertson and P.D. Seymour, Graph minors. V. Excluding a planar graph, *J. Combin. Theory Ser. B* 41 (1986) 92-114.

[15] N. Robertson, P.D. Seymour and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Ser. B* 62 (1994) 323-348.

[16] D.P. Sanders, On linear recognition of tree-width at most four, *SIAM J. Algebraic Discrete Methods* 19 (1996) 101-119.

[17] M. Yannakakis, Node- and edge-deletion NP-complete problems, in: *Proceedings 10th Annual Symposium on Theory of Computing* (ACM Press, New York, 1978) 253-264.

[18] Zhi-Zhong Chen, Practical approximation schemes for maximum induced-subgraph, in: F. Meyer auf der Heide and B. Monien, eds., *Proceedings 23rd International Colloquium on Automata, Languages, and Programming* (ICALP '96), Lecture Notes in Computer Science 1099 (Springer, Berlin, 1996) 268–279.