Branch-width and Tangles

Illya V. Hicks*
Department of Computational and Applied Mathematics
Rice University
Houston, Texas 77005-1892, USA.

Sang-il Oum†‡
Department of Mathematical Sciences
KAIST
Daejeon, 305-701, Republic of Korea.

February 21, 2010

Abstract

This article describes the notion of branch-width and its dual notion, tangles. Branch-width was introduced by Robertson and Seymour and has been applied to various combinatorial structures.

Keyword: branch-width; carving-width; rank-width; tangle

Branch-width, introduced by Robertson and Seymour [41], is a general concept to describe the difficulty of decomposing finitely many objects into a tree-like structure by partitioning them into two parts recursively, while maintaining each cut to have small connectivity measure. Branch-width normally is defined for graphs or hypergraphs, as discussed by Robertson and Seymour [41] but it is easy to be extended for other combinatorial objects such as matroids and any integer-valued symmetric submodular functions.

Roughly speaking branch-decomposition is a description on a maximal collection of non-overlapping partitions of a finite set E. The width of a branch-decomposition is the maximum "complexity" of each part appearing in the branch-decomposition, where the "complexity" is given by some function on subsets of E. The branch-width is the minimum possible width over all possible branch-decompositions of E. Precise definition will be discussed in the following section.

To show that branch-width is small, we need to illustrate how to decompose nicely; in other words, we need to present a branch-decomposition of small width in order to certify that branch-width is small. On the other hand, if we want to certify that branch-width is large, a naive approach would be trying all possible branch-decompositions, which is too time consuming. For that purpose

*Partially Supported by National Science Foundation CMMI-0926618.
†sangil@kaist.edu
‡Partially Supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0063183) and by TJ Park Junior Faculty Fellowship.

we use tangles. A tangle is a dual notion of branch-width which certifies why the branch-width is large. It was also defined by Robertson and Seymour in the same paper.

In this article we explain those definitions and list their algorithmic properties.

1 Branch-width

Usually branch-width is defined for graphs and hypergraphs. But for the sake of generality, we define it for integer-valued symmetric submodular functions first. An integer-valued function f on subsets of a finite set E is *symmetric* if $f(X) = f(E - X)$ for all subsets X of E and f is called *submodular* if $f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y)$ for all subsets X, Y of E.

Let us now assume that an integer-valued symmetric submodular function f on subsets of a finite set E is given. We call a tree *subcubic* if every vertex has degree 3 or 1. A *branch-decomposition* (T, τ) of f consists of a subcubic tree T and a bijection τ from the set of leaves of T to E. Then the *width* of an edge e of T is defined to be $f(\tau(A_e))$ when (A_e, B_e) is a partition of the set of leaves of T given by $T \setminus e$. Notice that this is well-defined because $f(\tau(A_e)) = f(\tau(B_e))$. The *width* of a branch-decomposition (T, τ) is the maximum width of all edges of T. The *branch-width* of f, denoted by $\mathrm{bw}(f)$, is the minimum width of all possible branch-decompositions of f. If $|E| \leq 2$, then there are no branch-decompositions and so we just define branch-width to be $f(\emptyset)$.

By choosing an appropriate set E and an integer-valued symmetric submodular function, we can generate various notions of width parameters. Let us present some of them here.

Branch-width of graphs and hypergraphs. Branch-width was first introduced by Robertson and Seymour [41] for graphs and hypergraphs. For a graph (or a hypergraph) G and a subset X of edges, let $\eta_G(X)$ be the number of vertices which are incident with an edge in X as well as an edge in $E(G) - X$. It is straightforward to prove that η_G is a symmetric submodular function on subsets of $E(G)$. The branch-width of G, denoted by $\mathrm{bw}(G)$, is defined as the branch-width of η_G.

For example, consider the Petersen graph and its optimal branch-decomposition in Figure 1. The width of the edge e given in Figure 1 is 4. Furthermore, one can evaluate the widths of the other edges of (T, τ) and determine that the width of (T, τ) is 4.

Branch-width of graphs is strongly related to better-known notion, tree-width by the following inequality by Robertson and Seymour [41, (5.2)]: if G is a graph, then

$$\text{branch-width}(G) \leq \text{tree-width}(G) + 1 \leq \frac{3}{2}\,\text{branch-width}(G).$$

Rank-width of graphs. Rank-width of graphs was introduced by Oum and Seymour [37]. For a graph G and a subset X of $V = V(G)$, let us consider the $|X| \times |V - X|$ binary matrix M_X such that rows and columns of M_X are indexed by X and $V - X$, respectively and the entry of M_X is 1 if the vertex corresponding to the row is adjacent to the vertex corresponding to the column, and otherwise, the entry is 0. The *cut-rank* function $\rho_G(X)$ is defined to be the rank of M_X, where M_X is considered as a matrix over the binary field GF(2). The cut-rank function is symmetric submodular, see [37]. The *rank-width* of a graph is defined as the branch-width of ρ_G.

Rank-width was motivated by another useful graph width parameter, *clique-width*, defined by Courcelle and Olariu [6]. They are related in the following sense; if the clique-width of a graph is k, then its rank-width is at most k and conversely if the rank-width of a graph is r, then the

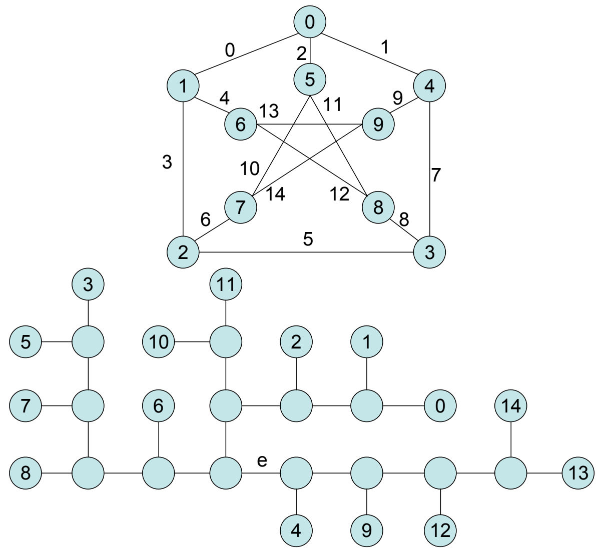

Figure 1: The Petersen graph and its optimal branch-decomposition

clique-width is at most $2^{r+1} - 1$ [37]. Oum [35] showed that the rank-width of a graph G is less than or equal to the branch-width of G, unless G has no edges.

Branch-width of matroids. Unlike tree-width, it is natural to extend the notion of branch-width of graphs to branch-width of matroids. For a matroid M on a finite set E with the rank function r, the *connectivity function* of M is given as $\eta_M(X) = r(X) + r(E - X) - r(M) + 1$. Since r is submodular, η_M is symmetric submodular. Branch-width of a matroid M is defined to be the branch-width of η_M. It was first studied by Dharmatilake [8] and has played an important role in the development of the matroid structure theory by Geelen, Gerards, and Whittle [16, 17].

If a graph G has at least one cycle of length at least 2, then G and its cycle matroid $M(G)$ has the same branch-width, shown by Hicks and McMurray Jr. [24] and independently by Mazoit and Thomassé [34] later.

Carving-width of graphs. Carving-width of graphs was introduced by Seymour and Thomas [42]. For a graph G and a subset A of vertices, we write $\delta_G(A)$ to denote the set of all edges joining a vertex in A with a vertex in $V(G) - A$. Let $p_G(X) = |\delta_G(A)|$. Again p_G is symmetric submodular. The *carving-width* of a graph is the branch-width of p_G. Carving-width is a useful tool for the branch-width of a planar graph because the branch-width of a planar graph is exactly half of the carving-width of its medial graph [42].

2 Tangles

Tangles are introduced as a means to certify that the branch-width is large. If we wish to convince that branch-width is small, we can simply present a branch-decomposition of small width. However,

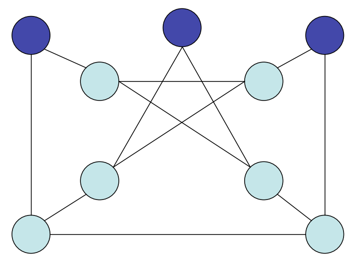

Figure 2: A “large” part in a order-4 tangle of the Petersen graph

we do not want to try all possible branch-decompositions in order to convince that branch-width is big. Tangles play such a role; if a tangle is presented, then no branch-decomposition of small width can exist.

For an integer-valued symmetric submodular function f on subsets of a finite set E, an f-tangle of order $k+1$ is a collection $\mathcal{T}$ of subsets of E satisfying the following three axioms.

(T1) For all $A \subseteq E$, if $f(A) \leq k$, then either $A \in \mathcal{T}$ or $E - A \in \mathcal{T}$.

(T2) If $A, B, C \in \mathcal{T}$, then $A \cup B \cup C \neq E$.

(T3) For all $e \in E$, we have $E - \{e\} \notin \mathcal{T}$.

Let us call a set X in a tangle $\mathcal{T}$ *small* and the complement $E - X$ *large*. Informally speaking, a large set is a “highly connected” set so that it is impossible to decompose a large set properly to construct a branch-decomposition of small width. In Figure 2, we illustrate a large set in a tangle of order 3 for the Petersen graph. Edges shown in Figure 2 form a large set.

Robertson and Seymour introduced tangles and proved lots of useful properties. The following duality theorem is very useful. The following theorem was implicitly proved by Robertson and Seymour [41, (3.5)]. Geelen et al. [19, Theorem 3.2] rewrote the proof.

Theorem 1. *Let f be an integer-valued symmetric submodular function on subsets of E. Then no f-tangle of order $k+1$ exists if and only if the branch-with of f is at most k.*

This allows us to define the branch-width from tangles; the branch-width is equal to the maximum k such that a tangle of order k exists. And to show that $\text{bw}(f) = k$ for an integer k, we frequently construct both a branch-decomposition of width k for an upper bound on the branch-width and an f-tangle of order k for a lower bound.

Providing a lower bound for the branch-width is generally harder than finding an upper bound. Therefore much of the work to find the exact branch-width is usually devoted to finding a tangle. For the branch-width of the $n \times n$ grid, Kleitman and Saks (in Robertson and Seymour [41]) presented a tangle of order n, thus proving that the branch-width of the $n \times n$ grid is n. Geelen et al. [18] used tangles to prove that the branch-width of the cycle matroid of the $n \times n$ grid is n. For the rank-width of the $n \times n$ grid G, Jelínek [30] presented a ρ_G-tangle of order $n-1$, thus certifying that the rank-width of the $n \times n$ grid is $n-1$.

Roughly speaking a set of maximal tangles is used to identify highly connected pieces in a combinatorial structure. Robertson and Seymour [41] (see also Geelen et al. [18]) showed that any symmetric submodular function on E has at most $(|E|-2)/2$ maximal tangles, which are displayed by a tree structure. That tree structure has been used to describe and prove the structure of graphs or binary matroids without some fixed minor.

3 Computing branch-width

One of the most natural questions after defining branch-width is the complexity of computing the branch-width of integer-valued symmetric submodular functions on subsets of a finite set E. Since we may need 2^n values of f for all subsets of E in order to input f, we will assume that f is given by an oracle so that we can query the oracle to compute $f(X)$ for the input set X at a unit time.

Hardness results. In general, it is hard to decide whether branch-width is at most k for an integer-valued symmetric submodular function f given by an oracle and an input k in time polynomial in n. Seymour and Thomas [42] showed that it is NP-hard to compute branch-width or carving-width of a graph. Kloks et al. [31] proved that computing branch-width is NP-hard even for bipartite graphs or split graphs. Computing branch-width of a matroid given as a matrix representation is also NP-hard and computing rank-width of a graph is also NP-hard, because of the relationship between branch-width of graphs and branch-width of cycle matroids [24, 34].

Exact exponential-time algorithms. For the efficient exact algorithm, Oum [36] found an $O^*(2^{|E|})$-time algorithm to compute the branch-width of any integer-valued symmetric submodular function f given by an oracle as above. (Here, $O^*(2^{|E|})$ means $O(2^{|E|}|E|^{O(1)})$.) It is not known whether $O^*(2^{|E|})$ can be improved to $O^*(c^{|E|})$ for some $1 < c < 2$. For graphs $G = (V, E)$, branch-width can be computed in time $O^*((2\sqrt{3})^{|V|})$, shown by Fomin et al. [14].

Exact polynomial-time algorithms for special classes. When we restrict inputs, the branch-width can sometimes be computed efficiently. Branch-width can be computed in polynomial time for circular arc graphs [33] and interval graphs [31, 39]. For planar graphs, branch-width and carving-width can be computed in polynomial time, shown by Seymour and Thomas [42]. More precisely their algorithm can decide in time $O(n^2)$ whether a given planar graph has branch-width at most k for a given k and output an optimal decomposition in time $O(n^4)$. Gu and Tamaki [20] improved that result to construct an $O(n^3)$-time algorithm to output an optimal carving-decomposition or an optimal branch-decomposition of n-vertex planar graphs.

Testing branch-width at most k for fixed k. As we discussed above, we can not hope to have a polynomial-time algorithm to test whether branch-width is at most k for an input k. However, if we fix k as a constant, then the situation is different. Oum and Seymour [38] proved that for any fixed constant k, one can answer whether the branch-width is at most k in time $O(|E|^{8k+c})$ where c only depends on $f(\emptyset)$. Moreover one can construct a branch-decomposition of width at most k in time $O(|E|^{8k+c+3})$.

For many applications on fixed-parameter tractable algorithms, it is desirable to have an algorithm which runs in time $O(g(k)n^c)$ for some function g and a constant c independent of k. Such an algorithm is called a *fixed-parameter tractable* algorithm with parameter k. It is still unknown whether there is a fixed-parameter tractable algorithm to decide whether branch-width of f is at most k when f is an integer-valued symmetric submodular function given as an oracle.

Fortunately fixed-parameter tractable algorithms are known for most interesting classes of integer-valued symmetric submodular functions. Bodlaender and Thilikos [43, 1] constructed a linear-time algorithm to test whether branch-width of an input graph is at most k for fixed k. Thilikos et al. [44] constructed a linear-time algorithm to decide whether carving-width is at most k for fixed k. Hliněný and Oum [29] showed that there exists a cubic-time algorithm to decide whether rank-width of a graph is at most k for fixed k. Their algorithm also works for branch-width of matroids represented over a fixed finite field. All of these algorithms mentioned above can output the corresponding branch-decomposition as well.

Fixed-parameter tractable approximation algorithms. For applications on fixed-parameter tractable algorithms with the branch-width as a parameter, we often need an fixed-parameter tractable algorithm to construct a branch-decomposition of small width in order to use the dynamic programming approach. So far, we do not know the existence of a fixed-parameter tractable algorithm that can output a branch-decomposition of width at most k if such a branch-decomposition exists, for an integer-valued symmetric submodular function given by an oracle. As we discussed above, the best algorithm known runs in time $O(|E|^{8k+c+3})$.

As a workaround, Oum and Seymour [37] constructed the following algorithm: for each fixed k, it runs in time $O(|E|^7 \log |E|)$ to either output a branch-decomposition of width at most $3k + c'$ or confirm that the branch-width is larger than k, where c' only depends on $f(\emptyset)$ and $\max\{f(\{e\} : e \in E\}$. (In fact, the paper [37] only discusses the case when $f(\emptyset) = 0$ and $f(\{e\}) \leq 1$ for all $e \in E$. But its argument can be modified to accommodate the case when there is an element $e \in E$ such that $f(\{e\}) - f(\emptyset) > 1$.) This allows us to construct a branch-decomposition of small width from the given adjacency list of a graph, and this branch-decomposition can be used to solve other algorithmic problems by dynamic programming technique.

There are similar algorithms for branch-width of matroids represented over a finite field [25].

Heuristics. Cook and Seymour [3, 4] gave a heuristic algorithm to produce branch-decompositions of graphs and used it in their work on the ring-routing problem and the traveling salesman problem. Hicks [21] also found another branch-width heuristic that was comparable to the heuristic of Cook and Seymour. Recently, Ma and Hicks [32] found two heuristics to derive near-optimal branch-decompositions of linear matroids; one of the heuristics uses classification techniques and the other one is similar to the heuristics for graphs which use flow algorithms.

4 Algorithmic Applications

Branch-width of graphs. There are many graph-theoretic algorithmic problems that are shown to be polynomial-time solvable on the class of graphs of bounded branch-width. Many of them actually run their algorithms based on tree-width. We refer to the section on tree-width for such applications.

Branch-width is used to design exact subexponential-time algorithms or efficient parameterized algorithms on the class of planar graphs or the class of graphs with no fixed minor [13, 10, 9, 15, 11, 12].

Branch-width of matroids. Hliněný [26] extended Courcelle's theorem on graphs of bounded tree-width or branch-width to matroids represented over a fixed finite field. Namely, for a fixed finite field F and a given monadic second-order formula φ on matroids, one can test whether an input F-represented matroid of bounded branch-width satisfies φ in time polynomial in the size of the matroid. The requirement that the matroid has to be represented over a finite field can not be relaxed unless NP=P, shown by Hliněný [28].

Hliněný [27] also found a fixed-parameter tractable algorithm to evaluate the Tutte polynomial of an input matroid represented over a fixed finite field of bounded branch-width.

Rank-width of graphs. Rank-width is a sibling of better known clique-width, that is a kind of a generalization of tree-width. Oum and Seymour [37] proved not only that for every class of graphs, rank-width is bounded if and only if clique-width is bounded, but also that one can translate a rank-decomposition into a decomposition for clique-width and vice versa in polynomial time. It had been known that many algorithmic properties of tree-width could be generalized to graphs of bounded clique-width, even before rank-width was introduced and it is easy to see that all of such algorithmic results on graphs of bounded clique-width apply to rank-width.

Here is one of the most important theorems for graphs of bounded rank-width. Courcelle, Makowsky, and Rotics [5] proved that there is a cubic-time algorithm to decide whether a fixed monadic second-order formula without edge-set quantification is satisfied by an input graph of bounded rank-width. As a corollary, many hard problems such as 3-colorability are solvable in a cubic time for graphs of bounded rank-width.

Practical algorithms. Although theory indicates the fruitful potential of these algorithms, the number of practical algorithms in the literature is scant. Most notable is the work of Cook and Seymour [4] who produced the best known solutions for the 12 unsolved problems in TSPLIB95, a library of standard test instances for the traveling salesman problem [40]. Hicks presented a practical algorithm for general graph minor containment [22] and constructing optimal branch decompositions [23]. One is also referred to the work of Christian [2].

Based on branch-width of matroids, Cunningham and Geelen [7] proposed a pseudopolynomial-time algorithm to solve an integer programming problem $\max(c^t x : Ax = b, x \geq 0, x \in \mathbb{Z}^n)$ when A is nonnegative and the matroid represented by A has bounded branch-width. Their algorithm shows some hope to make branch-width much more useful for practical applications, as many problems are modelled as an integer programming.

References

[1] H. L. Bodlaender and D. M. Thilikos. Constructive linear time algorithms for branchwidth. In *Automata, languages and programming (Bologna, 1997)*, volume 1256 of *Lecture Notes in Comput. Sci.*, pages 627–637. Springer, Berlin, 1997.

[2] W. Christian. *Linear-Time Algorithms for Graphs with Bounded Branchwidth.* PhD thesis, Rice University, 2003.

[3] W. Cook and P. Seymour. An algorithm for the ring-router problem. Technical report, Bellcore, 1994.

[4] W. Cook and P. Seymour. Tour merging via branch-decomposition. *INFORMS J. Comput.*, 15(3):233–248, 2003.

[5] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[6] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Appl. Math.*, 101(1-3):77–114, 2000.

[7] W. H. Cunningham and J. Geelen. On integer programming and the branch-width of the constraint matrix. In M. Fishetti and D. Williamson, editors, *Proceedings of the 13th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007*, volume 4513 of *Lecture Notes in Computer Science*, pages 158–166, 2007.

[8] J. S. Dharmatilake. *Binary matroids of branch-width* 3. PhD thesis, Ohio State University, 1994.

[9] F. Dorn. Dynamic programming and fast matrix multiplication. In *Algorithms—ESA 2006*, volume 4168 of *Lecture Notes in Comput. Sci.*, pages 280–291. Springer, Berlin, 2006.

[10] F. Dorn, F. V. Fomin, and D. M. Thilikos. Fast subexponential algorithm for non-local problems on graphs of bounded genus. In *Algorithm theory—SWAT 2006*, volume 4059 of *Lecture Notes in Comput. Sci.*, pages 172–183. Springer, Berlin, 2006.

[11] F. Dorn, F. V. Fomin, and D. M. Thilikos. Subexponential parameterized algorithms. In *Automata, languages and programming*, volume 4596 of *Lecture Notes in Comput. Sci.*, pages 15–27. Springer, Berlin, 2007.

[12] F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 631–640, New York, 2008. ACM.

[13] F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA 2005)*, volume 3669 of *Lecture Notes in Comput. Sci.*, pages 95–106, 2005.

[14] F. Fomin, F. Mazoit, and I. Todinca. Computing branchwidth via efficient triangulations and blocks. *Discrete Appl. Math.*, 157(12):2726–2736, 2009.

[15] F. V. Fomin and D. M. Thilikos. Dominating sets in planar graphs: branch-width and exponential speed-up. *SIAM J. Comput.*, 36(2):281–309 (electronic), 2006.

[16] J. Geelen, B. Gerards, and G. Whittle. Towards a structure theory for matrices and matroids. In *International Congress of Mathematicians. Vol. III*, pages 827–842. Eur. Math. Soc., Zürich, 2006.

[17] J. Geelen, B. Gerards, and G. Whittle. Towards a matroid-minor structure theory. In *Combinatorics, complexity, and chance*, volume 34 of *Oxford Lecture Ser. Math. Appl.*, pages 72–82. Oxford Univ. Press, Oxford, 2007.

[18] J. Geelen, B. Gerards, and G. Whittle. Tangles, tree-decompositions and grids in matroids. *J. Combin. Theory Ser. B*, 99(4):657–667, 2009.

[19] J. F. Geelen, B. Gerards, N. Robertson, and G. Whittle. Obstructions to branch-decomposition of matroids. *J. Combin. Theory Ser. B*, 96(4):560–570, 2006.

[20] Q.-P. Gu and H. Tamaki. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. *ACM Trans. Algorithms*, 4(3):Art. 30, 13, 2008.

[21] I. V. Hicks. Branchwidth heuristics. *Congressus Numerantium*, 159:31–50, 2002.

[22] I. V. Hicks. Branch decompositions and minor containment. *Networks*, 43(1):1–9, 2004.

[23] I. V. Hicks. Graphs, branchwidth, and tangles! Oh my! *Networks*, 45(2):55–60, 2005.

[24] I. V. Hicks and N. B. McMurray Jr. The branchwidth of graphs and their cycle matroids. *J. Combin. Theory Ser. B*, 97(5):681–692, 2007.

[25] P. Hliněný. A parametrized algorithm for matroid branch-width. *SIAM J. Comput.*, 35(2):259–277, loose erratum (electronic), 2005.

[26] P. Hliněný. Branch-width, parse trees, and monadic second-order logic for matroids. *J. Combin. Theory Ser. B*, 96(3):325–351, 2006.

[27] P. Hliněný. The Tutte polynomial for matroids of bounded branch-width. *Combin. Probab. Comput.*, 15(3):397–409, 2006.

[28] P. Hliněný. On some hard problems on matroid spikes. *Theory Comput. Syst.*, 41(3):551–562, 2007.

[29] P. Hliněný and S. Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008.

[30] V. Jelínek. The rank-width of the square grid. *Discrete Appl. Math.*, 2009. doi:10.1016/j.dam.2009.02.007.

[31] T. Kloks, J. Kratochvíl, and H. Müller. Computing the branchwidth of interval graphs. *Discrete Appl. Math.*, 145(2):266–275, 2005.

[32] J. Ma and I. V. Hicks. Branchwidth heuristics for linear matroids. preprint, 2009.

[33] F. Mazoit. The branch-width of circular-arc graphs. In *LATIN 2006: Theoretical informatics*, volume 3887 of *Lecture Notes in Comput. Sci.*, pages 727–736. Springer, Berlin, 2006.

[34] F. Mazoit and S. Thomassé. Branchwidth of graphic matroids. In A. Hilton and J. Talbot, editors, *Surveys in Combinatorics 2007*, volume 346 of *London Math. Soc. Lecture Note Ser.*, pages 275–286. Cambridge Univ. Press, Cambridge, 2007.

[35] S. Oum. Rank-width is less than or equal to branch-width. *J. Graph Theory*, 57(3):239–244, 2008.

[36] S. Oum. Computing rank-width exactly. *Inform. Process. Lett.*, 109(13):745–748, 2009.

[37] S. Oum and P. Seymour. Approximating clique-width and branch-width. *J. Combin. Theory Ser. B*, 96(4):514–528, 2006.

[38] S. Oum and P. Seymour. Testing branch-width. *J. Combin. Theory Ser. B*, 97(3):385–393, 2007.

[39] C. Paul and J. A. Telle. Branchwidth of chordal graphs. *Discrete Appl. Math.*, 157(12):2718–2725, 2009.

[40] G. Reinelt. TSPLIB – a traveling salesman library. *ORSA Journal on Computing*, 3:376–384, 1991.

[41] N. Robertson and P. Seymour. Graph minors. X. Obstructions to tree-decomposition. *J. Combin. Theory Ser. B*, 52(2):153–190, 1991.

[42] P. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[43] D. M. Thilikos and H. L. Bodlaender. Constructive linear time algorithms for branchwidth. Technical Report UU-CS 2000-38, Universiteit Utrecht, 2000.

[44] D. M. Thilikos, M. J. Serna, and H. L. Bodlaender. Constructive linear time algorithms for small cutwidth and carving-width. In *Algorithms and computation (Taipei, 2000)*, volume 1969 of *Lecture Notes in Comput. Sci.*, pages 192–203. Springer, Berlin, 2000.