Minimizing the Oriented Diameter of a Planar Graph

N. Eggemann* and S.D. Noble†

November 3, 2009

Abstract

We consider the problem of minimizing the diameter of an orientation of a planar graph. A result of Chvátal and Thomassen shows that for general graphs, it is NP-complete to decide whether a graph can be oriented so that its diameter is at most two. In contrast to this, for each constant l, we describe an algorithm that decides if a planar graph G has an orientation with diameter at most l and runs in time $O(c|V|)$, where c depends on l.

Keywords: diameter, graph orientation, graph minors, planar graph.

1 Introduction

Our work is motivated by an application involving the design of urban light rail networks. In such an application a number of stations are to be linked with unidirectional track in order to minimize some function of the travel times between stations and subject to constraints on cost, engineering and planning. In practice these constraints mean that the choice of which stations to link may be forced upon us and the only control we have is over the choice of direction of each piece of track. Since the stations that are linked tend to be those that are close to each other, we make the simplifying assumption that the travel time along each single piece of track or link is the same. Consequently the network can be viewed as an (unweighted) graph in which the vertices represent stations and the edges represent track. Furthermore planning constraints tend to rule out the possibility of tracks crossing so the graph is usually planar. The aim is to orient the resulting graph to minimize the maximum travel time between any two stations.

When the underlying graph is obvious, we use n to denote its numbers of vertices. We use (G, ω) to denote the directed graph obtained by applying the orientation ω to the edges of G. Sometimes we abbreviate this to $\vec{G}$. We let $d(x, y)$ denote the distance from vertex x to vertex y. We use $\text{diam}(\vec{G})$ to denote the diameter of $\vec{G}$.

Chvátal and Thomassen [2] showed that determining whether an arbitrary graph may be oriented so that its diameter is at most two is NP-complete. We establish the contrasting result that for any fixed constant l, there is a polynomial time algorithm that will take a planar graph and determine whether it may be oriented so that its diameter is at most l. The algorithm relies on graph minor theory.

*Supported by the EC Marie Curie programme NET-ACE (MEST-CT-2004-6724) address: Department of Mathematical Sciences, Brunel University, Kingston Lane, Uxbridge, UB8 3PH, UK, email:nicole.eggemann@brunel.ac.uk

†Partially supported by the Heilbronn Institute for Mathematical Research, Bristol. address: Department of Mathematical Sciences, Brunel University, Kingston Lane, Uxbridge, UB8 3PH, UK, email:steven.noble@brunel.ac.uk

2 Results

We assume familiarity with the concept of a tree-decomposition [5]. Bodlaender [1] gave an algorithm for finding tree decompositions of small width.

Theorem 2.1 *For each k there is an algorithm running in time $O\big(2^{\theta(k^3)}n\big)$ that inputs a graph G and determines whether the tree-width of G is at most k, and if so finds a tree-decomposition of G with tree-width at most k.*

It is not difficult to see that we may assume that in a tree-decomposition $(T, \mathcal{W})$, the tree T is rooted, each node is either a leaf or has two children and for each edge uv, there is a set W_l containing both u, v such l is a leaf of T. So we may arbitrarily associate any edge of G with a leaf of T. If v is a leaf of T then let E_v be the edges associated with v, otherwise let E_v be the edges associated with the leaves of T that are descendants of v. For any $v \in T$, let $Y_v = \bigcup\{W_u | u = v \text{ or } u \text{ is a descendant of } v\}$. Let G_v be the subgraph of G with vertex set Y_v and edge-set E_v.

We now describe an algorithm Diameter that for fixed k and l, inputs a graph G and a tree-decomposition with width k and determines whether there is an orientation of G with diameter at most l, running in time $O(cn)$, where c depends on k and l.

Given a directed graph $\overrightarrow{G}$ we let $M'(\overrightarrow{G})$ be the shortest path matrix, that is the matrix whose rows and columns are both indexed by $V(\overrightarrow{G})$ with zeros on the diagonal and otherwise (x, y)-entry equal to $d(x, y)$. For the description of the algorithm we will need to introduce the *truncated distance* $d^*(x, y)$ from x to y given by $d^*(x, y) = \min\{l + 1, d(x, y)\}$ if $d(x, y) < \infty$ and $d^*(x, y) = l + 1$ if $d(x, y) = \infty$. Then the truncated distance matrix $M(\overrightarrow{G})$ is defined by replacing each distance $d(x, y)$ in $M'(\overrightarrow{G})$ by $d^*(x, y)$.

The idea of the algorithm is to work upwards through the tree, at each step computing information about G_v that depends on all the orientations of G_v. The key point is that we do not need to consider each such orientation because we categorize the orientations according to what we call their characteristic. We then need to work through the set of characteristics, the size of which depends only on k and l.

Given a graph G, $X \subset V(G)$ and ω an orientation of $E(G)$, we define the *characteristic* $c(G, X, \omega)$ as follows. Suppose without loss of generality that $X = \{v_1, ..., v_{k+1}\}$. Let $v \in V(G) \backslash X$. First we define the *distance vector of v to X* and the *distance vector of v from X*. Let $\overrightarrow{d}(G, X, \omega, v) = (\overrightarrow{d}_1, ..., \overrightarrow{d}_{k+1})$ and $\overleftarrow{d}(G, X, \omega, v) = (\overleftarrow{d}_1, ..., \overleftarrow{d}_{k+1})$ where for $1 \le j \le k+1$, $\overrightarrow{d}_j = d^*(v, v_j)$ and $\overleftarrow{d}_j = d^*(v_j, v)$. Then $c(G, X, \omega)$ is a 4-tuple $(\overrightarrow{S}, \overleftarrow{S}, M, F)$ defined as follows:

- $\overrightarrow{S}$ and $\overleftarrow{S}$ are subsets of $[l + 1]^{k+1}$ with $\overrightarrow{S} = \{\overrightarrow{d}(G, X, \omega, v) | v \in V(G) \backslash X\}$ and $\overleftarrow{S} = \{\overleftarrow{d}(G, X, \omega, v) | v \in V(G) \backslash X\}$.
- M is the submatrix of the truncated distance matrix of (G, ω) corresponding to the vertices of X.
- $F \subseteq \overrightarrow{S} \times \overleftarrow{S}$ such that $(s, t) \in F$ if and only if there are vertices $v, w \in V(G) \backslash X$ such that the distance vector of v to X is s, the distance vector of w from X is t and $d(v, w) > l$.

So F keeps track of pairs of vertices in $V(G) \backslash X$ that are not yet joined by a short enough path. It turns out not to be necessary to store the identities of these vertices, since the information in F is enough.

Let $\Omega(G)$ be the set of all possible orientations of G. Now define the characteristic of $v \in V(T)$ to be $c(v) = \{c(G_v, W_v, \omega) | \omega \in \Omega(G_v)\}$. Because the members of $c(v)$ are distinct characteristics, each one may correspond to many orientations of G_v. The algorithm works upwards from the leaves of T computing $c(v)$ for each v until finally $c(r)$ is computed. Let $h(l, k) = (l + 1)^{k(k+1)} 2^{[2(l+1)^{k+1} + (l+1)^{2k+2}]}$.

Theorem 2.2 *The algorithm* DIAMETER *correctly determines whether the oriented diameter of a graph G with tree-width k is at most l and runs in time $O(nk^2(l+1)^{2k+2}h(l,k)^2)$.*

The theorem follows from the following claims:

- $c(v)$ can be computed for a leaf node v in time $O(k^3 \cdot 2^{\frac{k^2+k}{2}})$;
- for any non-leaf v with children v_1 and v_2, $c(v)$ can be computed in time $O(k^2(l+1)^{2k+2}h(l,k)^2)$ from $c(v_1)$ and $c(v_2)$;
- from $c(r)$ we can determine in time $O(k(l+1)^{k+1}h(l,k))$ whether the graph can be oriented as required.

Clearly the algorithm could easily be modified to find an orientation of minimum diameter for those graphs of tree-width at most k where there is an orientation of diameter at most l.

Our result for planar graphs follows by exploiting the well-known relationship between the diameter and the tree-width of a planar graph [4]. In [3] a general framework is introduced for parameters for which this technique will work. We require the following two results: the first follows because the diameter of a graph cannot increase if an edge is contracted, i.e. diameter is contraction-bidimensional [3], and the second is from [6].

Lemma 2.3 *There is no planar graph G with a $(2l+1)\times(2l+1)$-grid-minor and having diameter less than l.*

Theorem 2.4 *Every planar graph with no $g \times g$-grid-minor has tree-width at most $6g-5$.*

We now establish our main result.

Theorem 2.5 *For every l, there is an algorithm that inputs a planar graph G and determines whether it may be oriented so that the diameter is at most l, running in time $O\big(nl^2(l+1)^{24l+28}h(l,12l+13)^2\big)$.*

Proof: Let G be a planar graph. Using Bodlaender's algorithm we can determine in time $O\big(2^{\theta(l^3)}n\big)$ if G has tree-width at most $12l+13$. If so then it will also find a corresponding tree-decomposition if one exists and then the algorithm DIAMETER may be used to determine whether G can be oriented so that its diameter is at most l.

On the other hand if G has tree-width at least $12l+14$, then by Theorem 2.4, it has a $(2l+3)\times(2l+3)$-grid-minor and therefore diameter at least $l+1$, so clearly there is no orientation with diameter at most l. $\square$ A consequence of our result is that determining whether a planar graph can be oriented so that its diameter is at most l is fixed parameter tractable with respect to l.

3 Conclusion

It would be interesting to try to find a more efficient algorithm, not depending on graph minor theory, and also to determine the complexity when l is part of the input. Furthermore, the proof of Theorem 2.2 could be shortened if the problem could be formulated in monadic second order logic. We have not really pursued this but finding such a formulation seems far from simple due to the need to quantify over all orientations of the edges.

References

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

[2] Chvátal, V. and C. Thomassen, *Distances in orientations of graphs*, J. Combin. Theory Ser. B **24** (1978), pp. 61–75.

[3] Demaine, E. D., F. V. Fomin, M. Hajiaghayi and D. M. Thilikos, *Bidimensional parameters and local treewidth*, SIAM J. Discrete Math. **18** (2004), pp. 501–511.

[4] Eppstein, D., *Diameter and treewidth in minor-closed graph families*, Algorithmica **27** (2000), pp. 275–291.

[5] Robertson, N. and P. D. Seymour, *Graph minors iii: Planar tree-width*, J. Combin. Theory Ser. B **36** (1984), pp. 49–64.

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