An Algorithm for 3D-biplanar Graph Drawing

Meysam Tavassoli* Mohammd Ghodsi† Farnoosh Khodakarami ‡ S. Mehdi Hashemi §

Abstract

We introduce the concept of 3D-biplanar drawing in which we partition a graph into two planar induced subgraphs. Our goal is to find such a partition with the minimum number of edges between the two partitions. We prove that this problem is NP-complete and present a randomized parameterized algorithm with $O(n^k)$ time, where k is the ratio of the optimal solution to the min-cut size of the graph.

1 Introduction

Layered graph drawing [1, 14] is a popular paradigm for drawing graphs which has applications in visualization [15], in DNA mapping, and in VLSI layout [9]. In a layered drawing of a graph, vertices are arranged in horizontal layers, and edges are routed as polygonal lines between distinct layers. For acyclic digraphs, it may be required that edges point downward. Figure 1 shows a sample graph with its 3-Layer drawing.

The quality of layered drawings is assessed in terms of criteria to be minimized, such as the number of edge crossings; the number of edges when removed eliminates all crossings; the number of layers; the maximum span of an edge, i.e., the number of layers it crosses; the total span of the edges; and the maximum number of vertices in one layer.

Research on layered graph drawing has been mainly focused on drawing a graph which admits a 2-layer drawing with no edge crossings. There are some well known problems in this area:

Biplanar drawing: Given a bipartite graph $G = (A, B; E)$, G is said to be biplanar if the vertices can be drawn on two layers so that none of the edges of G cross. Eades and Whitesides proved that determining whether a given G has a biplanar subgraph with at least K edges is NP-complete. This remains true when the positions of the vertices on one layer are specified [5].

Planarization: 2-Layer Planarization problem, in which given a graph G (not necessarily bipartite), and an integer k called its parameter, the question is whether G can be made biplanar by deleting at most k edges. If a permutation π of A is given, this problem is called 1-Layer Planarization.

Crossing Minimization: Instead of deleting edges, one can seek to minimize the number of crossings in a 2-layer drawing (here the input graph must be bipartite). The corresponding problems are called 1- and 2-Layer Crossing Minimization.

Unfortunately, the question of whether a graph G can be drawn in two layers with at most k crossings (Crossing Minimization), where k is a part of the input, is NP-complete [6], as is the question of whether r or fewer edges can be removed from G so that the remaining graph has a crossing-free drawing on two layers (Planarization) [5]. Both problems remain NP-complete when the permutation of vertices in one of the layers is given [5, 6].

Two-layer drawings are of fundamental importance to Sugiyama approach to multilayer drawing [14]. There are numerous different algorithms for planarization and crossing minimization problems, such as integer linear programming algorithms [8, 16], heuristic methods [6, 8, 14], approximation algorithms [12], and fixed parameter algorithms [3, 2].

We extend biplanar drawing method on 3D space, and instead of line layers we use the plane layers. Note that a kind of drawing similar to 3D-biplanar drawing, has been purposed before for clustered graphs [4]. We let vertices be placed in two parallel planes, and the edges can connect two vertices in the same layer or in different layers, but in each layer the induced subgraph must be planar as illustrated in Figure 2. We call such drawing *3D-biplanar*, and define 3D-biplanar cut as the number of edges between the two different layers. Our goal is to find such partition with minimum number of edges between these two partitions. In other words, we want to find 3D-biplanar cut with minimum size.

We prove that this problem is NP-complete and present a randomized parameterized algorithm for it with $O(n^k)$ time, where k is the ratio of the optimal solution to the min-cut size of graph.

This paper is organized as follows. After proving the NP-hardness of this problem in Section 2, we present our randomized parameterized algorithm in Section 3. In Section 4 we analyze our algorithm. Fi-

*Department of Computer Engineering, Sharif University of Technology, tavassoli@ce.sharif.edu

†Department of Computer Engineering, Sharif University of Technology, and IPM School of Computer Science (No. CS1382-2-02), ghodsi@sharif.edu

‡Department of Mathematics and computer science, AmirKabir University of Technology, khodakarami@aut.ac.ir

§Department of Mathematics and computer science, AmirKabir University of Technology, hashemi@aut.ac.ir

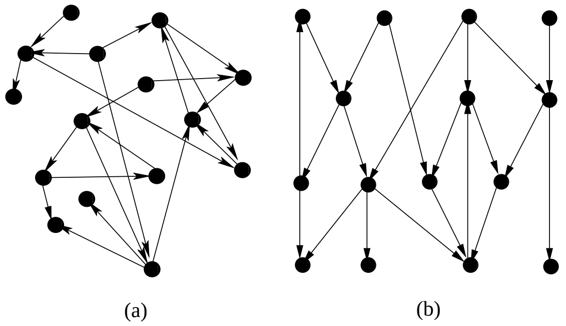

Figure 1: (a)A sample graph and (b)its 3-Layer drawing

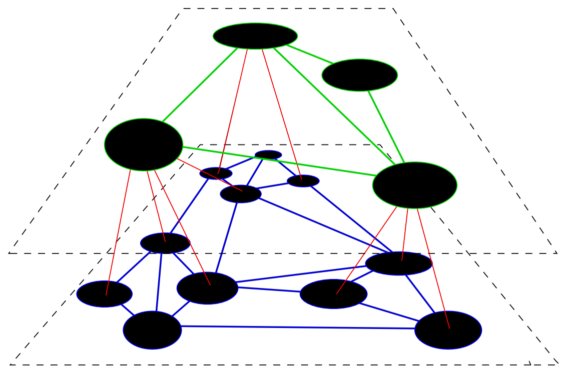

Figure 2: A 3D-biplanar drawing

nally, in Section 5 we draw some conclusions.

2 Hardness of 3D-biplanar Drawing

In this section we prove that finding a 3D-biplanar drawing with minimum 3D-biplanar cut, is NP-complete as many other layered graph drawing problems.

To prove that finding a 3D-biplanar drawing with minimum 3D-biplanar cut is NP-complete, we use a theorem from Lewis and Yannakakis [10] which is based on independent work by the two authors that actually proves a more general result. They use this result to prove that Maximum Induced Planar Subgraph [1] is NP-complete.

Theorem 1 *[10] Supposeπ is a graph property satisfying the following conditions:*

1. *There are infinitely many graphs for which π holds.*
2. *There are infinitely many graphs for which π does not hold.*
3. *If π holds for a graph G and if G' is an induced subgraph of G, then π holds for G'. This is called hereditary property.*

Then, the following problem is NP-complete: Given a graph $G = (V, E)$ and a positive integer $k \leq |V|$, is there a subset $V' \subseteq V$ with $|V'| \geq k$ such that π holds for the subgraph of G induced by V'?

Theorem 2 *Given a graph $G = (V, E)$, the problem of finding a cut C that splits vertex set V into two subsets V_1 and V_2 such that each subset being planar is NP-complete.*

Proof. Consider planarity property for a graph G. Planarity satisfies the following three conditions:

[1]Given a graph $G = (V, E)$ and a positive integer $k \leq |V|$, is there a subset $V' \subseteq V$ with $|V'| \geq k$ such that the subgraph of G induced by V' is planar?

1. It is straightforward that there are infinitely many planar graphs. For example all trees are planar.
2. There are infinitely many graphs that has $K_{3,3}$ or K_5 as a subgraph. Each graph that has $K_{3,3}$ or K_5 as a subgraph is not planar.
3. Each induced subgraph G' of a planar graph G is planar too.

Hence, planarity property satisfies the three conditions discussed in theorem 1. So, finding a planar induced subgraph with vertex set V' such that $|V'| \geq k$ for some $k \leq |V|$ is NP-complete. We know that $|V_1| + |V_2| = |V|$ thus one of $|V_1|$ or $|V_2|$ is greater than or equal to $|V|/2$. If we choose $k = |V|/2$, we are done. □

We can't find a 3D-biplanar drawing for all graphs. So, in the following lemma we will show a necessary condition for graphs that have a 3D-biplanar drawing.

Lemma 3 *If a graph G has a 3D-biplanar drawing, then G can not contain K_9 as a subgraph.*

Proof. Suppose graph G contains K_9 as a subgraph. Hence, when we split vertex set V into two subsets V_1 and V_2, the induced subgraph with one of vertex subsets V_1 or V_2 contains K_r with $r \geq 5$ as a subgraph. We know that planar graphs can not contain K_5 as a subgraph. So graph G can not split into two induced planar subgraphs. □

3 Randomized Parameterized Algorithm

In this section we introduce our main algorithm for finding a minimum 3D-biplanar cut for a graph G if it exists. With lemma 3, we can first skip graphs that doesn't have necessary condition for 3D-biplanar drawing.

We repeat the following step: pick an edge uniformly at random and merge the two vertices at its end-points as illustrated in Figure 3. If as a result there are several edges between some pairs of (newly formed) vertices, retain them all. Edges between vertices that are merged are removed, so that there are never any self-loops. We refer to this process of merging the two end-points of an edge into a single vertex as *contraction* of that edge. With each contraction, the number of vertices of G decreases by one. The crucial observation is that an edge contraction does not reduce the 3D-biplanar cut size of graph G. This is because every cut in the graph at any intermediate stage is a cut in the original graph. The algorithm continues the contraction step until only two vertices remain. At this point, the set of edges between these two vertices is a cut in G and is output as a candidate 3D-biplanar cut.

Now, there are two subsets V_1 and V_2 with a cut-edge C. Consider two induced subgraphs G_1, G_2 with respectively vertex sets V_1 and V_2. We can easily test the planarity of these two induced subgraphs by one of the planarity test algorithms like [7, 11, 13]. These algorithms take linear time in the worst case.

If G_1 or G_2 are not planar, we should repeat this algorithm to find two induced planar subgraphs. This algorithm does not always find a minimum 3D-biplanar cut even we find two induced planar subgraphs. So, we need to repeat this algorithm to achieve the minimum 3D-biplanar cut. In the next section we will compute the number of times needed to repeat this algorithm until to find a minimum 3D-biplanar cut, if it exists.

4 Analysis of the Algorithm

Let k denote the minimum biplanar cut size. We fix our attention to a particular minimum biplanar cut C with k edges. We will bound from below the probability that no edge of C is ever contracted during an execution of the algorithm, so that the edges surviving till the end are exactly the edges in C.

Suppose the min-cut size in graph $G = (V, E)$ is k'. So $k \geq k'$ and $k = hk'$ (h is a positive constant factor that is not dependent on input size, it depends on only the minimum biplanar cut size and the min-cut size). Clearly, G has at least $k'n/2$ edges; otherwise there would be a vertex of degree less than k', and its incident edges would be a min-cut of size less than k'.

Let $\mathcal{E}_i$ denote the event of not picking an edge of C at the ith step, for $1 \leq i \leq n-2$. The probability that the edge randomly chosen in the first step is in C is at most $k/(nk'/2) = 2h/n$, so that $Pr[\mathcal{E}_1] \geq 1 - 2h/n$. Assuming that $\mathcal{E}_1$ occurs, during the second step there are at least $k'(n-1)/2 = k(n-1)/2h$ edges, so the probability of picking an edge in C is at most $2h/(n-1)$, so that $Pr[\mathcal{E}_2|\mathcal{E}_1] \geq 1-2h/(n-1)$. At the ith step, the number of remaining vertices is $n-i+1$. The size of the min-cut is still at least k', so the graph has at least $k'(n-i+1)/2 = k(n-i+1)/2h$ edges remaining at this step. Thus, $Pr[\mathcal{E}_i|\cap_{j=1}^{i-1}\mathcal{E}_j] \geq 1-2h/(n-i+1)$.

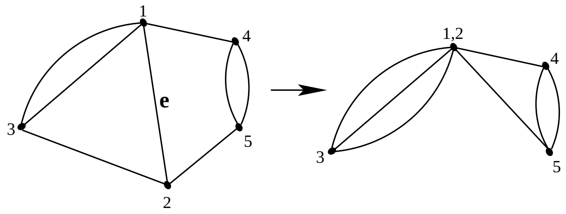

Figure 3: contraction of edge e

We use the basics to compute probability that no edges of C is ever picked in the process:

$$Pr[\cap_{j=1}^{n-2}\mathcal{E}_i] \geq \prod_{i=1}^{n-2}(1 - \frac{2h}{n-i+1}) \geq \frac{2}{n^{2h}}$$

The probability of discovering a particular 3D-biplanar drawing (which may in fact be the unique 3D-biplanar drawing in G) is larger than $2/n^{2h}$. Thus, our algorithm may err in declaring the drawing it outputs to be a optimum 3D-biplanar drawing. Suppose we were to repeat the above algorithm $n^{2h}/2$ times, where h is the ratio of optimal solution to min-cut size of graph, making independent random choices each time. Obviously, the probability that a min-cut is not found in any of the $n^{2h}/2$ attempts is at most

$$\left(1 - \frac{2}{n^{2h}}\right)^{\frac{n^{2h}}{2}} \leq \frac{1}{e}.$$

By this process of repetition, we have managed to reduce the probability of failure from $1 - 2/n^{2h}$ to a more respectable $1/e$. Further, executions of the algorithm will make the failure probability arbitrarily small–the only consideration being that repetitions increase the running time.

5 Conclusion

This paper introduces the concept of 3D-biplanar drawing in which a graph partitioned into two planar induced subgraphs. By straightforward reduction from a more general result, the paper shows that deciding whether a given graph can be cut into two planar graphs is NP-complete. A randomized algorithm for finding an optimal cut is given, whose running time depends on the ratio of the optimal solution and the min-cut size, as a parameter.

A further step will be to design an efficient algorithm or an approximation algorithm with a good approximation factor for finding a minimum 3D-biplanar cut.

References

[1] M. J. Carpano. Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Trans. Syst. Man Cybern.*, SMC-10(11):705–715, 1980.

[2] V. Dujmovic, M. R. Fellows, M. T. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. On the parameterized complexity of layered graph drawing. In *ESA '01: Proceedings of the 9th Annual European Symposium on Algorithms*, pages 488–499, London, UK, 2001. Springer-Verlag.

[3] V. Dujmovic, M. R. Fellows, M. T. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. A fixed-parameter approach to two-layer planarization. In *GD '01: Revised Papers from the 9th International Symposium on Graph Drawing*, pages 1–15, London, UK, 2002. Springer-Verlag.

[4] P. Eades and Q.-W. Feng. Multilevel visualization of clustered graphs. In S. North, editor, *Graph Drawing, Berkeley, California, USA, September 18-20, 1996*, pages pp. 101–112. Springer, 1997.

[5] P. Eades and S. Whitesides. Drawing graphs in two layers. *Theoretical Computer Science 131*, pages 361–374, 1994.

[6] P. Eades and N. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 10:379–403, 1994.

[7] J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[8] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *Graph Algorithms Appl.*, 1(1):1–25, 1997.

[9] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout.* Teubner, Stuttgart, 1990. Also published by Wiley in Chichester UK.

[10] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[11] K. Mehlhorn and P. Mutzel. On the embedding phase of the hopcroft and tarjan planarity testing algorithm. *Algorithmica*, 16(2):233–242, 1996.

[12] F. Shahrokhi, O. Sýkora, L. A. Székely, and I. Vrto. On bipartite drawings and the linear arrangement problem. *SIAM Journal on Computing*, 30(6):1773–1789, 2001.

[13] W.-K. Shih and W.-L. Hsu. A new planarity test. *Theor. Comput. Sci.*, 223(1-2):179–191, 1999.

[14] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man Cybernetics*, 11(2):109–125, 1981.

[15] I. G. Tollis, G. D. Battista, P. E. (Editor), and R. Tamassia. *Graph Drawing: Algorithms for the Visualization of Graphs.* PH, 1998. ISBN: 0–133–01615–3.

[16] V. Valls, R. Marti, and P. Lino. A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *Journal of Operational Research*, 90:303–319, 1996.