Linear-Time Algorithms for Problems on Planar Graphs of Fixed Disk Dimension

FAISAL N. ABU-KHZAM[1] , MICHAEL A. LANGSTON[2]

ABSTRACT. The disk dimension of a planar graph G is the least number k for which G embeds in the plane minus k open disks, with every vertex on the boundary of some disk. Useful properties of graphs with a given disk dimension are derived, leading to an efficient algorithm to obtain an outerplanar subgraph of a graph of disk dimension k by removing at most $2k-2$ vertices. This reduction is used to obtain linear-time exact and approximation algorithms for problems on graphs of fixed disk dimension. In particular, a linear-time 3-approximation algorithm is presented for the pathwidth problem on graphs of fixed disk dimension. This approximation ratio was previously known only for outerplanar graphs (graphs of disk dimension one).

1 Introduction

Disk dimension was introduced in [6], and can be defined as follows. Let G denote a finite, simple planar graph. The disk dimension of G is the least positive integer k for which G embeds in the plane minus k open disks, with every vertex of G lying on the boundary of some disk. Deciding whether G has disk dimension k is $\mathcal{NP}$-complete when k is part of the input [2, 5], but solvable in quadratic-time for every fixed value of k [6].

Our focus here is not, however, on solving the disk dimension problem itself. Instead, we restrict our attention to graphs of bounded disk dimension and show that they possess several useful properties. For example, we prove that graphs of disk dimension k are within $2k-2$ vertices of being outerplanar. We also employ this fact to derive exact and approximation algorithms for a variety of problems on such graphs.

[1]Research supported by the Lebanese American University under grant URC-c2004-63.

[2]Research supported by the U.S. National Science Foundation under grants CCR–0075792 and CCR–0311500, by the U.S. Office of Naval Research under grant N00014–01–1–0608, and by the U.S. Department of Energy under contract DE–AC05–00OR22725.

2 Background and Definitions

Let $dd(G)$ denote the disk dimension of G. If $k \geq dd(G)$, then G has a planar embedding minus k open disks such that every vertex of G lies on the boundary of some disk. We call this a "k dimensional embedding of G," written $E_k(G)$. We use d_i to denote disk i, c_i to denote the center of disk i, and b_i to denote the boundary of disk i. The "wheel graph" of G with respect to a given k dimensional embedding $E_k(G)$ is defined by $G_E^w = (V \cup \{c_i\}_{i=1}^k, E \cup \{c_i v : v \in b_i\}_{i=1}^k)$. Note that G_E^w is defined in terms of a particular embedding, and thus is not necessarily unique to G. Nevertheless its planarity turns out to be very useful to us. We will drop the reference to E when it causes no ambiguity.

Topological containment plays an important role in our algorithms. H is contained in G in the topological order, written $H \leq_t G$, if and only if a graph isomorphic to H can be obtained from G by a series of these two operations: taking a subgraph and contracting an edge at least one of whose endpoints has degree two. Equivalently, $H \leq_t G$ if and only if there is a one-to-one mapping from the vertices of H to the vertices of G under which the edges of H are mapped to vertex-disjoint paths in G. The set of endpoint images and paths is collectively called an *H-model* in G. The image endpoints are more specifically called *corners* of the model. It is well known that a graph G is planar if and only if neither $K_5 \leq_t G$ nor $K_{3,3} \leq_t G$. For series-parallel graphs, the exclusion is K_4. For outerplanar graphs, the exclusion is $K_{2,3}$ and K_4.

Path decompositions are also central to our methods. Such a decomposition of G is a pair (P, X), where P is a path and $X = \{X_i : i \in V(P)\}$ is a collection of subsets of $V(G)$ satisfying: (1) $\forall uv \in E(G), \exists\, i$ such that $\{u, v\} \subseteq X_i$ and (2) $\forall i, j, k \in V(P)$, if $i \leq j \leq k$, then $X_i \cap X_k \subseteq X_j$. The *width* of (P, X), $w(P, X)$, is $max\{|X_i| : X_i \in X\} - 1$. The *pathwidth* of G, $pw(G)$, is $min\{w(P, X) : (P, X)$ is a path decomposition of $G\}$. We observe that the pathwidth of any path is 1, while the pathwidth of the complete graph K_t is $t - 1$. Trees may have arbitrarily large pathwidth as witnessed, for example, by ternary trees whose pathwidth grows with their height.

3 Fundamentals

In what follows, let G denote a finite simple graph with n vertices and e edges. G is said to be "maximal planar" if it is planar and the addition of any new edge produces a nonplanar graph. The disk dimension of a maximal planar graph is at least $n/3$. This is because the boundary of any disk in an embedding of a maximal planar graph contains at most three vertices, else a chord could be added across the disk and planarity preserved. It is possible, of course, for arbitrary planar graphs to need fewer than $n/3$ disks. Some even require more. For example, a consequence of Theorem 2 (to follow) is that $dd(K_{2,11}) = 6 > \lceil(13/3)\rceil$.

LEMMA 1. *If* $(A, B) = K_{2,3} \leq_t G$*, and* $i \geq dd(G)$*, then the three vertices of* B *do not lie on the boundary of a single disk in any* i *dimensional embedding of* G*.*

Proof. Suppose otherwise. Extending G to G^w produces $K_{3,3} = (A \cup \{c_i\}, B) \leq_t G^w$, which is impossible because G^w must be planar. ■

THEOREM 2. *If* $K_{2,2m+1} \leq_t G$*, then* $dd(G) > m$*.*

Proof. Let $(A, B) = K_{2,2m+1} \leq_t G$. If $dd(G) \leq m$, then at least 3 vertices of B must lie on the boundary of some disk d_i. This contradicts the assertion of Lemma 1. ■

OBSERVATION 3. Let G denote a graph of disk dimension 2. Then G contains a pair of vertices, u and v, such that $dd(G - uv) = 1$. To see this, note that removing two adjacent vertices that lie on different disks leads to an embedding that unifies the disks.

LEMMA 4. *Let* G *denote a planar graph satisfying* $dd(G) = k > m > 0$*. Then* G *has at most* $2m$ *vertices* $\{u_1, u_2, ...u_l\}, l \leq 2m$*, such that* $dd(G - \{u_1, u_2, ...u_l\}) \leq k - m$*.*

Proof. Let G be given with an optimal k dimensional embedding. There are at least two disks, d_i and d_j, in the embedding that have an edge joining some u on d_i to some v on d_j. By Observation 3, removing u and v results in replacing d_i and d_j by one disk, and may reduce the disk dimension by more than one. This can be repeated at most m times or until we get a graph whose disk dimension is at most $k - m$. ■

4 A Reduction Algorithm

Given a property P of graphs, the "within-k-vertices-of P" set, denoted $W_k(P)$, is the set of all graphs that contain at most k vertices whose removal yields a graph satisfying P.

We present an algorithm that, when given any planar graph G and a positive integer k, tries to remove at most $2k - 2$ vertices of G in order to obtain an outerplanar subgraph H of G. If the algorithm fails to produce any such H, then $dd(G) > k$. Hence, our algorithm proves the following result.

THEOREM 5. *The family of graphs whose disk dimension is at most* k *is a subfamily of* $W_{2k-2}(outerplanar)$*.*

According to Lemma 4, if G has disk dimension k, then there are pairs of adjacent vertices whose removal reduces the disk dimension of G. The question is, how do we find such pair when G is not given by a k dimensional embedding?

THEOREM 6. *Let G be a graph of disk dimension k satisfying $K_4 \leq_t G$. Let u be any of the 4 corners of the K_4 model in G. Then the three neighbors of u in this model cannot all belong to the boundary of the same disk as u.*

Proof. Note first that the four corners $\{u_i\}_{i=1}^4$ of a K_4-model cannot all lie on the boundary of the same disk. Otherwise G^w would contain a K_5 in the topological order. Let v_2, v_3, and v_4 be the neighbors of u_1 in the model. Without loss of generality, assume that either $v_i = u_i$ or v_i is on the $u_1 - u_i$ path of the K_4 model. Let $\{d_i\}_{i=1}^k$ be a k dimensional embedding of G such that $u_1 \in b_1$. Assume $\{v_2, v_3, v_4\} \subset b_1$. If $u_2 \neq v_2$, then $(\{u_1, u_2\}, \{v_2, v_3, v_4\}) = K_{2,3} \leq_t G$. To see this, note that $u_2 - v_2$, $u_2 - u_3 - v_3$, and $u_2 - u_4 - v_4$ are vertex disjoint paths in the model of $K_4 \leq_t G$. But this is impossible by Lemma 1. ■

THEOREM 7. *Let G be graph of disk dimension k satisfying $(A, B) = K_{2,3} \leq_t G$. Let u be either of the two corners corresponding to A in the $K_{2,3}$ model. Then the three neighbors of u in the model can't all lie on the same disk as u.*

Proof. Let v_1, v_2 and v_3 be the neighbors of u in the model. Then $(A, \{v_3, v_4, v_5\})$ is another $K_{2,3}$ model in G. The result follows by Lemma 1. ■

We are now ready to present the reduction algorithm, Procedure $REDUCE$. A 2-corner of a $K_{2,3}$-model is either of the two elements of A when $(A, B) = K_{2,3} \leq_t G$. Function outerplanar is an outerplanarity test. If G is not outerplanar and contains a topological K_4, then a corner of a K_4-model is returned together with its three neighbors in that model. If G is not outerplanar and contains no topological K_4, then a 2-corner of a $K_{2,3}$ model is returned together with its three neighbors in that model.

Procedure REDUCE

Input: A planar graph G with n vertices and e edges, along with a non-negative integer k.
Output: Either S, a vertex subset with at most $2k-2$ elements such that $G-S$ is outerplanar, or *null* if no such subset exists.

```
begin procedure
    if (k = 0) then return null and halt
    S ← φ
    if (G is outerplanar) then return S and halt
    if (K_4 ≤_t G)
        then u ← corner of a K_4-model M in G
        else u ← 2-corner of a K_{2,3}-model M in G
    {v_0, v_1, v_2} ← neighborhood of u in M
    for i = 0 to 3 do
        S' ← REDUCE(G − {u, v_i}, k − 1)
        if (S' ≠ null) then return S ∪ S' and halt
        end do
    return null
    end procedure
```

Note that corners of a K_4-model can be found by the linear-time algorithm described in [8]. Corners of a $K_{2,3}$-model can also be found in linear time [10].

LEMMA 8. *Let (G,k) be the input to procedure $REDUCE$, where G is any planar graph of order n. The output of $REDUCE$ is either an outerplanar subgraph of order $n-2k+2$ or the fact that G is not a graph of disk dimension k. Moreover, $REDUCE$ runs in $O(3^k n)$ time.*

Proof. The preceding lemmas and discussion explain the output of $REDUCE$. As for its time complexity, note that each time vertex u is found (u is the corner of a K_4-model or a 2-corner of a $K_{2,3}$-model), we branch with three cases in the search tree. The height of the search tree is at most k. Thus it contains at most 3^k nodes. All other statements in the code require only linear time. ■

5 Exact Linear-time Algorithms

Our constructive reduction algorithm can be used to obtain efficient algorithms for other problems that are easy on outerplanar graphs and whose input is a graph of (fixed) disk dimension k.

Consider the independent set problem as an example. Let G be a graph of disk dimension k. Start by applying procedure $REDUCE$ to obtain a set R of l vertices, where $l \leq 2k-2$, such that $G \backslash R$ is outerplanar. Observe that any maximum independent set of G induces an independent set when restricted to R. We can solve the independent set problem by enumerating all independent sets of the subgraph induced by R. For each independent set I of R (including the empty set), we can find, in linear time, a maximum independent set $J(I)$ of the outerplanar graph induced by the vertices of $V(G) - (R \cup neighborhood_G(I))$. A maximum independent set of G is a maximum-size element of $\{J(I) \cup I : I$ is an independent set of $R\}$. It takes $O(4^k)$ time to enumerate all subsets of R (from these we consider only those that are independent). Therefore, the total run time is: $O(3^k n + 4^k k + n)$, which is $O(n)$ since k is fixed.

The maximum independent set algorithm just described can be imitated to obtain linear-time algorithms for other problems. These include vertex cover, 3-coloring, 4-coloring, dominating set and many others.

6 Pathwidth Approximation

The treewidth of graphs of disk dimension k is $O(\sqrt{k})$ [1]. Therefore, optimal path decompositions of graphs of disk dimension k can at least theoretically be obtained in polynomial time. This is due to [4], where it is asserted that optimal path decompositions can be found in polynomial time for graphs of bounded treewidth. The algorithm suggested there is not practical, however, because it operates on sets of size $O(n^{11})$ in its first step.

Motivated by recent fast approximation algorithms for the pathwidth of graphs of disk dimension one [7], we show how to get similar algorithms for graphs of disk dimension k. We rely again on Lemma 4, by using Procedure $REDUCE$, to obtain a linear-time algorithm whose worst-case performance ratio is 3. Take, for example, a graph of disk dimension two. Delete a suitable pair of vertices $\{u, v\}$. The resulting graph, H, is outerplanar. Using the work of [7], we can find a path decomposition (P, X) that is not more than $3pw(H) + 2$ in $O(n)$ time. (The bound stated in [7] is only O($nlogn$), because the algorithm described there relies on obtaining optimal path decompositions of trees. It has been shown, recently, that such decompositions can be computed in linear time [9]. Thus the algorithm of [7] runs in linear time as well.) Adding the two vertices to every element of X gives a path decomposition of G of width $\leq 3pw(H) + 4 \leq 3pw(G) + 4$ because $H \subseteq G$. If H were biconnected, we could use the algorithm of [3] to obtain a path decomposition of width $\leq 2pw(H) + 1$.

Procedure $DECOMPOSE$, shown below, uses the function *outpl_pw* to obtain a path decomposition of a given outerplanar graph, H, in linear time. The width of the path decomposition returned by *outpl_pw* is not more than $3pw(H)+2$. Thus, given graph G as input, $DECOMPOSE$ returns a path decomposition of width not exceeding $3pw(G)+2k$. It outputs *null* only if $dd(G) > k$.

Procedure DECOMPOSE

```
Input: A planar graph G with n vertices and e edges, along with a
          non-negative integer k.
Output: A path decomposition (P, X) of G.
begin procedure
    if (G is outerplanar) then do
        (P, X) ← outpl_pw(G)
        return (P, X) and halt
        end do
    S ← REDUCE(G, k)
    if (|S| ≤ 2k − 2) then do
        (P, X′) ← outpl_pw(G\S)
        X ← {X_i ∪ S : X_i ∈ X′}
        return (P, X) and halt
        end do
    return null
    end procedure
```

THEOREM 9. *If G is graph of disk dimension k and order n, a path decomposition of G with width at most $3pw(G)+2k$ can be constructed in $O(3^k n)$ time.*

Proof. Obtaining the path decomposition of width at most $3pw(G)+2k$ has been described in details in the preceding paragraphs. The claimed time complexity follows from Lemma 8 since the time consuming part of function $DECOMPOSE$ is the call to $REDUCE$. Other parts of the code run in linear time. ■

7 Remarks

We showed that graphs of disk dimension k or less are within $2k-2$ vertices of outerplanar graphs. The containment of graphs of disk dimension k in $W_{2k-2}(outerplanar)$ is proper. In fact, for arbitrary $r > 0$, a graph, G_r, that has disk dimension r and is within one vertex of outerplanar can be constructed from K_1 and $(r-1)$ copies of K_3 by connecting a vertex u representing K_1 to all vertices of $(r-1)K_3$. The resulting graph, $G_r =$

$K_1 + (r-1)K_3$, is shown in Figure 1. G_r is (obviously) planar and has disk dimension r. Procedure $REDUCE$ will do a good job on this graph since it would only take out two vertices of which one is u. We are investigating the behaviour of $REDUCE$ on general $W_{2k-2}(outerplanar)$ graphs and we hope to be able to show that our algorithms for graphs of disk dimension k extend automatically to $W_{2k-2}(outerplanar)$.

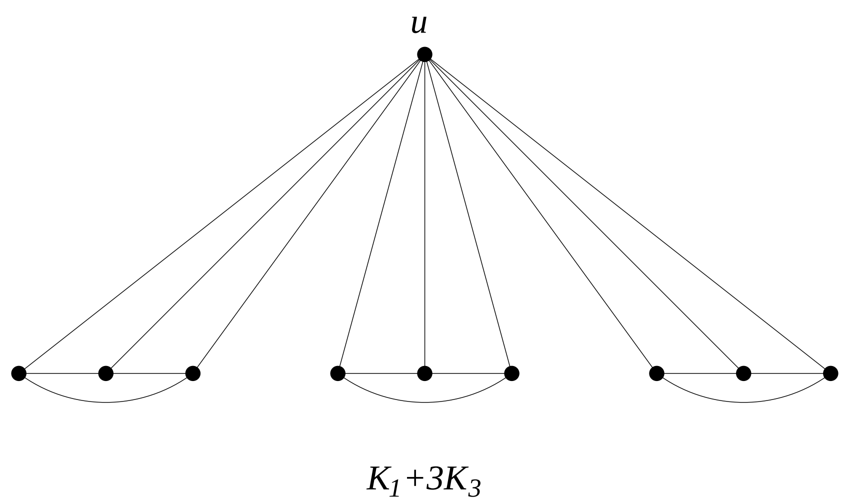

Figure 1. A "within one vertex of outerplanar" graph whose disk dimension is four.

BIBLIOGRAPHY

[1] F. N. Abu-Khzam. *Topics in Graph Algorithms: Structural Results and Algorithmic Techniques, with Applications*. PhD thesis, Department of Computer Science, University of Tennessee, 2003.

[2] D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM Journal on Computing*, 17:53–76, 1988.

[3] H. L. Bodlaender and F. V. Fomin. Approximation of pathwidth of outerplanar graphs. *Journal of Algorithms*, 43(2):190–200, 2002.

[4] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.

[5] M. Fellows, F. Hickling, and M. Syslo. A topological parameterization and hard graph problems. *Congressius Numeratium*, 59:69–78, 1987.

[6] M. R. Fellows and M. A. Langston. Nonconstructive advances in polynomial-time complexity. *Information Processing Letters*, 26:157–162, 1987.

[7] R. Govindan, M. A. Langston, and X. Yan. Approximating the pathwidth of outerplanar graphs. *Information Processing Letters*, 68:17–23, 1998.

[8] P. C. Lui and R. C. Geldmacher. An O(max(m, n)) algorithm for finding a subgraph homeomorphic to K_4. *Congressius Numeratium*, 29:597–609, 1980.
[9] K. Skodinis. Computing optimal linear layouts of trees in linear time. Technical Report MIP-9904, Department of Computer Science, University of Passau, Passau, Germany, 1999.
[10] R. Thomas. Private communication, 2001.

Faisal N. Abu-Khzam
Division of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon
faisal.abukhzam@lau.edu.lb

Michael A. Langston
Department of Computer Science
University of Tennessee
Knoxville, TN 37996–3450
USA
langston@cs.utk.edu