



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Positioning a New Vertex that Minimize the Number of New Crossings

Hamid Sanatnama, Afshin Amini, Alireza Bitaraf Haghighi and Farshad Brahimi
Faculty of Mathematics and Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

Abstract: Routing, scheduling and location problems in our real-life can be formulated as combinatorial optimization problems whose goal is to find a linear layout of an input graph in such a way that the number of edge crossings is minimized. The problem of adding a new vertex to an existing straight-line graph drawing and connecting it to its adjacent vertices in such a way that the number of crossings made by adding this vertex is minimized is known as crossing minimization problem. Although many crossing number related problems are NP-Complete or NP-Hard. In this study, we present a polynomial time algorithm; $O(|V|^5|E|)$. It checks a point in every Fully Drawing Region (FDR) and returns a point which is the position of the new vertex that minimizes the number of added crossings. It may be useful when we have an existing drawing which cannot be changed and we don't know how vertices will be added in future.

Key words: Graph drawing, graph algorithm, computational complexity, crossing minimization, straight-line drawing

INTRODUCTION

Graphs exhibit hierarchies with edges between the vertices to visualize the architectural structure. Visualization of architectural structure will increase our perception of complexity. The visualization algorithms can either be applied in 2D or 3D, a nice overview being (Sanatnama and Brahimi, 2010). Embedding a graph in the plane with the minimum number of edge crossings is known as crossing minimization problem. Recently systems engineers have shown interest in algorithms for drawing directed graphs so that they are easy to understand and remember. Each of the commonly used methods has a step which aims to adjust the drawing to decrease the number of arc crossings. Hierarchical graphs are abstractions of the various information schemes, flowcharts, PERT diagrams and relationship-structures from economic and social science. Hierarchical drawing of a graph is also the most popular drawing convention for directed graphs. Radial drawing is an adaption of Sugiyama framework for visualizing hierarchical information where vertices are placed on concentric circles and edges are drawn as poly line segments of spirals which are monotone from the concentric center to the outside (Bachmaier, 2007). Improved algorithms by Even *et al.* (2003) can be applied to improve a variety of VLSI layout problems such as minimizing the number of crossings in a drawing on the plane of a bounded degree graph and minimizing the VLSI layout area of a graph of maximum degree four.

A straight-line drawing of graph G is a drawing where edges are straight-line segments and rectilinear crossing number of graph G is the minimum number of crossings in any straight-line drawing of G . Unfortunately, the algorithms for computing the crossing number which are NP-complete or NP-Hard, has been shown by Bachmaier *et al.* (2010) and Muoz *et al.* (2002). This means that the crossing minimization problem of is probably to be intractable. Pelsmajer *et al.* (2009) showed that computing the crossing number of a graph with a given rotation system is NP-complete and also they investigated the special case of multi-graphs with rotation systems on a fixed number k of vertices. Fernau *et al.* (2005) considered two-layer crossing minimization problems for the purpose of comparing two unordered trees which have applications in various areas like text data bases, bioinformatics, compiler construction etc. Buchin *et al.* (2008) worked on drawing so-called tanglegrams, that is, pairs of trees whose leaf sets are in one-to-one connection and this is essential that both trees are drawn without edge crossings and that the inter-tree edges have as few crossings as possible. Biedl *et al.* (2009) proposed another modification, where edges are simple arcs and allow self-crossings of arcs representing the edges but all self-crossings should be transversal. Mohar (2009) presented a method to compute genus distributions of open chains in which the degree of the amalgamated vertex can be arbitrarily large. Hong and Nagamochi (2009) presented the first constant-factor approximation

algorithm for the one-sided crossing minimization problem in a 2-layered radial drawing. Pach and Toth (2009) depicted that if G is a graph with n vertices and $e = 4n$ edges, drawn in the plane in such a way that if two or more edges (arcs) share an interior point. Then it is shown that the number of crossing points, counted without multiplicity, is at least constant times e and that the order of magnitude of this bound cannot be improved. Alpert *et al.* (2009) presented the generalized star drawings of the d -regular (if all the vertices have degree d , the graph is called d -regular) graphs $S_{n,d}$ of order n where, $n + d = 1 \pmod{2}$ and prove that they maximize the maximum rectilinear crossing numbers.

Our research question can be formulated as follow: How can we add a vertex v to its incident edges in an existing straight-line drawing of graph G where the number of new crossings made by adding this new vertex is minimized as shown in Fig. 1?

Definitions and notation: The terms we use for graph theory terminology have been defined in (West, 2001). For a graph G we use the following notations:

- $V(G)$: The set of vertices of G
- $|V|$: The number of vertices of G
- $|E|$: The number of edges of G

Note 1: All concepts and figures in this paper are in 2D coordinate system and for ambiguity we use definition of a drawing of graph G as presented by Singer (1971):

A drawing of graph G is a map $D: G \rightarrow \mathbb{R}^2$ such that no point in is \mathbb{R}^2 image of more than two points of G which must not belong to the same edge of G and neither of which may be a vertex.

We also use definition of crossing as depicted by Singer (1971):

A point of \mathbb{R}^2 which is image of two points of G is called a crossing.

Definition 1: Full Drawing of a straight-line drawing D of graph G denoted as FD is the union of all (infinite) lines passing through images of every pair of distinct vertices of G in D :

$$FD = \bigcup_{L_{i,j}} \quad i,j \in \{1,2,\dots, |V|\}, i < j \quad (1)$$

where, L_{ij} is the line passing through images of vertices v_i and v_j of graph G in drawing D as exposed in Fig. 2.

We clarify the meaning of a region of full drawing and name it full drawing region.

Definition 2: A Full Drawing Region of a full drawing FD denoted as FDR is a maximal set of points $P \in \mathbb{R}^2$ such that

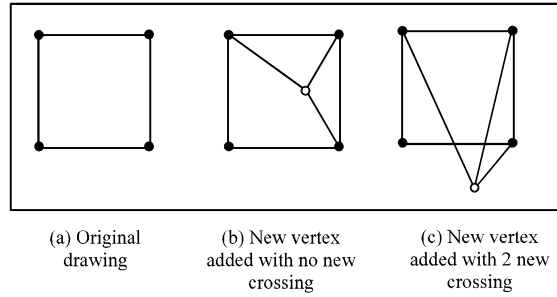


Fig. 1: Adding a new vertex to existing straight-line drawing

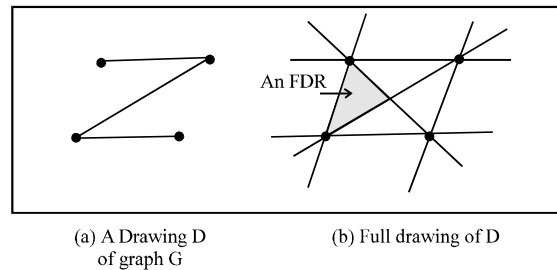


Fig. 2: Full Drawing and FDR

for each pair of points $P_1, P_2 \in \mathbb{R}^2$ there exists a curve C which connects P_1 to P_2 and $C \cap FD = \emptyset$ as depicted in Fig. 2.

Note 2: An FDR may be unbounded.

Definition 3: The Point crossing number of point p and vertex v of graph G in an existing straight-line drawing D of G denoted as $PCR(p, v)$ is the number of crossings between existing drawing edges and the edge made by putting a new vertex on p and connecting it to v .

PROBLEM DEFINITION

Suppose we have an existing straight-line drawing D of graph G . We want to add a new vertex v_{new} to G and connect it to a set $C \subseteq V(G)$. The goal is to find a position for v_{new} such that the number of new crossings made by adding this new vertex is minimized. We can state the problem formally as:

$$\text{Best point} = \{P \in \mathbb{R}^2 \mid \sum_{v \in C} PCR(p, v) \text{ is minimum} \} \quad (2)$$

THE PROPOSED ALGORITHM

The main idea of our algorithm is that adding the new vertex in any point of an FDR produces the same number of crossings. Therefore, we have to check only one point inside each FDR.

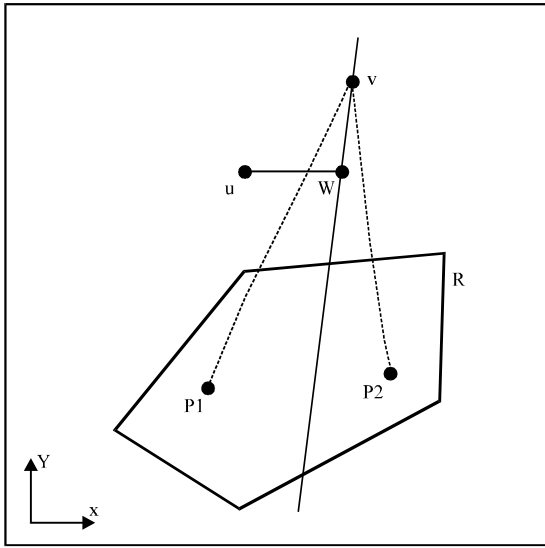


Fig. 3: Proof of theorem 1

Theorem 1: Let R be an FDR. For every pair of points $P_1, P_2 \in \mathbb{R}$ and every vertex $v \in V(G)$:

$$PCR(P_1, v) = PCR(P_2, v) \tag{3}$$

Proof by contradiction: Suppose there exist two points P_1 and P_2 in the same FDR R which when connected to a vertex v create different number of crossings. Then there should be at least one edge $e(u, w)$ which crosses exactly one of the line segments (P_1, v) or (P_2, v) but doesn't cross the other. So either the line is passing through v and u or the line is passing through v and w partition R into two FDRs which contradict R being an FDR (Fig. 3).

Corollary 1: Let R be an FDR and C be a subset of $V(G)$. For every pair of points $P_1, P_2 \in R$:

$$\sum_{v \in C} PCR(P_1, v) = \sum_{v \in C} PCR(P_2, v) \tag{4}$$

Proof. Trivial result of theorem 1: Our algorithm checks a point in every FDR and returns a point which minimizes the number of added crossings.

MAIN ALGORITHM

Input: A drawing D of graph G and a set $C \in V(G)$ that are adjacent to the new vertex

- Step 1:** Calculate SFDR the set of all FDRs
- Step 2:** For each FDR R in S_{FDR} calculate and save a point P_R inside of R
- Step 3:** For each FDR R in S_{FDR} calculate and save

$$Cost(R) = \sum_{v \in C} PCR(P_R, v) \tag{5}$$

Step 4: Define R_{min} as an FDR in S_{FDR} with Minimum of Cost (R)

Any point in R_{min} which doesn't make 3 edges cross at the same point is acceptable.

COMPUTATIONAL COMPLEXITY

Lemma 1: Lines passing through every pair of distinct vertices of G create $O(|V|^4)$ FDRs.

Proof: The maximum number of the regions that we can have by partitioning the plane with n distinct lines is $\frac{n(n+1)+2}{2} \in O(n^2)$ as mentioned by Manber (1989). The

number of lines in full drawing of drawing D of graph G with $|V|$ vertices is equal to the number of edges in a complete graph with $|V|$ vertices i.e. $O(|V|^2)$. So if we partition the plane by those lines, we get $O((|V|^2)^2)$ regions i.e., $O(|V|^4)$ FDRs.

To implement step 3 of our algorithm we have checked crossings between every edge incident to new vertex with every edge in existing graph drawing which is $O(|V| \cdot |E|)$ for a single point. Because we check a point in every FDR, $O(|V|^4)$ points are checked so the total complexity of step 3 of our algorithm is $O(|V|^5 \cdot |E|)$. Because other steps of algorithm have a lower time complexity the overall time complexity of our algorithm is $O(|V|^5 \cdot |E|)$.

APPLICATION OF THE ALGORITHM

Besides that our algorithm can be used as visualizing algorithms which have been mentioned by Sanatnama and Brahimi (2010) it can be also used in network connectivity. One of the problems in connectivity in a Wireless LAN Network is to find a good location for installing an Access Point (AP) or a wireless device in a building in order to increase the probability of receiving the signal in a multipath environment.

Indoor wave propagation is affected by the building walls. Walls can affect in sent wavelengths between a computer and AP. The point is to put an AP in a location where the walls affect less on wavelengths. A building with 5 rooms can be presented by a graph as reveals in Fig. 4a. In this graph walls are edges of the graph and vertices of the graph are the points where the walls are connected together. We can assume there are some computers in each room as depicted in Fig. 4b. It is also assumed that computers are vertices in this graph which are not connected to each other.

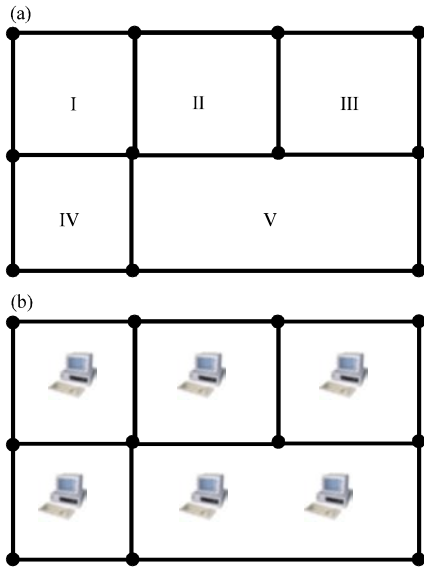


Fig. 4: A building presented by a graph, (a) a building with 5 rooms and (b) Computer (s) in each room

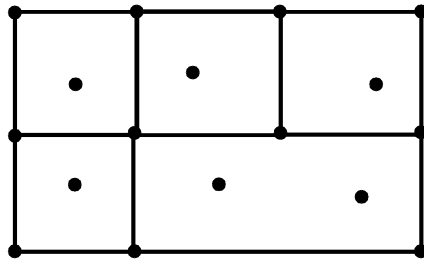


Fig. 5: The graph as parameter to our algorithm

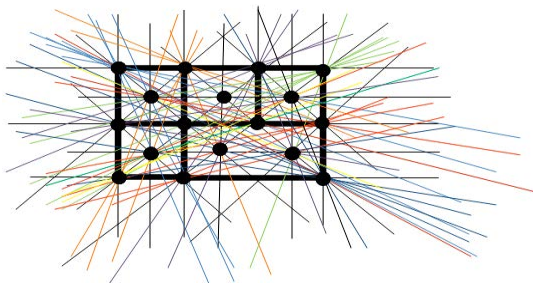
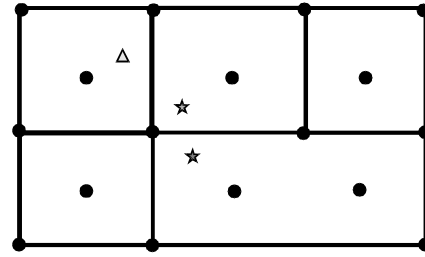


Fig. 6: All possible crossings

Then the graph which we want to apply our algorithm on can be presented as in Fig. 5. This is the graph as an entry parameter to our algorithm. A new vertex should be connected to vertices which present computers in the graph. The objective is to find a best place for the new vertex which has minimum crossing through the walls.

The algorithm will check a point in every Fully Drawing Region (FDR) and returns a point which is the



$$\begin{aligned} \text{Cost } (\star) &= 1+0+1+2+1+1 = 6 \\ \text{Cost } (\star) &= 2+1+2+1+0+0 = 6 \\ \text{Cost } (\Delta) &= 0+1+2+1+2+2 = 8 \end{aligned}$$

Fig. 7: Points in the regions which have minimum crossings

position of the new vertex that minimizes the number of added crossings. All possible crossings in each region are shown in Fig. 6.

After applying the algorithm on the graph it will find the FDR and return a point which has minimum crossing compare to the other points in other regions. It is also possible that among regions there are many regions that have minimized points which our algorithm will find all of them as shown in Fig. 7.

CONCLUSION AND FUTURE WORKS

We have given a polynomial algorithm for a crossing number related problem while many crossing number related problems are NP-Complete or NP-Hard. Although being polynomial time, the algorithm is only practical for graphs with small number of vertices, so faster algorithms are needed. This algorithm can be extended as a greedy algorithm for general problem of drawing graphs with smaller number of crossings, by adding vertices one by one and applying this algorithm. Analyzing the performance and outcome of this greedy approach is our consideration for future works.

Although we have mentioned general application of this algorithm, field specific applications of this algorithm is a consideration for future works.

This research project was conducted from May 2010 to Nov 2010 at Shahid Bahonar University of Kerman Iran.

REFERENCES

Alpert, M., E. Feder and H. Harborth, 2009. The maximum of the maximum rectilinear crossing numbers of d-regular graphs of order n. *Electronic J. Combinatorics*, 16: 1-16.

Bachmaier, C., 2007. A radial adaption of the sugiyama framework for visualizing hierarchical information. *IEEE Trans. Vis. Comput. Graph.*, 13: 583-594.

- Bachmaier, C., H. Buchner, M. Forster and S. Hong, 2010. Crossing minimization in extended level drawings of graphs. *Discrete Applied Mathe.*, 158: 159-179.
- Biedl, T., F.J. Brandenburg and X. Deng, 2009. On the complexity of crossing in permutations. *Discrete Mathe.*, 309: 1813-1823.
- Buchin, K., M. Buchin, J. Byrka, M. Nollenburg, Y. Okamoto, R.I. Silveira and A. Wol, 2008. Drawing (Complete) Binary Tanglegrams: Hardness, Approximation, Fixed-Parameter Tractability. Springer-Verlag, New York, pp: 324-335.
- Even, G., S. Guha and B. Schieber, 2003. Improved approximations of crossings in graph drawings and VLSI layout areas. *SIAM J. Comput.*, 32: 231-252.
- Fernau, H., M. Kaufmann and M. Poths, 2005. Computing trees via crossing minimization. Proceedings of 25th Conference on Foundations of Software Technology and Theoretical Computer Science, LNCS 3821, (FSTTCS'05), Springer-Verlag, Berlin, pp: 457-469.
- Hong, S. and H. Nagamochi, 2009. Approximation algorithms for minimizing edge crossings in radial drawings. *Algorithmica*, 58: 478-497.
- Manber, U., 1989. Introduction to Algorithm: A Creative Approach. Addison-Wesley, Reading.
- Mohar, B., 2009. The genus crossing number. *ARS Mathematica Contemporanea*, 2: 157-162.
- Muoz, X., W. Unger and I. Vrto, 2002. One sided crossing minimization is NP-hard for sparse graphs. *Lecture Notes Comput. Sci.*, 2265: 16-20.
- Pach J. and G. Toth, 2009. Degenerate crossing numbers. *Discrete Comput. Geometry*, 41: 376-384.
- Pelsmajer, M.J., M. Schaefer and D. Stefankovic, 2009. Crossing number of graph with rotation systems. *Algorithmica*. 10.1007/s00453-009-9343-y
- Sanatnama, H. and F. Brahim, 2010. Graph drawing algorithms: Using in software tools. *J. Applied Sci.*, 10: 1894-1901.
- Singer, D., 1971. The rectilinear crossing number of certain graphs. Unpublished Manuscript, 1971. Quoted in Gardner, M. *Knotted Doughnuts and Other Mathematical Entertainments*. W.H. Freeman, New York, 1986.
- West, D.B., 2001. Introduction to Graph Theory. 2nd Edn., Pearson Education, Inc., London.