One Sided Crossing Minimization Is NP-Hard for Sparse Graphs*

Xavier Muñoz[1]**, W. Unger[2], and Imrich Vrťo[3],***

[1] Departament de Matemàtica IV, Universitat Politècnica de Catalunya
Jordi Girona 1-3, 08034 Barcelona, Spain
[2] Lehrstuhl für Informatik I, RWTH Aachen
Ahornstrasse 55, 52074 Aachen, Germany
[3] Department of Informatics, Institute of Mathematics, Slovak Academy of Sciences,
Dúbravská 9, 842 35 Bratislava, Slovak Republic

Abstract. The one sided crossing minimization problem consists of placing the vertices of one part of a bipartite graph on prescribed positions on a straight line and finding the positions of the vertices of the second part on a parallel line and drawing the edges as straight lines such that the number of pairwise edge crossings is minimized. This problem represents the basic building block used for drawing hierarchical graphs aesthetically or producing row-based VLSI layouts. Eades and Wormald [3] showed that the problem is NP-hard for dense graphs. Typical graphs of practical interest are usually very sparse. We prove that the problem remains NP-hard even for forests of 4-stars.

1 Introduction

The one sided crossing minimization problem consists of placing of the vertices of one part of a bipartite graph on prescribed positions on a straight line and finding the positions of the vertices of the second part on a parallel line and drawing the edges as straight lines such that the number of pairwise edge crossings is minimized. This task represents the basic building block used for drawing hierarchical graphs [2,9] or producing row-based VLSI layouts [11,13]. Hierarchical graphs are abstractions of the various information schemes, flowcharts, PERT diagrams and relationship-structures from economic and social science. It is known that the aesthetics and readability of such diagrams heavily depends on the number of line crossings. The main aim is to produce such drawings automatically which led to the design of tens of various heuristics and minimization procedures. We only point on a very exhaustive recent comparative survey of the best heuristics [1].

* This work has been partially done with the aid of GRAAL network (Graphs and Algorithms)

** Supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) under project TIC 2000 1017.

*** This research was partially supported by the DFG grant No. Hr14/5-1 and the VEGA grant No. 02/7007/20.

P. Mutzel, M. Jünger, and S. Leipert (Eds.): GD 2001, LNCS 2265, pp. 115–123, 2002.

From theoretical point of view the problem is NP-hard [3]. However the graphs produced by the NP-hardness proof are very dense. More precisely, the graphs contain $n_1 n_2/3$ edges where n_1 and n_2 are the numbers of vertices on the first and the second line, respectively. The median heuristic of Eades and Wormald [3] approximates the crossing number in polynomial time within a factor 3. If the maximum degree of the vertices on the free side is 2,3 or 4, then there exists a 2-approximation algorithms [16]. DiBattista et al. [2] proved that any algorithm gives a good approximation for dense graphs but the problem remains the class of sparse graphs. Note that the typical instances of the problem in practice are very sparse graphs [1]. The companion problem - the two sided crossing minimization, when there are no prescribed positions of vertices on either sides is usually solved by iterating the one sided crossing minimization problem. The problem is NP-hard too [5], approximable for regular graphs by a polylogarithmic factor and solvable in polynomial time for trees [12]. This give rise to a natural question, what is the complexity of the one sided crossing minimization problem for sparse graphs or trees. In this paper we prove that the problem remains NP-hard even for forests of 4-stars. The result is in some sense expectable as the related problem, the one sided maximal planar subgraph problem, was proved to be NP–hard for forests of 2-stars [4].

The paper is organized as follows. In Section 2 we give basic notions. In Section 3 we solve the problem for graphs having degrees at most 2 on the free side. Our main result is presented in Section 4 and Section 5 contains some final remarks.

2 Notations

Given a digraph $G = (V, A)$, and given two its spanning subgraphs $H_1 = (V, A_1)$, $H_2 = (V, A_2$, the direct sum $H_1 \cup H_2$ is defined to be the digraph $(V, A_1 \cup A_2)$. Given a digraph G the (multi)digraph nG is obtained from G by replacing each arc by n parallel arcs.

Given a directed graph G, the Feedback Arc Set problem (FAS) consists of finding the minimum number of arcs of A whose removal makes G acyclic. The decision version of the FAS problem is NP–complete [6]. Let $fas(G)$ denote the size of the minimum fedback arc set for G.

A *k–star* S_k is a tree with $(k + 1)$ vertices and k leaves. The *center* of the star is the only vertex of S_k which is not a leaf.

A *directed star* is a digraph whose underlying graph is a star and in which all arcs are pointing towards the center or all arcs are pointing away from the center. We will call them *centripetal stars* and *centrifugal stars* respectively.

We will refer to the One Sided Crossing Minimization problem for graphs of maximal degree k on the free side as the OSCM-k problem.

We will focus on bipartite graphs being forests of stars S_k in which leaves are on prescribed positions on the bottom line and centers must be placed on the top line in such a way that the number of crossings is minimum.

The vertices on the bottom (resp. top) line will be denoted by $v_1, v_2, ..., v_m$ (resp. $w_1, w_2, ..., w_n$). The order of $v_1, v_2, ..., v_m$ will be defined by the indexes.

For any two top vertices w_i, w_j, $cr(i, j)$ stands for the number of crossings produced by edges incident to w_i and w_j when w_i is placed to the right of w_j. In the example of Figure 1 we get $cr(2, 3) = 6$.

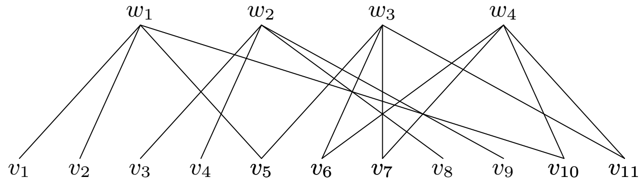

Fig. 1. Example of an input for the OSCM-4

3 Complexity of OSCM-2

In this section we prove that a barycenter algorithm solves the OSCM-2 problem in linear time. Besides its independent interest, the result will be used in proving our main result.

First we give a lower bound on the number of crossings for the OSCM-2 problem. Let G be the input graph. Let w_i (resp. w_j) top vertices with neigbours v_a, v_b with $a < b$ (resp. v_c, v_d with $c < d$). Without loss of generality we assume $a \leqslant c$. The following table give the lower bound on the number of crossings $cr(i, j)$ and $cr(j, i)$.

Situation	$min(cr(i,j), cr(j,i))$	Example in Figure2
$b \leqslant c$	0	$i = 6, j = 8$
$c < b \leqslant d$	1	$i = 1, j = 2$
$d < b \wedge a = c$	1	$i = 1, j = 3$
$d < b \wedge a < c$	2	$i = 6, j = 2$

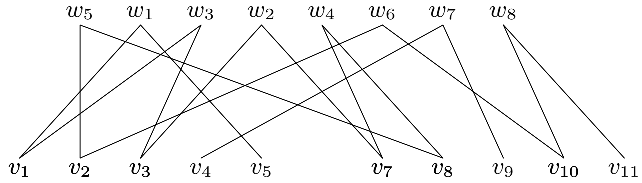

Fig. 2. Example of an input for the OSCM-2

These lower bounds are achieved by placing a node w_i with neighbours v_a and v_b on a unique place on the top line in the position $(a+b)/2 + i/n$ (see for example Figure 3). From this discussion the following corollary is an easy exercise.

Corollary 1. *The OSCM-2 problem can be solved in linear time.*

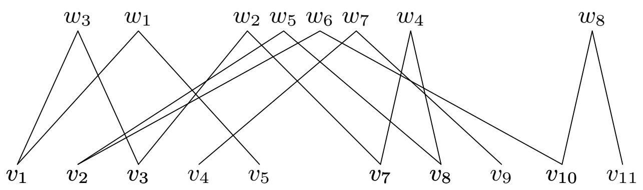

Fig. 3. An optimal solution for the above example

4 Complexity of OSCM-4

In this section we prove that the decision version of the OSCM-4 problem is NP-complete even for forests of 4-stars by a reduction from the decision version of the FAS problem, which immediately implies the NP-hardness of the OSCM-4 problem.

4.1 Restrictions Graph

Definition 1. *Given an instance for the OSCM-k problem with* n k*–stars labeled* $1,\ldots,n$*, the* restrictions graph *is defined as the (multi)digraph* $R_P = (V, A)$ *with vertex set* $V = \{1,\ldots,n\}$ *corresponding to the* n *stars of the instance and in which there are* $cr(i,j) - cr(j,i)$ *arcs from* i *to* j*,* $i,j \in V$ *iff* $cr(i,j) > cr(j,i)$*.*

Proposition 1. *Solving the OSCM-k problem for an instance* P *is equivalent to solving the FAS problem for its restrictions graph* R_P*.*

Proof. In any ordering of the centers of the stars there will always be a certain number of unavoidable crossings. Moreover, the number of unavoidable crossings will be

$$\sum_{1 \leqslant i < j \leqslant n} \min\{cr(i,j), cr(j,i)\}.$$

Notice that if there exists a labeling for the vertices of R_P with numbers $1,\ldots,n$ such that all arcs (i,j) satisfy that $i < j$ (i.e. R_P is a partial order), then the placing of the centers of the stars in the order given by such a labeling will make the number of crossings to be minimum.

If such a labeling is not possible, then minimizing the number of crossings is equivalent to find a labeling of the vertices of R_P with a minimal number of arcs (i, j) with $i > j$. This is also the same as removing the minimum number of arcs of R_P such that R_P becomes a acyclic.

Since a digraph is compatible with a total order if and only if it is acyclic [14], removing the minimum number of arcs such that R_P becomes acyclic is equivalent to minimizing the number of crossings. Finally observe that if G_P is the graph corresponding to the instance P

$$cr(G_P) = \sum_{1 \leqslant i < j \leqslant n} \min\{cr(i, j), cr(j, i)\} + fas(R_P).$$

□

Let us recall that proving the complexity for OSCM-2 could have be done easily by proving that the restrictions graph is always a partial order (i.e. is always acyclic). We prefered to show a constructive method to make further sections clearer.

Remark 1. Note that the connection between the FAS and the one sided crossing minimization was first observed by Sugiyama et al. [15] by their penalty minimization heuristic. It was also essentially used by Eades and Wormald [3] in their NP-hardness proof for dense graphs. Demetrescu and Finocchi [1] utilized the connection in designing a practical heuristic for the problem.

In the rest of the paper we will show that given any simple digraph G there exists a polynomial time algorithm for finding an instance of the OSCM-4 problem with restrictions graph R_P such that FAS for R_P is equivalent to FAS for G.

4.2 Descent Digraphs

Definition 2. *Given a permutation* π *on the set* $1, 2, \ldots, n$, *the* descent digraph $D_\pi(n)$ *is defined as the directed graph with vertices being integers from 1 to* n, *and in which there is an arc from vertex* i *to vertex* j *whenever* $i > j$ *and* $\pi(i) < \pi(j)$.

The name of descent digraphs comes from the concepts of descents in the context of permutations (see [14] for definition and related results on descents).

Recall the following Lemma:

Lemma 1. *Any forest of directed stars is a descent digraph.*

Proof. It is easy to see that the digraph $D_\pi(n)$ with π being the cyclic permutation $(12 \ldots n)$ is a centrifugal star with center n and that $D_\rho(n)$ with ρ being the cyclic permutation $(n(n-1) \ldots 21)$ is a centripetal star with center 1.

Any forest of directed stars can be represented by a product of disjoint cycles of consecutive numbers, and hence it is a descent digraph. □

Notice that given a forest of stars the permutation for the descent digraph is not unique.

4.3 k–Recognizability

Definition 3. *A digraph G is said to be k–recognizable if there is an instance for the OSCM-k problem with restrictions graph being isomorphic to G that can be found in polynomial time.*

Proposition 2. *Let D_π be a descent digraph. Then $2D_\pi$ is 2-recognizable.*

Proof. First of all let us recall that constructing an instance for the OSCM-2 problem with n stars S_2 consists of determining the placement for the $2n$ leaves on a line.

Let us label the two leaves of each S_2 with labels $1, 2$ and let us name stars with numbers $1, \ldots, n$. That is, each of the $2n$ leaves will be denoted by a pair (i, j), $i \in \{1, 2\}$, $j \in \{1, \ldots, n\}$.

If π is a permutation on n symbols, then placing the leaves in the order

$$v_{1,\pi^{-1}(n)} \quad v_{1,\pi^{-1}(n-1)} \quad \cdots \quad v_{1,\pi^{-1}(1)} \qquad v_{2,1} \quad v_{2,2} \quad \cdots \quad v_{2,n}$$

produces an instance for the OSCM-2 problem with restrictions graph being $2D_\pi(n)$. In fact, suppose that there is an arc from the vertex i to the vertex j in the descent digraph D_π. It means that $i > j$ and $\pi(i) < \pi(j)$. The inequality $\pi^{-1}(\pi(i)) > \pi^{-1}(\pi(j))$ implies that the left (right) leaf of the star j is placed left from the left (right) leaf of the star i. It means that $c(i, j) = 3$ and $c(j, i) = 1$ and there are 2 arcs from i to j. (See Figure 4.) □

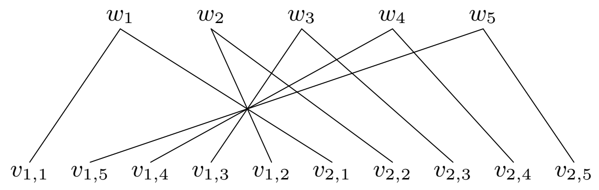

Fig. 4. An instance for the OSCM–2 with the restrictions graph being a centripetal star with 2 parallel arcs isomorphic to $2D_\pi$ with $\pi = (54321)$.

The following lemma is evident

Lemma 2. *Let $\overline{G}$ be a digraph obtained from a digraph G by reversing all arc orientations in G. Then G is k-recognizable iff $\bar{G}$ is k-recognizable.*

Proposition 3. *Let H_0 and H_1 be spanning subgraphs of a digraph G. If H_0 and H_1 are 2–recognizable, then $H_0 \cup H_1$ is 4–recognizable.*

Proof. Place the leaves of the stars corresponding to H_0 and append at the rightmost part the leaves of the stars corresponding to H_1. Identify centers of stars corresponding to the same vertices in H_0 and H_1. The reader can trivially notice that the result is an instance for the OSCM–4 with restrictions graph being $\overline{(H_0 \cup H_1)}$, which by Lemma 2 implies that $H_0 \cup H_1$ is 4–recognizable. Notice that the instance can be generated in linear time. □

4.4 Transformation from FAS

Definition 4. *Given a digraph $G = (V, A)$, the stretch of G, $G' = (V', A')$, is defined as the digraph with vertex set $V' = V \cup A$ and arc set $A' = \{(x, (x, y)) \mid (x, y) \in A\} \cup \{((x, y), y) \mid (x, y) \in A\}$.*

In other words, the stretch of G is defined as the digraph obtained from G by replacing arcs in G by dipaths of length 2.

Proposition 4. *FAS for G is equivalent to FAS for its stretch G'.*

Proof. Let $G = (V, A)$ be a graph and $G' = (V', A')$ be it's stretch.

Assume that $G = (V, F)$ with $F \subset A$ is acyclic, i.e G becomes acyclic by deleting $|A \setminus F|$ arcs. Then $H' = (V', A' \setminus \{((x, y), (x, y) \mid (x, y) \in A \setminus F\})$ is acyclic.

Assume now that $H'' = (V', F')$ with $F' \subset A'$ is acyclic. Then $G' = (V, A \setminus \{(x, y) \mid (x, (x, y)) \in F' \subset A' \vee ((x, y), y) \in F' \subset A'\})$ is acyclic.

In both cases the number of deleted arc stays the same, proving that FAS for G is equivalent to FAS for its stretch G'. □

Proposition 5. *Given any digraph G its stretch G' is the direct sum of two descent digraphs and both digraphs can be identified in polynomial time.*

Proof. Let us consider each vertex $v \in V)$ and assign label 1 to the arcs pointing away from it and label 0 to the arcs pointing towards to it. Clearly after completing the labeling for all vertices in V all arcs in A' will have a label (and only one). The subgraph induced by arcs of label 1 is a spanning subgraph of G' consisting of a forest of directed stars and so is the subgraph induced by arcs with label 0. Since forests of stars are descent digraphs, the result is proved. Notice that the digraphs can be identified in linear time. □

Finally consider the following trivial result:

Lemma 3. *FAS for a digraph G is equivalent to FAS for nG.*

Putting together all these results, we have

Theorem 1. *The decision version of the OSCM-4 problem is NP–complete.*

Proof. Given a generic digraph G, construct the stretch G' of G, and descent digraphs H_0, H_1 such that G'= $H_0 \cup H_1$, (Lemma 1 and Proposition 5). By Proposition 2, $2H_0$ and $2H_1$ are both 2-recognizable, and by Proposition 3, $2G'$ is 4-recognizable. Determine an instance I of the OSCM-4 problem with restriction graph isomorphic to $2G'$. By Proposition 1 and Lemma 3, solving I is equivalent to FAS for G' and for G. Finally, let $G_I = (V_0, V_1, E), V_1 = \{1, 2, 3, ..., n\}$ be the graph corresponding to the instance I then

$$cr(G_I) = \sum_{1 \leqslant i < j \leqslant n} \min\{cr(i,j), cr(j,i)\} + 2fas(G).$$

□

5 Conclusions and Further Work

We proved that the one sided crossing minimization problem is NP-hard even for forests of stars of degree 4. So far it was known that the problem is NP-hard for dense graphs. Our result motivates for looking for good approximation algorithms for sparse graphs, which are typical practical instances. As we shown in Section 3, the OSCM-2 problem can be solved in linear time using the barycenter algorithm. It remains to resolve the problem OSCM-3. We conjecture that it is NP-hard.

References

1. Demetrescu, C., Finocchi, I.: Removing Cycles for Minimizing Crossings. J. Experimental Algorithmics. To appear
2. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for Visualization of Graphs. Prentice Hall (1999)
3. Eades, P. Wormald, N.C.: Edge Crossings in Drawing Bipartite Graphs. Algorithmica **11** (1994) 379-403
4. Eades, P., Whitesides, S.: Drawing Graphs in Two Layers. Theoretical Computer Science **131** (1994) 361-374
5. Garey, M.R., and Johnson, D.S.: Crossing Number is NP-Complete. SIAM J. Algebraic and Discrete Methods **4** (1983) 312–316
6. Gavril, F.: Some NP-Complete Problems on Graphs. In: 11th Conf. on Information Sciences and Systems. John Hopkins University Press, Baltimore (1977) 91-95
7. Jünger, M., Mutzel, P.: 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms. J. Graph Algorithms and Algorithms **1** (1999) 11-25
8. Matuszewski, C., Schönfeld, R., Molitor, P.: Using Sifting for k-Layer Crossing Minimization. In: 7th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science, Vol. 1731. Springer-Verlag, Berlin (1999) 217-224
9. Mutzel, P.: Optimization in Leveled Graphs. In: Pardalos, M., Floudas, C.A. (eds.): Encyclopedia of Optimization. Kluwer, Dordrecht (2001)
10. Purchase, H.: Which Aesthetic Has the Greatest Effect on Human Understanding? In: 5th Intl. Symposium on Graph Drawing. Lecture Notes in Computer Science, Vol. 1353. Springer-Verlag, Berlin (1997) 248-261

11. Sechen, C.: VLSI Placement and Global Routing Using Simulated Annealing, Kluwer, Dordrecht (1988)
12. Shahrokhi, F., Sýkora, O., Székely, L.A., Vrt'o, I.: On Bipartite Drawings and the Linear Arrangement Problem. SIAM J. Computing **30** (2000) 1773-1789
13. Stallmann, M., Brglez, F., Ghosh, D.: Heuristics, Experimental Subjects, and Treatment Evaluation in Bigraph Crossing Minimization. J. Experimental Algorithmics. To appear
14. Stanley, R. H.: Enumerative Combinatorics. Wadsworth & Brooks, Monterey (1986)
15. Sugiyama, K., Tagawa, S., Toda, M.: Methods for Visual Understanding of Hierarchical System Structures. IEEE Transactions on Systems, Man, and Cybernetics **SMC–11** (1981) 109-125
16. Yamaguchi, A., Sugimoto, A.: An Approximation Algorithm for the Two-Layered Graph Drawing Problem. In: 5th Annual Intl. Conf. on Computing and Combinatorics. Lecture Notes in Computer Science, Vol. 1627. Springer-Verlag, Berlin (1999) 81-91