INFORMS Journal on Computing
Vol. 17, No. 4, Fall 2005, pp. 402–412
ISSN 1091-9856 | EISSN 1526-5528 | 05 | 1704 | 0402

DOI 10.1287/ijoc.1040.0075

Planar Branch Decompositions I: The Ratcatcher

Illya V. Hicks
Department of Industrial Engineering, Texas A&M University, College Station, Texas 77843-3131, USA,
ivhicks@tamu.edu

The notion of branch decompositions and its related connectivity invariant for graphs, branchwidth, were introduced by Robertson and Seymour in their series of papers that proved Wagner's conjecture. Branch decompositions can be used to solve NP-hard problems modeled on graphs, but finding optimal branch decompositions of graphs is also NP-hard. This is the first of two papers dealing with the relationship of branchwidth and planar graphs. A practical implementation of an algorithm of Seymour and Thomas for only computing the branchwidth (not optimal branch decomposition) of any planar hypergraph is proposed. This implementation is used in a practical implementation of an algorithm of Seymour and Thomas for computing the optimal branch decompositions for planar hypergraphs that is presented in the second paper. Since memory requirements can become an issue with this algorithm, two other variations of the algorithm to handle larger hypergraphs are also presented.

Key words: planar graph; branchwidth; branch decomposition; carvingwidth
History: Accepted by Jan Karel Lenstra, former Area Editor for Design and Analysis of Algorithms; received December 2002; revised August 2003, January 2004; accepted February 2004.

1. Introduction

A *planar graph* is a graph that can be drawn on a sphere or plane without having edges that cross. A subdivision of a graph G is a graph obtained from G by replacing its edges by internally vertex disjoint paths. In the 1930s, Kuratowski (1930) proved that a graph G is planar if and only if G does not contain a subdivision of K_5 or $K_{3,3}$. Wagner (1937) later proved that a graph G is planar if and only if G does not contain K_5 or $K_{3,3}$ as a minor of G. A graph H is a *minor* of a graph G if H can be obtained from a subgraph of G by contracting edges. Let $\mathcal{F}$ be a class of graphs. $\mathcal{F}$ is *minor closed* when all the minors of any member of $\mathcal{F}$ also belong to $\mathcal{F}$. Given a minor closed class of graphs $\mathcal{F}$, the *obstruction set* of $\mathcal{F}$ is the set of minor minimal graphs that are not elements of $\mathcal{F}$. Clearly, any class of graphs embeddable on a given surface is a minor closed class. The question of characterizing the obstruction set for other surfaces remained open until 1979–1980 when Archdeacon (1980) and Glover et al. (1979) solved the case for the projective plane where they proved that there are 35 minor minimal "nonprojective-planar" graphs. In the 1930s, Erdös posed the question of whether the obstruction set for a given surface is finite. Archdeacon and Huneke (1989) proved that the obstruction set for any nonorientable surface is finite, and Robertson and Seymour (1985) proved the case for any surface as a corollary of the graph minors theorem, which proved Wagner's conjecture, every minor closed class has a finite obstruction set. In addition, Robertson and Seymour conceived of two new ways to decompose a graph that were beneficial toward the proof. Decomposing a graph by a branch decomposition is one of these ideas.

Branch decompositions open algorithmic possibilities for intractable problems that can be modeled on graphs. Arnborg et al. (1991) showed that several NP-complete problems can be solved in polynomial time using dynamic-programming techniques on input graphs with bounded treewidth, another related connectivity invariant for graphs introduced by Robertson and Seymour (1983). Actually, the work of Arnborg et al. was spurred from the work of Courcelle (1990). In addition, similar results have been obtained by Borie et al. (1992). The Arnborg et al. result is also equivalent to graphs with bounded branchwidth since the branchwidth and treewidth of a graph bound each other by constants (Robertson and Seymour 1991). In addition, there is a plethora of theory about branchwidth and branch decompositions. For example, branchwidth and branch decompositions have been shown useful in producing matroid analogs of the graph minors theorem and shorter proofs of the graph minors theorem related to graphs with bounded branchwidth/treewidth (Geelen et al. 2002). Also, Seymour and Thomas (1994) proposed a polynomial-time algorithm to compute the branchwidth for planar graphs, which will be referred to as the *ratcatcher* method.

Despite the number of theoretical results about branch decompositions and branch decomposition-based algorithms for solving NP-hard problems, there has been little effort to develop practical algorithms. Examples of practical branch decomposition-based algorithms are the work of Cook and Seymour (2003) on the traveling salesman problem (TSP) and the work of Hicks (2004a) on the minor containment problem. One is also referred to the work of Christian (2003) for other examples of branch decomposition-based algorithms. Similar results for tree decomposition-based algorithms have been given by Koster et al. (2002) for their work on partial constrain satisfaction problems, and Alber and Neidermeier (2002) for their work on domination problems. These examples exemplify the possible usefulness of branch decomposition and tree decomposition-based algorithms for other NP-hard problems.

Seymour and Thomas (1994) proved that testing whether a general graph has branchwidth at most k, some input integer, is NP-complete. However, the aforementioned ratcatcher method is a polynomial-time algorithm used to compute the branchwidth of planar graphs. Given an integer k, the ratcatcher method tests whether the branchwidth of the graph is at least k by finding a structure in the graph that prohibits finding a branch decomposition of the graph with width at most $k-1$. Seymour and Thomas (1994) also proposed a polynomial-time algorithm using the ratcatcher method to compute optimal branch decompositions for planar graphs, which will be referred to as the *edge-contraction* method. Thus, the current knowledge base to compute optimal branch decompositions for planar graphs is a two-stage process. The two-stage process involves first finding the branchwidth of the input planar graph using the ratcatcher method, then using the edge-contraction method in conjunction with the ratcatcher method to examine planar hypergraphs needed to construct the optimal branch decomposition for the input planar graph. A more detailed discussion of the implementation of the edge-contraction method with the addition of a heuristic, the *cycle* method, for runtime speedup is presented in the related paper (Hicks 2005).

This paper is spurred from the Seymour and Thomas (1994) paper; a practical implementation of the ratcatcher method is presented. Unfortunately, the algorithm requires a significant amount of memory to achieve the proposed complexity of $O(e^2)$ where e is the number of edges in the input graph. This memory requirement limits the size of the input graphs for this algorithm (up to 5,000 edges). Thus, two additional methods, the *comprat* method and the *memrat* method, with complexity $O(e^3)$, are presented to handle medium (up to 20,000 edges) and large (up to 70,000 edges) graphs respectively.

Foundational definitions are given in §§2.1, 2.2, and 2.3. Section 2.4 describes the two-player game related to the algorithm. Sections 3 and 4 are reserved for the details of the algorithms with a proof of correctness for the additional algorithms. Finally, §§5 and 6 contain computational results and conclusions, respectively. The reader is referred to West (2001) for background material on graph theory.

2. Preliminaries

2.1. Branch Decompositions

Throughout this paper, a graph is only considered to be undirected and may have loops and multiple edges unless stated otherwise. Also, we may assume that all graphs are biconnected. For instance, for a graph G, one can derive an optimal branch decomposition for G from the optimal branch decompositions of G's connected components. In addition, for a connected graph G_o, one can derive an optimal branch decomposition for G_o from the optimal branch decompositions of the biconnected components of G_o.

A *hypergraph* H consists of a finite set $V(H)$ of nodes, a finite set $E(H)$ of edges, and an incidence relation between them that is not restricted to two ends for each edge. Thus, hypergraphs are generalizations of graphs where edges can have any number of ends. Since this paper is about planar graphs and planar hypergraphs, let us define planarity for hypergraphs. Let H be a hypergraph; then $I(H)$, the *incidence graph* of H, is the simple bipartite graph with vertex set $V(H) \cup E(H)$ such that $v \in V(H)$ is adjacent to $e \in E(H)$ if and only if v is an end of e in H. Seymour and Thomas (1994) define a hypergraph H as being planar if and only if $I(H)$ is planar. In addition, we will denote G^* as a planar dual of an embedding of the graph G in the sphere or plane. Furthermore, for a planar graph G drawn in the sphere, $R(G)$ will denote the set of regions of G.

Let us now define branch decompositions and branchwidth of a graph. Let G be a graph with node set $V(G)$ and edge set $E(G)$. Let T be a tree having $|E(G)|$ leaves in which every nonleaf node has degree three. Let ν be a bijection between the edges of G and the leaves of T. The pair (T, ν) is called a *branch decomposition* of G. Notice that removing an edge e of T partitions the edges of G into two subsets A_e and B_e. The *middle set* of e, denoted by *mid*(e), is the set $V(A_e) \cap V(B_e)$. The *width* of a branch decomposition (T, ν) is the maximum cardinality of the middle sets over all edges in T. The *branchwidth* of G, denoted by $\beta(G)$, is the minimum width over all branch decompositions of G. A branch decomposition of G is *optimal* if its width is equal to the branchwidth of G. If there exists a nonleaf node in T with degree greater than three, then the pair (T, ν) is called a *partial branch decomposition*. Figure 1 offers an example planar graph with an optimal branch decomposition where some of the middle sets are given. Graphs of small branchwidth are characterized by the following theorem.

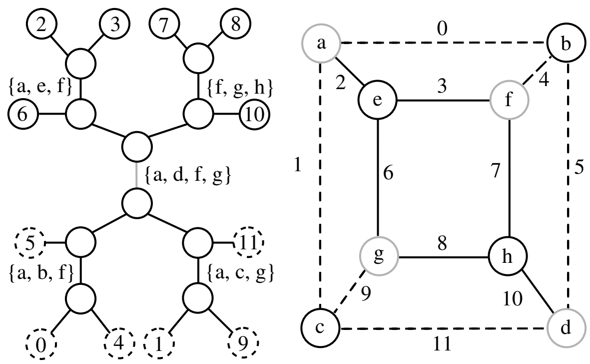


Figure 1 **Example Planar Graph and Optimal Branch Decomposition**

Theorem 1 (Robertson and Seymour 1991). *Let G be a graph.*

- $\beta(G) = 0$ *if and only if every component of G has* ≤ 1 *edge.*
- $\beta(G) \leq 1$ *if and only if every component of G has* ≤ 1 *node of degree* ≥ 2.
- $\beta(G) \leq 2$ *if and only if G has no* K_4 *minor.*

Therefore, 1-factors (perfect matchings) have branchwidth equal to zero; *stars* (trees with only one nonleaf node) are the only connected graphs with branchwidth equal to one; all forests that are not in the union of one factors and stars have branchwidth equal to two. Also, series-parallel and outerplanar graphs have branchwidth at most two because these graphs do not have K_4 as a minor. The branchwidth of other familiar graphs can be found in Hicks (2000).

To achieve a better understanding of the ratcatcher method, it would behoove us to discuss the min-max relationship associated with branchwidth. Let G be a graph and let $k \geq 1$ be an integer. A *separation* of a graph G is a pair (G_1, G_2) of subgraphs of G with $G_1 \cup G_2 = (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2)) = G$ and $E(G_1) \cap E(G_2) = \varnothing$ and the *order* of this separation is defined as $|V(G_1) \cap V(G_2)|$. A *tangle* in G of *order* k is a set $\mathcal{T}$ of separations of G, each of order $<k$ such that:

(T1) for every separation (A, B) of G of order $<k$, one of (A, B), (B, A) is an element of $\mathcal{T}$;

(T2) if $(A_1, B_1), (A_2, B_2), (A_3, B_3) \in \mathcal{T}$, then $A_1 \cup A_2 \cup A_3 \neq G$; and

(T3) if $(A, B) \in \mathcal{T}$ then $V(A) \neq V(G)$.

These are called the *first*, *second*, and *third tangle axioms*. The *tangle number* of G, denoted by $\theta(G)$, is the maximum order of any tangle in G. Figure 2 gives an example of a tangle of order 4 for the graph of Figure 1 where $G[S]$ denotes the induced subgraph of G by the set S, which could be a node set or an edge set. Notice in Figure 2 that the inclusion of separations of the graph of order four to the tangle would result in a violation of one of the tangle axioms. A tangle $\mathcal{T}$ of G with order k can be thought of as a "k-connected" component of G because some

Separation of order 0
$(\varnothing, G)$
Separations of order 1
$(G[\{v\}], G)\ \forall v \in V(G)$
Separations of order 2
$(G[\{v\}] \cup G[\{w\}], G)\ \forall v, w \in V(G)$
$(G[\{e\}], G\backslash\{e\})\ \forall e \in E(G)$
Separations of order 3
$(G[\{v\}] \cup G[\{w\}] \cup G[\{u\}], G)\ \forall v, w, u \in V(G)$
$(G[\{v\}] \cup G[\{e\}], G\backslash\{e\})\ \forall e \in E(G)$ and $v \in V(G)$
such that v is not incident to e
$(G[\{e, f\}], G\backslash\{e, f\})\ \forall$ pairs $e, f \in E(G)$
such that e and f share a node
$(G[\{0, 1, 2\}], G\backslash\{a\})$
$(G[\{0, 4, 5\}], G\backslash\{b\})$
$(G[\{1, 9, 11\}], G\backslash\{c\})$
$(G[\{2, 3, 6\}], G\backslash\{e\})$
$(G[\{3, 4, 7\}], G\backslash\{f\})$
$(G[\{5, 10, 11\}], G\backslash\{d\})$
$(G[\{6, 8, 9\}], G\backslash\{g\})$
$(G[\{7, 8, 10\}], G\backslash\{h\})$

Figure 2 **Tangle of Order Four for the Graph of Figure 1**

"k-connected" component of G will either be on one side or the other for any separation of $\mathcal{T}$. Robertson and Seymour (1991) proved a strong min-max relation between tangles and branchwidth:

Theorem 2 (Robertson and Seymour 1991). *For any simple graph G such that* $E(G) \neq \varnothing$, $\max(\beta(G), 2) = \theta(G)$.

In terms of finding branch decompositions for general graphs, there is an algorithm in Robertson and Seymour (1995) to approximate the branchwidth of a graph within a factor of three. For example, the algorithm decides whether a graph has branchwidth of at least five or finds a branch decomposition with a width at most fifteen. This algorithm has not been used in a practical implementation and its improvements by Bodlaender (1996), Bodlaender and Kloks (1996), and Reed (1997) have not been shown to be practical either. Bodlaender and Thilikos (1999) did give an algorithm to compute the optimal branch decomposition for any chordal graph with maximum clique size at most four, but the algorithm has been shown to be practical only for a particular type of 3-tree. Bodlaender and Thilikos (1997) also developed a tree decomposition-based linear-time algorithm for finding an optimal branch decomposition, but it appears to be impractical.

Under practical algorithms, Kloks et al. (1999) gave a polynomial-time algorithm to compute the branchwidth of interval graphs, but for general graphs, one has to rely on heuristics. Cook and Seymour (2003) gave a heuristic algorithm to produce branch decompositions that shows promise. In addition, Hicks (2000, 2002) also found another branchwidth heuristic that was comparable to the algorithm of Cook and Seymour. Recently, Tamaki (2003) has presented a linear-time heuristic for near-optimal branch decompositions of planar graphs. This algorithm performs well when compared to the heuristics of Cook and Seymour (2003) and Hicks (2002) and may give some insight to finding an algorithm for optimal branch decompositions of planar graphs with lower complexity than the currently known methods.

Robertson and Seymour (1983) conceived a new way to decompose a graph to display the similarities between the graph and trees. This concept, *tree decompositions*, and the associated connectivity invariant, *treewidth*, have been extensively researched by Thomas (1990), Seymour and Thomas (1993), Bodlaender and Kloks (1996), Bodlaender (1996, 1998), Ramachandramurthi (1997), Reed (1992, 1997) and many others (see the survey papers by Bodlaender 1993, 1997). The reader is also referred to the work of Koster et al. (2001, 2002), Telle and Proskurowski (1997), and Alber and Neidermeier (2002) for some computational results related to tree decompositions and tree decomposition-based algorithms. The difference between branchwidth and treewidth is that branchwidth deals with edges and treewidth deals with nodes.

Since the ratcatcher method is related to a game, it was only right to discuss a game related to treewidth. Seymour and Thomas (1993) presented a characterization of treewidth of any connected graph by a search game conducted on that graph. The game consists of a robber and k cops where k is some positive integer. The robber, with immense speed, can travel from node to node using paths of the graphs. The robber cannot run through a node occupied by a cop. Each of the cops is either on a node of the graph or in a helicopter en route to a node. Clearly, the objective of the cops is for at least one cop to land on a node occupied by the robber; the robber's objective is to avoid capture. In addition, the robber can observe a helicopter approaching a node and may run to a new node before the helicopter lands. The game stops once the robber has been captured. Seymour and Thomas (1993) showed that the graph has treewidth of at most k if and only if $k+1$ cops can capture the robber on the graph. For other games related to treewidth, the reader is referred to an excellent survey paper by Bienstock (1991).

2.2. Carving Decompositions

Now that we have a attained a good grasp of the definitions associated with branchwidth and branch decompositions, we can concentrate on definitions relevant to the ratcatcher method. Let G be a graph with node set $V(G)$ and edge set $E(G)$. Let T be a tree having $|V(G)|$ leaves in which every nonleaf node has degree three. Let μ be a bijection between the nodes of G and the leaves of T. The pair (T, μ) is called a *carving decomposition* of G. Notice that removing an edge e of T partitions the nodes of G into two subsets A_e and B_e. The *cut set* of e is the set of edges that are incident to nodes in A_e and to nodes in B_e (also denoted $\delta(A_e)$ or $\delta(B_e)$). The *width* of a carving decomposition (T, μ) is the maximum cardinality of the cut sets for all edges in T. The *carvingwidth* for G, $\kappa(G)$, is the minimum width over all carving decompositions of G. A carving decomposition is also known as a *minimum-congestion routing tree* and one is referred to Alvarez et al. (2000) for a link between carvingwidth and network design. The ratcatcher method is really an algorithm to compute the carvingwidth for planar graphs. To show the relation between carvingwidth and branchwidth we need another definition.

Let G be a planar graph and denote the planar embedding of the graph on the sphere. For every node v of G, the edges incident to v can be ordered in a clockwise or counter-clockwise order. This ordering of edges incident to v is the *cyclic order* of v. Let $M(G)$ be a graph with the vertex set $E(G)$ and let the edges of $M(G)$ be the union of cycles C_v for all $v \in V(G)$ where C_v is the cycle through the nodes of $M(G)$ that correspond to the edges incident with v according to v's cyclic order. $M(G)$ is called a *medial graph* of G. The medial graph of Figure 1 is given in Figure 3. One can notice that every connected planar hypergraph G with $E(G) \neq \varnothing$ has a medial graph and every medial graph is planar. In addition, notice that their is a bijection between the regions of $M(G)$ and the nodes and regions of G. Using this relationship, Hicks (2000) proved that if a planar graph and its dual are both loopless then they have the same branchwidth. Figure 4 illustrates this result by presenting one branch decomposition for both cube3 and the octahedron. For the relationship between branchwidth and carvingwidth, Seymour and Thomas (1994) proved:

Theorem 3 (Seymour and Thomas 1994). *Let G be a connected planar graph with $|E(G)| \geq 2$, and let $M(G)$ be the medial graph of G. Then the branchwidth of G is half the carvingwidth of $M(G)$.*

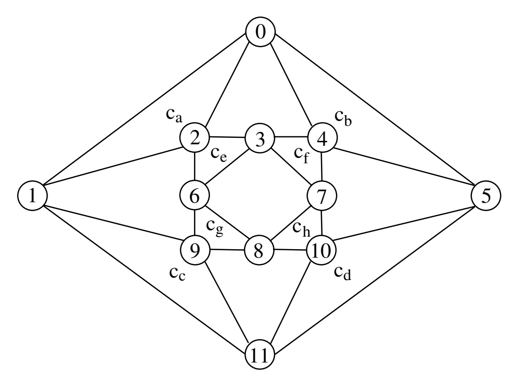


Figure 3 Medial Graph of the Graph in Figure 1

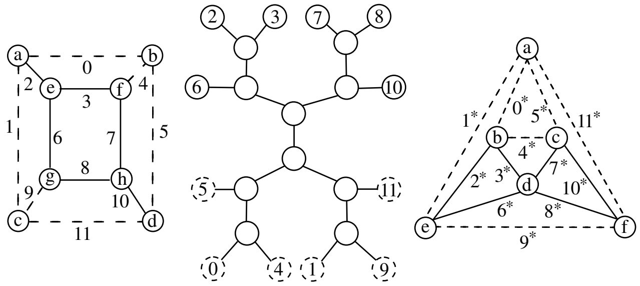


Figure 4 Cube3 and the Octahedron Have Branchwidth Four

Therefore, computing the carvingwidth of $M(G)$ gives us the branchwidth of G. Also, having a carving decomposition of $M(G)$, (T, μ), gives us a branch decomposition of G, (T, ν), such that the width of (T, ν) is exactly half the width of (T, μ). The ratcatcher method actually computes the carvingwidth of planar graphs. In addition, the ratcatcher method does not search for low cut sets in the medial graph but for objects that prohibit the existence of low cut sets. These objects are called *antipodalities*.

2.3. Tilts, Slopes, and Antipodalities

To discuss fully the relationship between carvingwidth and antipodalities, more definitions are needed. Let G be a graph. A *tilt* in G of order k is a collection $\mathcal{B}$ of subsets of $V(G)$ such that the following hold:

(B1) for every $X \subseteq V(G)$ such that $\delta(X) < k$, either X or $V(G)\backslash X$ is an element of $\mathcal{B}$;

(B2) if X_1, X_2, $X_3 \in \mathcal{B}$ then $X_1 \cup X_2 \cup X_3 \neq V(G)$; and

(B3) $X \in \mathcal{B}$ for all $X \subseteq V(G)$ with $|X| = 1$.

For the graph in Figure 1, one can think of a tilt of order eight for the medial graph in Figure 3 as restricting the tangle in Figure 2 to only separations where both sides of the separations are subgraphs induced by edge sets.

In the nature of planar graphs, if there exists a tilt of order k for a planar graph G then there should exist some analogous structure for G^*. These are called *slopes*. Let G be a planar graph drawn in the sphere Σ and let $k \geq 1$ be an integer. A *slope* in G of order $k/2$ is a function *ins* that assigns to every cycle C of G of length $<k$ a closed disc $ins(C) \subseteq \Sigma$ that is one of the two closed discs bounded by C such that the following hold:

(S1) If C and C' are cycles of length $<k$ and C is drawn within $ins(C')$ then $ins(C) \subseteq ins(C')$; and

(S2) If P_1, P_2, and P_3 are three internally vertex disjoint paths of G joining the nodes u and v and the three cycles $P_1 \cup P_2$, $P_2 \cup P_3$, and $P_1 \cup P_3$ all have length $<k$, then $ins(P_1 \cup P_2) \cup ins(P_2 \cup P_3) \cup ins(P_1 \cup P_3) \neq \Sigma$.

A slope is *uniform* if for every region $r \in R(G)$ there is a cycle C of G with length $<k$ such that $r \subseteq ins(C)$. Slopes were the inspiration for the cycle method discussed in the related paper (Hicks 2005). In addition, part of the proof of Theorem 3 is a theorem by Robertson and Seymour (1994) that for any graph G that is embeddable on a surface Σ, a relationship is developed between uniform slopes of the dual of the medial graph of G embedded in Σ and tangles of G restricted by the embedding.

Finally, we can define antipodalities and give the relationship between carvingwidth, tilts, slopes, and antipodalities. Given a planar graph G and its dual G^*, an edge e of G is *incident* with a region $r \in R(G)$ if e^* is incident to $r \in V(G^*)$. A *walk* in G is a sequence $v_0, e_1, v_1, e_2, \ldots, e_t, v_t$, where $v_0, v_1, \ldots, v_t \in V(G)$, $e_1, e_2, \ldots, e_t \in E(G)$, and v_{i-1}, v_i is the set of ends of e_i $(1 \leq i \leq t)$. A walk is *closed* if v_0 equals v_t. An *antipodality* in G of *range* at least k is a function α with domain $E(G) \cup R(G)$, such that for all $e \in E(G)$, $\alpha(e)$ is a nonnull subgraph of G, and for all $r \in R(G)$, $\alpha(r)$ is a nonempty subset of $V(G)$, satisfying:

(A1) If $e \in E(G)$ then no end of e belongs to $V(\alpha(e))$.

(A2) If $e \in E(G)$, $r \in R(G)$, and e is incident to r, then $\alpha(r) \subseteq V(\alpha(e))$, and every component of $\alpha(e)$ has a vertex in $\alpha(r)$.

(A3) If $e \in E(G)$ and $f \in E(\alpha(e))$ then every closed walk of G^* using e^* and f^* has length $\geq k$.

In addition, an antipodality α is called *connected* if $\alpha(e)$ is connected for all edges e of G. The main result in Seymour and Thomas (1994) is the following.

THEOREM 4. *Let G be a connected planar graph with $|V(G)| \geq 2$ drawn in a sphere Σ, let G^* be a dual graph, and let $k \geq 0$ be an integer. Then the following are equivalent*:

$\delta(v) < k$ $\forall v \in V(G)$ and $\kappa(G) \geq k$;
$\delta(v) < k$ $\forall v \in V(G)$ and G has a tilt of order k;
G^* has a uniform slope of order $k/2$;
G has a connected antipodality of range $\geq k$;
G has an antipodality of range $\geq k$.

Thus, given an integer k, the ratcatcher method is an algorithm to test whether a planar graph has carvingwidth at most k by searching for an antipodality of range at most k.

2.4. The Ratcatcher Game

Consider a two-player game played on a planar graph G. Think of the graph as a floor plan of a house where the regions are rooms and the edges are walls; also, assume that each wall e has a corresponding door and a positive integer thickness $p(e)$. One player of the game is a rat that is content to stay in the house and run along the walls of the house. The other player is a ratcatcher who was hired by the owners of the house to catch the rat. The ratcatcher's only weapon to catch the rat is a whistle of some fixed sound level k. Depending on the thickness of the walls, the whistle's sound can penetrate the walls affecting the rat; the rat can not run along noisy walls. A edge e is considered "quiet" if there is no closed walk in G^* of p-length $<k$ using e^* and r^* if the ratcatcher is in room r, or using e^* and f^* if the ratcatcher is in wall f. This full-knowledge game starts with the ratcatcher selecting a room (the outside is considered a room also) and the rat selects a corner. The ratcatcher moves first and the players move in turn until the rat is caught. The ratcatcher can catch the rat if the rat is cornered by noisy walls. The rules of the game are that each player knows the location of the other and once the ratcatcher enters a doorway on one move, the ratcatcher has to go through the doorway into the next room on the next move. Seymour and Thomas (1994) showed that the largest sound level such that the rat will always escape when the wall thickness is uniformly one is the carvingwidth of that planar graph.

Thus, an antipodality of range at least k is an escape strategy for the rat when the sound level of the ratcatcher's whistle is k. Initially, if the ratcatcher selects room r then the rat should select a corner in $\alpha(r)$. Also, when the ratcatcher is in the doorway of some edge e, the rat knows the ratcatcher will visit some room s. Therefore, the rat travels along the edges of antipodality of e until it reaches a node in the antipodality of s. In addition, when the ratcatcher moves from a doorway to a room, the rat should stay still. If the rat follows these rules, the rat will never be caught.

3. The Ratcatcher Algorithm

Given a graph G, the ratcatcher algorithm constructs $M(G)$ and $M(G)^*$ and tests whether the carvingwidth of $M(G)$ is at least k by testing whether there exists an antipodality of range at least k. To test for the existence of an antipodality of range at least k for an edge e of the medial graph, edges are identified that satisfy (A1) and (A3). For each $e \in E(M(G))$, let $\phi(e)$ be the subgraph of $M(G)$ where the nodes of $\phi(e)$ consist of $V(M(G))$, except the ends of e and the edges of $\phi(e)$ consist of all the edges f such that no end of f is an end of e, and no closed walk of $M(G)^*$ of length less than k contains both e^* and f^*. For a given graph G, let $d(u, v)$ denote the length of the shortest path between $u, v \in V(G)$. If e^* and f^* are distinct and have ends u_1, u_2 and v_1, v_2, respectively, in G^*, then there is a closed walk of length $<k$ in G^* using e^* and f^* if and only if either $d(u_1, v_1) + d(u_2, v_2) + 2 < k$ or $d(u_1, v_2) + d(u_2, v_1) + 2 < k$. Thus, an array is kept to keep the information for the all-pairs shortest paths for the dual of the medial graph. The authors proved that if there exists an antipodality of $M(G)$ of range at least k then there is one, say α, such that $\alpha(e)$ is a union of components of $\phi(e)$ for each $e \in E(M(G))$ (Seymour and Thomas 1994). The dashed edges in the graph of Figure 5 illustrates the edge set of $\alpha(e)$ where e is the dotted edge.

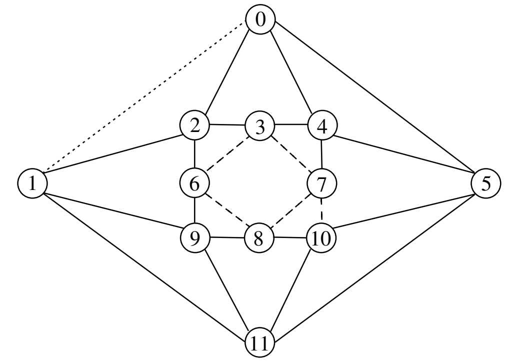


Figure 5 **Example $\alpha(e)$ for an Antipodality of Range Eight**

Once $\phi(e)$ for all edges $e \in E(M(G))$ has been established, for each region $r \in R(M(G))$, Seymour and Thomas focus on finding the appropriate subset of nodes of $M(G)$ that satisfy (A2) for that region. Hence, they prove that if $\alpha(r)$ is nonempty for each region $r \in R(M(G))$, then for each edge $e \in E(M(G))$ incident to a region r_o, $\alpha(e)$ is the union of all components of $\phi(e)$ that contain a node of $\alpha(r_o)$. Thus, $M(G)$ has an antipodality of range k for some positive integer k if and only if for each region $r \in R(M(G))$ there exists a nonempty set that satisfies (A2).

To explain how Seymour and Thomas (1994) suggested finding the antipodality of range k for the regions of $M(G)$, we need another definition. Let N and L be simple graphs and let $(X_i \mid i \in V(N))$ be a partition of $V(L)$ such that if $u \in X_i, v \in X_j$ are adjacent in L, then $i \neq j$ and i, j are adjacent in N. A set $R \subseteq V(L)$ is *round* if for all $i, j \in V(N)$ adjacent in N, every vertex in $X_i \cap R$ is adjacent to some vertex in $X_j \cap R$.

For an edge $e \in E(M(G))$, let C_e denote the set of components of $\phi(e)$. Also, for each $e \in E(M(G))$, let X_e be the set of all pairs $\{(e, C) \mid C \in C_e\}$ and for each $r \in R(M(G))$, let X_r be the set of all pairs $\{(r, v) \mid v \in V(M(G))\}$. Let $I = E(M(G)) \cup R(M(G))$. Seymour and Thomas (1994) construct a graph L with vertex set $\bigcup(X_i \mid i \in I)$ in which $(e, C) \in X_e$ is adjacent to (r, v) in X_r if $e \in E(M(G))$, $r \in R(M(G))$, e is incident with r, and $v \in V(C)$. They also construct a graph N with vertex set I in which $e \in E(M(G))$ is adjacent to $r \in R(M(G))$ if e is incident with $r \in M(G)$. In addition, they also construct the graph H with vertex set $V(L) \cup V(N)$ in which if $e \in E(M(G))$ and $r \in R(M(G))$ are adjacent in N then e is adjacent in H to every vertex in X_r and r is adjacent in H to every vertex in X_e. The edges of H are referred as pairs vj. The weight of the edges of vj are dictated by the interaction between v and j. Then, the authors use an algorithm with complexity $O(|E(H)| + |V(L)| + |E(L)|)$ to test whether the graph L has a round set. The algorithm starts by constructing a stack of vertices of $V(M)$ incident to a zero-weight edge in H and initializing the set S to be

$V(M)$. As a node u is taken off the stack, it is taken out of S and for all of its neighbors v not already in the stack, reduce the weight of the edge uv by one and if the weight becomes zero, then insert v into the stack. Once the stack is empty and S is nonempty, then S is a round set for L. Otherwise, a round set for L does not exist. The authors showed that L has a round set if and only if $M(G)$ has an antipodality of range k.

The ratcatcher method included this round-set algorithm with the modification of replacing the three graphs by arrays that kept all vital information. Specifically, each region r has a set $\phi(r)$ initially set to be $V(M)$. In addition, each edge e of the medial graph had three arrays whose size is the number of nodes of the medial graph, to keep the knowledge of the components of $\phi(e)$ and the number of $\phi(r_1)$ and $\phi(r_2)$ nodes on the components, where r_1 and r_2 are the regions incident with e. Also, each "root" of the components kept a linked list of the members of the components to ensure the same complexity as Seymour and Thomas. The stack is filled with the pairs rv where r is a region and v is a node incident with r. Once rv is taken off the stack, for each edge e incident with r, the number of $\phi(r)$ nodes on the same component of $\phi(e)$ as v is decreased by one. For an edge e incident with r and another region r_o, if the number of $\phi(r)$ nodes on the same component of $\phi(e)$ as v is zero, then the pair $r_o w$ is put in the stack if $r_o w$ is not already on the stack. In addition, if $\phi(r)$ is empty for a region r, then the algorithm terminates early because an antipodality for the input range does not exist (Seymour and Thomas 1994).

4. Memrat and Comprat

The ratcatcher algorithm is good for small graphs (with at most 5,000 edges). However for bigger graphs, the algorithm will run out of memory quickly. Thus, for bigger graphs, a new algorithm had to be created. The memory-friendly ratcatcher method presented in this paper has complexity $O(e^3)$, compared to the original ratcatcher method with a complexity of $O(e^2)$, where e is the number of edges of the original graph. We will refer to this method as the *memrat method*. The memrat method uses two ideas: The dual of the medial graph is bipartite; and by $A(2)$, for a region r incident with edges $e_1, e_2, \ldots$ and neighbors $r_1, r_2, \ldots$ respectively, the set $\phi(r) = \bigcap\{V(\phi_{r_i}(e_i)) \; \forall\, e_i\}$ contains the antipodality set for r if one exists, where $\phi_{r_i}(e_i)$ is the subgraph of $\phi(e)$, where every component of $\phi_{r_i}(e_i)$ has a node in the set $\phi(r_i)$.

For the memrat method, the all-pairs shortest-paths array and any information kept on the edges for the antipodality are discarded. Thus, a number of shortest-path calculations (breadth-first-search) are conducted to gather the necessary information. In addition, an array of bool (char for C users) is kept with the size of the nodes of the medial graph for the ϕ sets to test for membership for just one side of the bipartition of the medial graph (say the original nodes). For clarity, we will label these nodes red. For a region of $M(G)$, say r, let e be an edge incident with r and let r_o be the other region incident with e. To satisfy (A2) completely, a node v in $\phi(r)$ has also to be in $\phi(r_o)$ or there must exist a path in the components of $\phi(e)$ from v to some node in $\phi(r_o)$. If a node does not satisfy (A2) for r then delete that node from $\phi(r)$. Once every $\phi(r)$ satisfies (A2) for all regions r, then $\alpha(r) = \phi(r)$ for all regions r. If $\phi(r)$ becomes the null set then an antipodality of range at least k does not exist. An example of $\alpha(r)$, the dashed nodes, when r is the outside region of the medial graph in Figure 3, is illustrated in Figure 6.

Thus, to accomplish the goal of testing (A2), a queue is created containing every blue region. Once a region r is taken out of the queue, we create its ϕ set by the fact given in the previous paragraphs and we test (A2) on $\phi(r)$ by testing the aforementioned vertex and path condition on every edge incident to r. If $\phi(r)$ has changed, it is put at the end of the queue. Also, if for one of r's neighbors in $M(G)^*$, say r_o, $\phi(r_o)$ has changed, then all of r_o's neighbors are put in the stack if they are not already in the queue. We continue testing members of the queue until the queue is empty or there exists a region r such that $\phi(r)$ is empty. We will call this procedure *comptest*.

Theorem 5. *The procedure comptest terminates and an antipodality can be derived from the information if one exists.*

Proof. Since the ϕ set for all regions is finite and the fact that a region is put back in the queue only if one of its neighbors' ϕ set decreases, the procedure terminates. It is easy to see that if the procedure terminates with every region having nonempty ϕ sets, the resulting ϕ sets for the region satisfy (A2)

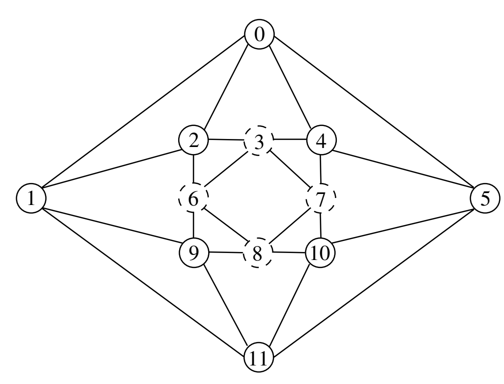


Figure 6 **Example $\alpha(r)$ for an Antipodality of Range Eight**

and are indeed the corresponding α sets. For the antipodalities of the edges, for each $e \in E(M(G))$ and a region r incident with e, define $\alpha(e)$ to be the union of all components of $\phi(e)$ that contain a node of $\phi(r)$. It is clear that α satisfies (A1), (A2), and (A3). Since one has to compute $\phi(e)$ for every edge e incident to r and the worst case that the antipodality does not exist, then the complexity of comptest is $O(|V(M(G))| * |E(M(G))| * |V(M(G)) + E(M(G))|)$. Furthermore, this complexity result is bounded by $O(e^3)$ where $e = |V(M(G))|$. $\square$

In addition, another algorithm was also created that kept the all-pairs shortest path array and $\phi(r)$ for every region r. This algorithm is called the *comprat method*. For each region r taken off the queue, comprat would verify the path condition between the two ϕ sets for every edge incident to r. If $\phi(r)$ gets changed, then r is put back onto the queue. If the ϕ set of region r_o, a neighbor of r, has been changed by the procedure and r_o is not on the queue, then r_o is put on the queue. Comprat has the same complexity as memrat but with addition of the array it should be inherently faster. For all three methods, an initial guess is used in the hope of being close to the carvingwidth. For the initial guess, we find the shortest eccentricity (the length of the longest shortest path) of one side of the bipartition (the original nodes) of nodes of the dual of the medial graph and initialize the guess to be twice that amount, minus four. If there exist an antipodality in $M(G)$ of this range, we then proceed to increment this value by two until an antipodality doesn't exist. If the initial guess fails, then we proceed to decrease this value by two until an antipodality does exist.

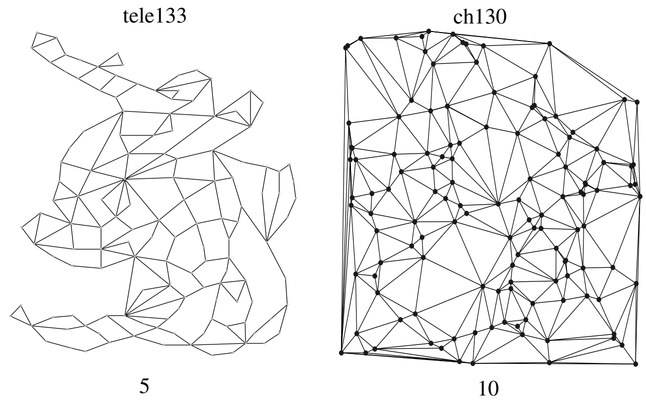


Figure 7 Some Test Instances with Their Corresponding Branchwidth

5. Computational Results

The test instances are partitioned into two classes: telecommunications (T) and Delaunay (D). The members of the telecom class are maximum planar subgraphs of test instances from the telecommunications industry provided by Bill Cook at Georgia Tech. The maximum planar subgraphs were derived using a branch-and-cut scheme developed by Hicks (2004b). The other test instances are Delaunay triangulations of some test instances from the TSPLIB (Reinelt 1991). For more information about Delaunay triangulations, the reader is referred to Edelsbrunner (1987). Figures 7, 8, and 9 illustrate some of the test instances.

Table 1 offers a summary of the computational results by offering the runtimes (rounded to the nearest integer if greater than one) of the ratcatcher, comprat, and memrat methods on a representative group.

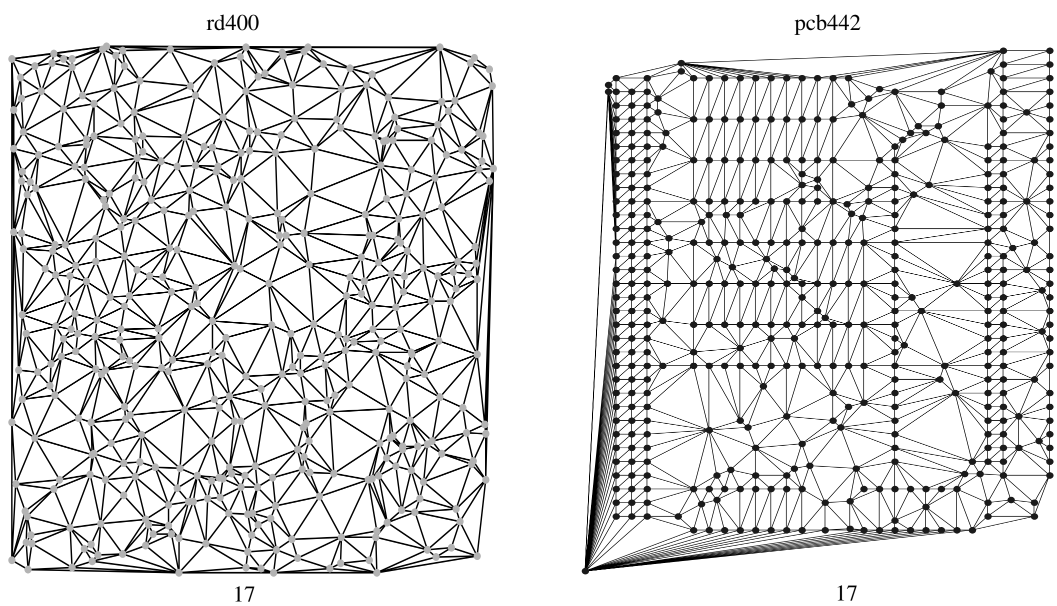
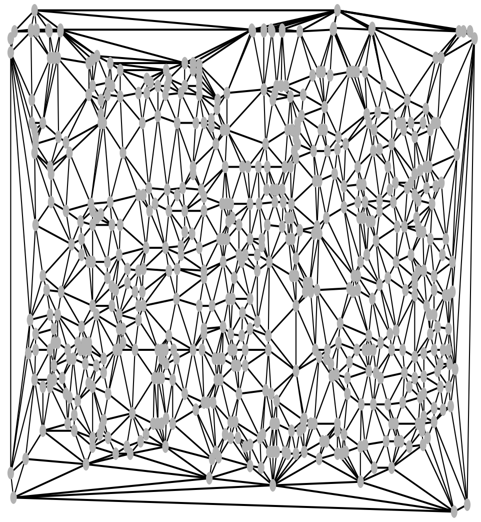


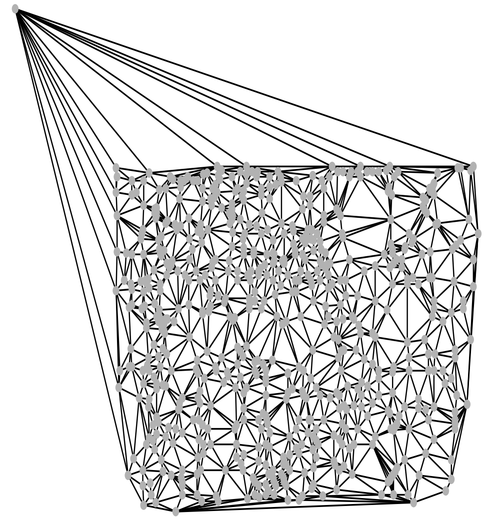
Figure 8 Some Test Instances with Their Corresponding Branchwidth

u574



17

d657



22

Figure 9 Some Test Instances with Their Corresponding Branchwidth

Table 1 Comparison of the Three Methods

Graphs	Class	Nodes	Edges	$\beta(G)$	iter	rat (sec)	comprat (sec)	memrat (sec)
tele39	T	39	81	4	3	0.3	0.3	0.5
tele53	T	53	91	5	4	0.5	0.5	0.9
tele56	T	56	85	3	2	0.2	0.2	0.3
tele62	T	62	129	4	2	0.5	0.5	0.7
tele89	T	89	136	4	3	0.4	0.5	0.8
tele133	T	133	212	5	1/1/3	2	3	4
tele226	T	226	2,286	4	2	1	2	2
eil51	D	51	140	8	4	1	1	2
lin105	D	105	292	8	3	4	7	9
ch130	D	130	377	10	4	9	13	18
pr144	D	144	393	9	4	9	12	19
kroB150	D	150	436	10	4	12	18	24
pr226	D	226	586	7	3	17	22	31
tsp225	D	225	622	12	4	22	29	46
a280	D	280	788	13	4/4	17	22	35
pr299	D	299	864	11	4/3	21	34	45
rd400	D	400	1,183	17	3	65	112	136
pcb442	D	442	1,286	17	4	90	135	211
u574	D	574	1,708	17	3	145	226	298
p654	D	654	1,806	10	3	161	198	276
d657	D	657	1,958	22	4	224	396	545
pr1002	D	1,002	2,972	21	2	338	448	562
rl1323	D	1,323	3,950	22	3	876	1,519	1,590
d1655	D	1,655	4,890	29	3	1,318	1,608	2,206
rl1889	D	1,889	5,631	22	3	M	3,931	4,012
u2152	D	2,152	6,312	31	4	M	3,207	4,704
pr2392	D	2,392	7,125	29	3	M	3,813	5,167
pcb3038	D	3,038	9,101	40	4	M	13,817	15,865
fl3795	D	3,795	11,326	25	3	M	18,469	17,142
fnl4461	D	4,461	13,359	48	4	M	35,933	51,035
rl5934	D	5,934	17,770	41	3	M	73,468	66,461
pla7397	D	7,397	21,865	33	2	M	65,197	53,564
usa13509	D	13,509	40,503	63	1/2	M	M	413,861
brd14051	D	14,051	42,128	68	3	M	M	594,468

All three methods were implemented using the C++ language. All computations for Table 1 were performed on a SGI Power Challenge with 6×194 MHz processors with a gigabyte of memory and a gigabyte of swap space. The column labeled "iter" gives the number of iterations needed to compute the branchwidth of the graph to show performance of the initial guess and to offer an estimate of the runtime per iteration for each method. In addition, forward slashes partition the number of iterations needed for each biconnected component of the input graph. Also an "M" in the table means that the code ran out of memory on that particular test instance.

As one can see from Table 1, the initial guess works well (the minimum possible number for iterations is two if the branchwidth is not two). In addition, ratcatcher is faster than the other methods but only by a factor of at most 1.77 for comprat and at most 2.43 for memrat. Thus, each method is good for a particular class of graphs depending on the number of edges. With the inclusion of the all-pairs shortest-path matrix, comprat is good only for graphs with about 20,000 edges, as illustrated in Table 1. Understand that if the original graph is biconnected with 20,000 edges, then the medial graph has 20,000 nodes, 40,000 edges and 20,002 regions. Thus, the medial graph is about twice as large as the original graph.

Furthermore, as the sizes of the graphs get closer to the memory limitations for the comprat method, memrat outperforms comprat. This could be due to the time spent using the swap space. Since the memrat method is structured to handle big graphs, Table 1 illustrates that the memrat method can handle graphs with edges with at least 40,000 edges. For brd14051, memrat uses about 600 megabytes of memory. Thus, the estimated limit for a computer with one gigabyte of memory would be a graph with about 70,000 edges. In contrast, the runtime for the memrat method for brd14051 is not as impressive. As illustrated in Table 1, the memrat method took over 165 hours for completion. Thus, the memrat method took about 55 hours per iteration for this test instance. Obviously, all three methods illustrate the classic conflict between memory and runtime speed.

6. Conclusions and Future Work

In conclusion, the ratcatcher method can calculate the branchwidth of planar graphs having up to 5,000 edges while comprat and memrat can handle planar graphs having up to 20,000 and 70,000 edges respectively. Thus, each method is essential for a particular class of planar graphs.

There are several directions for future work in the area of this paper. First, it seems that an antipodality would exists if (A3) were rephrased to "every bond using e and f has cardinality $\geq k$." This would probably be the first step toward finding an analogue to antipodalities for medial graphs on other orientable surfaces. Another direction would be to use the antipodalities to construct an optimal branch decomposition. For motivation, in a connected antipodality, there is a bond in $M(G)$ (a separation in G) associated with the $\alpha(r)$ for every region r of $M(G)$. Using only the ratcatcher method to produce an optimal branch decomposition would decrease the complexity of finding an optimal branch decomposition of a planar graph which is currently known to be $O(e^4)$ (Hicks 2005). In addition, recent results by Tamaki (2003) may offer insight toward this challenging problem. Finally, another direction for research is finding an algorithm to compute the optimal treewidth of a planar graph; see Robertson and Seymour (1984) and Bouchitte et al. (2001) for related background.

Acknowledgments

The author would like to acknowledge Bill Cook for his guidance and patience. The author would also like to thank Nate Dean and Paul Seymour for conversations about the subject. In addition, this effort could have not been accomplished without the planarity testing codes provided by David Applegate and John Boyer (see Boyer and Myrvold 1999 for a description of the algorithm). Applegate's code was used for earlier versions of the methods, while Boyer's code was used for the recent versions of the methods. This research was partially supported by NSF Grant DMI-0217265, ONR Grant N00014-98-1-0014, and by Texas ATP Grant 99-3604-0034. The author was also supported by the AT&T Labs Fellowship Program as a graduate student in the Computational and Applied Mathematics Department at Rice University. Finally, the author would like to thank the anonymous referees that gave valuable advice for the presentation of the results.

References

Alber, J., R. Niedermeier. 2002. Improved tree decomposition based algorithms for domination-like problems. *Fifth Latin American Theoret. Informatics* (*LATIN 2002*). Springer-Verlag, Berlin, Germany, 613–627.

Alvarez, C., R. Cases, J. Diaz, J. Petit, M. Serna. 2000. Routing tree problems on random graphs. Technical report LSI-01-10-R, Software Department, Universitat Politecnica de Catalunya, Barcelona, Spain.

Archdeacon, D. 1980. A Kuratowski theorem for the projective plane. Ph.D. thesis, Mathematics Department, Ohio State University, Columbus, OH.

Archdeacon, D., P. Huneke. 1989. A Kuratowski theorem for nonorientable surfaces. *J. Combin. Theory, Ser. B* **46** 173–231.

Arnborg, S., J. Lagergren, D. Seese. 1991. Easy problems for tree-decomposable graphs. *J. Algorithms* **12** 308–340.

Bienstock, D. 1991. Graph searching, path-width, tree-width and related problems. *Reliability of Computer and Communication Networks. Series Discrete Math. Theoret. Comput. Sci.* (*DIMACS*), Vol. 5. American Mathematical Society, Providence, RI, 33–49.

Bodlaender, H. 1993. A tourist guide through treewidth. *Acta Cybernetica* **11** 1–21.

Bodlaender, H. 1996. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.* **25** 1305–1317.

Bodlaender, H. 1997. Treewidth: Algorithmic techniques and results. 22*nd Internat*. *Sympos*. *Math*. *Foundations Comput*. *Sci*. (*MFCS* 1997). *LNCS*, No. 1295. Springer-Verlag, Berlin, Germany, 29–36.

Bodlaender, H. 1998. A partial k-arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.* **209** 1–45.

Bodlaender, H., T. Kloks. 1996. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *J. Algorithms* **21** 358–402.

Bodlaender, H., D. Thilikos. 1997. Constructive linear time algorithms for branchwidth. 24*th Internat*. *Colloquium Automata, Languages*, *Programming*. *LNCS*, No. 1256. Springer-Verlag, Berlin, Germany, 627–637.

Bodlaender, H., D. Thilikos. 1999. Graphs with branchwidth at most three. *J. Algorithms* **32** 167–194.

Borie, R., R. Parker, C. Tovey. 1992. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica* **7** 555–581.

Bouchitte, V., F. Mazolt, I. Todinca. 2001. Chordal embeddings of planar graphs. Technical report RR-2001-03, Laboratoire d'Informatique Fondametale d'Orléans, Orleans, France.

Boyer, J., M. Myrvold. 1999. Stop minding your P's and Q's: A simplified O(n) planar embedding algorithm. *Tenth Annual ACM-SIAM Sympos. Dicrete Algorithms.* ACM, New York, 140–146.

Christian, W. 2003. Linear-time algorithms for graphs with bounded branchwidth. Ph.D. thesis, Computational and Applied Mathematics Department, Rice University, Houston, TX.

Cook, W., P. Seymour. 2003. Tour merging via branch-decomposition. *INFORMS J. Comput.* **15** 233–248.

Courcelle, B. 1990. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Inform. Comput.* **85** 12–75.

Edelsbrunner, H. 1987. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, Germany.

Geelen, J., A. Gerards, G. Whittle. 2002. Branch width and well-quasi-ordering in matroids and graphs. *J. Comb. Theory, Ser. B* **84** 270–290.

Glover, H., P. Huneke, C. Wang. 1979. 103 graphs that are irreducible for the projective plane. *J. Comb. Theory, Ser. B* **27** 332–370.

Hicks, I. V. 2000. *Branch Decompositions and their Applications*. Ph.D. thesis, Computational and Applied Mathematics Department, Rice University, Houston, TX.

Hicks, I. V. 2002. Branchwidth heuristics. *Congressus Numerantium* **159** 31–50.

Hicks, I. V. 2004a. Branch decompositions and minor containment. *Networks* **43** 1–9.

Hicks, I. V. 2004b. New facets for the planar subgraph polytope. Working paper, Industrial Engineering Department, Texas A&M University, College Station, TX.

Hicks, I. V. 2005. Planar branch decompositions II: The cycle method. *INFORMS J. Comput.* **17** 418–421.

Kloks, T., J. Kratochvil, H. Müller. 1999. New branchwidth territories. *Sixteenth Annual Sympos. Theoret. Aspects Comput. Sci. Trier, Germany, March 1999.* Springer-Verlag, Berlin, 173–183.

Koster, A., H. Bodlaender, S. van Hoesel. 2001. Treewidth: Computational experiments. Technical report 01-38, Konrad-Zuge, Zentrum für Informationstechnik Berlin, Berlin, Germany.

Koster, A., S. van Hoesel, A. Kolen. 2002. Solving partial constraint satisfaction problems with true decomposition. *Networks* **40** 170–180.

Kuratowski, K. 1930. Sur le probleme des courbes gauches en topologie. *Fundamenta Math.* **15** 271–283.

Ramachandramurthi, S. 1997. The structure and number of obstructions to treewidth. *SIAM J. Discrete Math.* **10** 146–157.

Reed, B. 1992. Finding approximate separators and computing tree width quickly. *Twenty Fourth Annual ACM Sympos. Theory Comput.* ACM Press, New York, 221–228.

Reed, B. 1997. Tree width and tangles: A new connectivity measure and some applications. R. Bailey, ed. *Survey in Combinatorics*. Cambridge University Press, Cambridge, U.K.

Reinelt, G. 1991. TSPLIB—A traveling salesman library. *ORSA J. Comput.* **3** 376–384.

Robertson, N., P. Seymour. 1983. Graph minors I: Excluding a forest. *J. Combin. Theory, Ser. B* **35** 39–61.

Robertson, N., P. Seymour. 1984. Graph minors III: Planar tree-width. *J. Combin. Theory, Ser. B* **36** 49–64.

Robertson, N., P. Seymour. 1985. Graph minors: A survey. *Survey in Combinatorics*. Cambridge University Press, Cambridge, U.K., 153–171.

Robertson, N., P. Seymour. 1991. Graph minors X: Obstructions to tree-decompositions. *J. Combin. Theory, Ser. B* **52** 153–190.

Robertson, N., P. Seymour. 1994. Graph minors XI: Circuits on a surface. *J. Combin. Theory, Ser. B* **60** 72–106.

Robertson, N., P. Seymour. 1995. Graph minors XIII: The disjoint paths problem. *J. Combin. Theory, Ser. B* **63** 65–110.

Seymour, P., R. Thomas. 1993. Graph searching and a min-max theorem for tree-width. *J. Combin. Theory, Ser. B* **58** 22–33.

Seymour, P., R. Thomas. 1994. Call routing and the ratcatcher. *Combinatorica* **14** 217–241.

Tamaki, H. 2003. A linear time heuristic for the branch-decomposition of planar graphs. Technical report MPI-I-2003-1-010, Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Telle, J., A. Proskurowski. 1997. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discrete Math.* **10** 529–550.

Thomas, R. 1990. A Menger-like property of tree-width: The finite case. *J. Combin. Theory, Ser. B* **48** 67–76.

Wagner, K. 1937. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen* **115** 570–590.

West, D. 2001. *Introduction to Graph Theory*. Prentice-Hall, Upper Saddle River, NJ.