Contemporary Engineering Sciences, Vol. 4, 2011, no. 7, 301 - 309

Layered Drawing of Planar Graphs Inside Simple Polygons

Fatemeh Rajabi Alni

Department of Computer Engineering, Islamic Azad University
North Tehran Branch, Tehran, Iran
Corresponding author: fatemehrajabialni@yahoo.com

Alireza Bagheri

Department of Computer Engineering & IT
Amirkabir University of Technology, Tehran, Iran
ar_bagheri@aut.ac.ir

Abstract

A considerable amount of work has been done on graph drawing field but much of it is related to drawing graphs on unbounded surfaces. Drawing graphs on two dimensional surfaces with given size and shape is studied in a few papers. The aim of this paper is to improve the graph drawing algorithms which draw graphs inside simple polygons. We present a new algorithm for drawing plane graphs, i.e. planar graphs with given embeddings, inside simple polygons. The algorithm uses external Hamiltonian cycles of plane graphs, i.e. Hamiltonian cycles with at least one edge on the outer face.

Keywords: Graph drawing, planar graphs, Hamiltonian cycle

1. Introduction

The aim of graph drawing is to represent graphs on the plane such that they can be easily understood by means of viewing. Graphs can be used for representing any information that can be modeled as objects and relations between them. So graph drawing has many applications in the various fields, such as VLSI circuit design, social network analysis, cartography, and bioinformatics.

A graph is planar if it can be drawn on the plane such that no two edges intersect in the drawing. If planar embedding of a planar graph in the plane is given it is called a plane graph. In this paper, we present a new algorithm for planar poly-line drawing of plane graphs inside simple polygons. The input of the algorithm is a plane graph, which has an external Hamiltonian cycle, and a simple polygon; the output of the algorithm is a planar poly-line drawing of the plane graph that its edges are drawn inside the given simple polygon.

Various graphic standards are used for drawing graphs. In this paper, vertices are presented by points and edges are presented by polygonal chains. The quality of a drawing can be measured by drawing criteria, such as the number of edge crossings, the number of edge bends, planarity, distribution of vertices, and area. We focus on planar drawing of graphs and uniform distribution of nodes.

There are few studies on drawing graphs inside simple polygons despite of many applications of it [1-6], and these studies are very general. We concentrate on drawing plane graphs inside simple polygons. The remainder of the paper is organized as follows. In section 2, we explain the drawing algorithm. In section 3, we present an example for the drawing algorithm. Finally we conclude in Section 4.

2. The drawing algorithm

In this section, we present a new algorithm for planar poly-line drawing of a plane graph, which has an external Hamiltonian cycle, inside a given simple polygon on the plane. We follow the idea of using Hamiltonian cycles of planar graphs that already have been used by Kaufman and Wiese [8]. Our algorithm has five phases that are explained in the following.

Phase 1. Finding the Hamiltonian cycle

In this phase, we find an external Hamiltonian cycle of the given graph. Finding the Hamiltonian cycle of a graph is an NP-complete problem, so we use the linear-time algorithm that finds the external Hamiltonian cycle of a four-connected planar graph [7]. If the input graph is a plane graph but not a four-connected one, we convert it to a triangulated plane graph and convert it to a four-connected one by means of the linear-time algorithm that is presented by [9].

After finding the external Hamiltonian cycle of the input plane graph we index the vertices of the graph. The external Hamiltonian cycle has at least one edge on the outer face of the plane graph. We select one of the external edges, called e, and index the vertices of graph by $v1$to vm such that e be incident to $v1$ andvm.

Then we rank edges of the graph and also assign the direction of drawing of the edges, that is clockwise or counterclockwise. The direction of edges are important since all vertices of the graph are placed around a cycle, so there is two way for connecting two vertices, clockwise and counterclockwise. We assume that always the edges start from the vertex with smaller index and ends at vertex with bigger index.

Using the external Hamiltonian cycle, we can divide the edges of the graph to three group: the edges of Hamiltonian cycle, the edges inside Hamiltonian cycle and the edges outside the Hamiltonian cycle. The edges of the Hamiltonian cycle take rank zero and the edges that are inside and outside the Hamiltonian cycle are ranked independently, so the inner and outer edges may have the same numbers.

We first rank the edges that are inside the Hamiltonian cycle, called inner edges. If the indices of start and end of the edge be k and k', the direction of an edge is selected such that the connecting polygonal chain has minimum length, that is if $|k' - k| \leq \frac{m}{2}$ the direction of the edge is clockwise else it is counterclockwise. After assigning the direction of the edge we assign its rank considering the direction. We count the number of inner edges that are between start and end vertices and their directions are as the direction of the edge.

For ranking the outside edges after assigning the direction of them, we move from the Hamiltonian cycle toward outside the cycle and count the number of outer edges that are between start and end vertices and their directions are as the direction of the edge.

Phase 2. Layering the surface of the polygon

In this phase, we layer the surface of the polygon. For this phase we use the onion layers used in [1] by a small change. In [1], the onion layers are constructed around the longest path of the dual graph of triangulation of the polygon, but we construct the layers around the entire dual graph. So we use the entire surface of the polygon, and this is appropriate for the multi branch simple polygons.

Phase 3. Mapping the Hamiltonian cycle on the polygon layers

In this phase, we map the Hamiltonian cycle of the plane graph, found in previous phase, to one of the layers of polygon such that the number of layers that are inside it are equal to the maximum rank of the edges inside Hamiltonian cycle,R_{in}, and the number of layers that are outside it are equal to the maximum rank of the edges outside Hamiltonian cycle,R_{out}.

Starting from the outer layer we select the $\mathrm{Rout} + 1\mathrm{th}$ layer as the Hamiltonian cycle layer. After selecting the layer of the Hamiltonian cycle, we rank this layer with zero and rank other layers starting from this layer. For ranking the layers of the polygon that are inside this layer we move toward inside the polygon and rank the layers such that the first layer is ranked one, the second layer is ranked two and so on.

For ranking the layers of the polygon that are outside this layer we move toward border of the polygon and rank the layers such that the first layer is ranked one, the second layer is ranked two and so on. Then we divide the layer with rank zero, i.e. the layer of the Hamiltonian cycle, to m equal partitions and put the vertices of the graph in the dividing points starting from 1.

Phase 4. Drawing the edges of the graph

In this phase, using the rank of edges we draw them. For drawing each edge we use the layer of polygon that its rank is as same as the edge's rank. After selecting the appropriate layer of polygon we connect two vertices of the edge using the polygonal chain of that layer. We always assume that the drawing of the edge start from the vertex with the smaller index. If the direction of the edge is clockwise we start from the vertex with the smaller index and move in clockwise direction toward the other vertex, i.e. the vertex with the bigger index. In this way, the edges of the graph can be drawn without crossing and the number of the bends of each edge is $\mathrm{O(n)}$, since each polygon layer has $\mathrm{O(n)}$ bends.

3. The time complexity

In this section, we discuss about the time complexity of our algorithm. In phase 1, we find an external Hamiltonian cycle of the input plane graph that has m node. This can be done in $\mathrm{O(m)}$ time by the Chiba and Nishizeki's algorithm [3]. In phase 2, we construct the layers of the given simple polygon. We use the onion layers of Bagheri [1] with a little modification that doesn't change its time complexity, so if the input simple polygon with n vertices is not a rectilinear polygon the time complexity of this phase is $\mathrm{O(nL)}$ else its time complexity is $\mathrm{O(n^2 + nL)}$. Where L is the number of layers of the input plane graph and so it is bounded by $\mathrm{O(m)}$, the time complexity of this phase for general simple polygons is $\mathrm{O(nm)}$ and for rectilinear polygons is $\mathrm{O(n^2 + nm)}$.

In phase 3, we map the external Hamiltonian cycle on one of the polygon layers that takes $\mathrm{O(m)}$ time. In phase 4, we draw the input graph edges. The bends of each edge is bounded by $\mathrm{O(n)}$ and the number of edges are bounded by $\mathrm{O(m)}$, so this phase takes $\mathrm{O(mn)}$ time. We conclude that the time complexity of our new algorithm for general simple polygons is bounded by $O(nm)$ and for rectilinear polygons is bounded by $O(n^2 + nm)$.

4. An example

In this section, we present an example for the algorithm to show its act in practice. In this example we draw the plane graph of Figure1 inside the simple polygon of Figure 2.

In Figure 3 we show the triangulation dual of the polygon of Figure 2 with dashed line. Figure 4 shows the layering of the polygon of Figure 2 around the dual graph. In Figure 5 we draw the planar graph of Figure 1 on the layers of polygon of Figure 2. The drawing before reducing the edge bends is given in Figure 6. The final drawing is given in Figure 7.

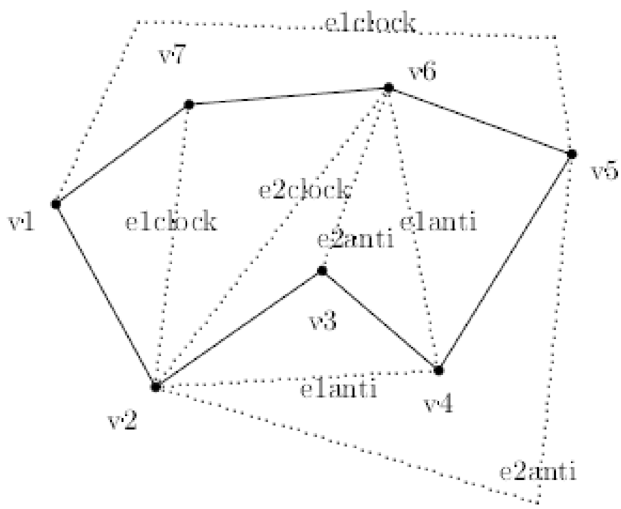

Fig 1. A plane graph with an external Hamiltonian cycle

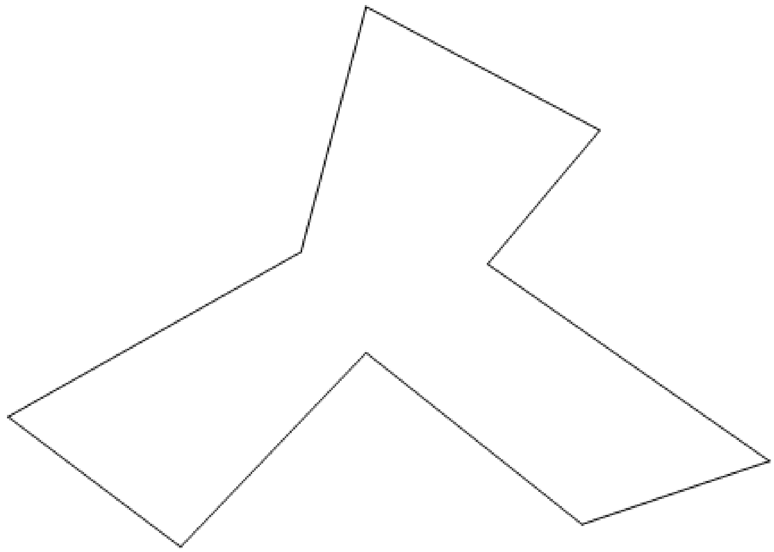

Fig 2. A simple polygon

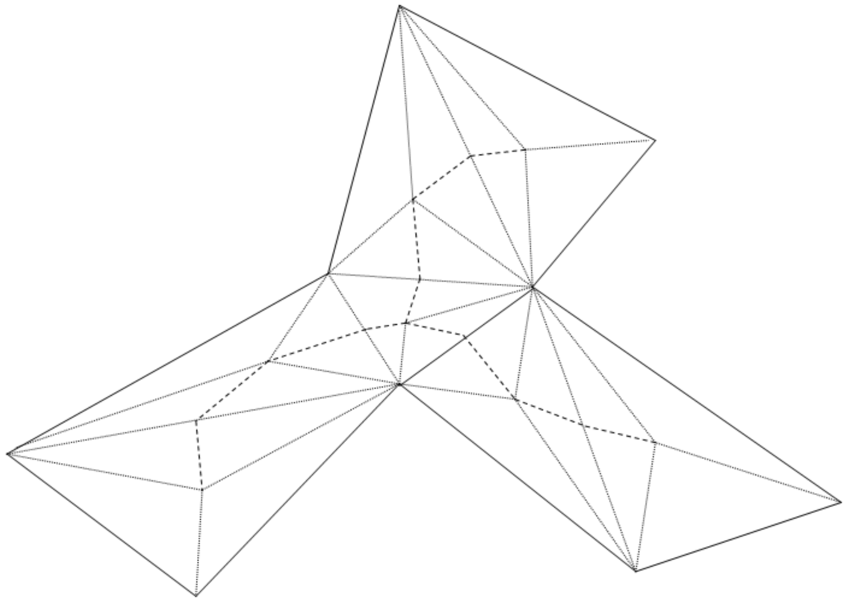

Fig 3. . The triangulation dual of the polygon of figure 2

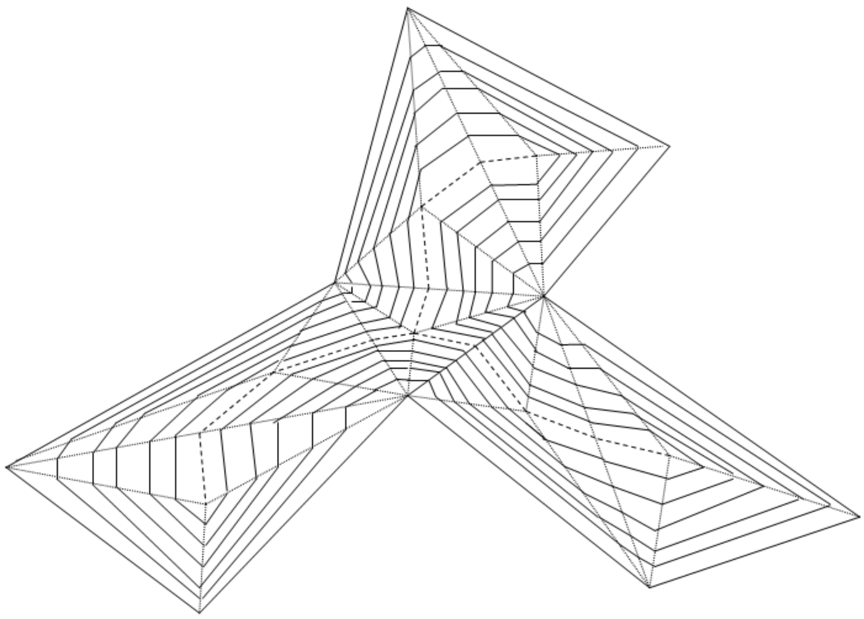

Fig 4. The layering of the polygon of figure 2

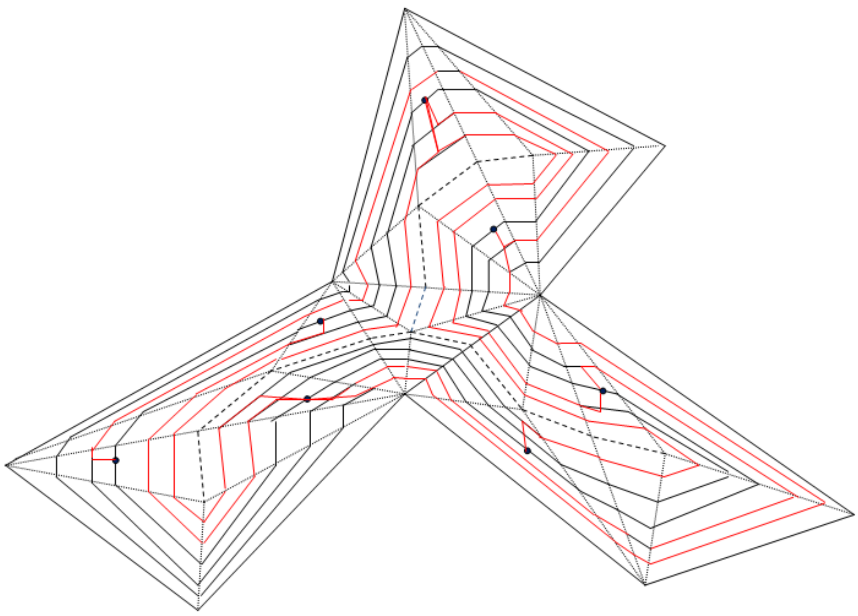

Fig 5. The drawing of the graph of Fig1 on a layered surface of a simple polygon

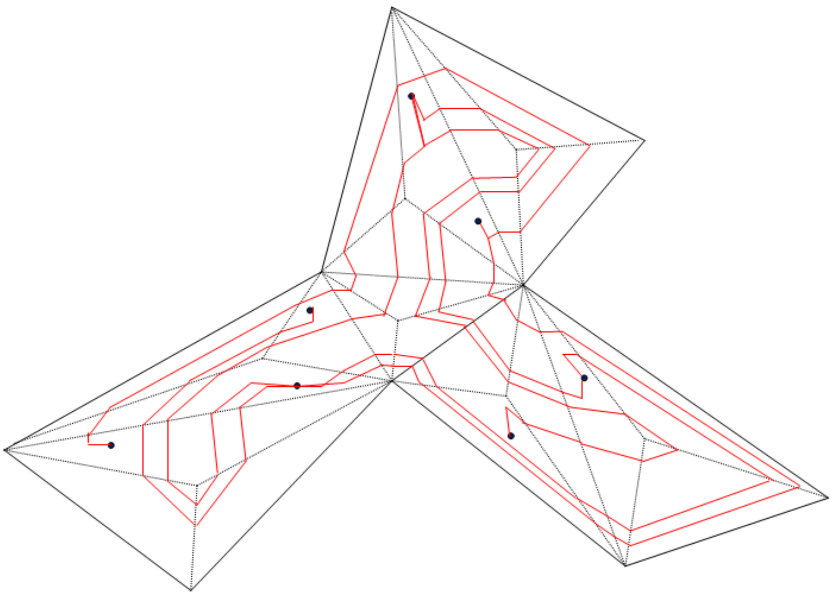

Fig 6. The drawing of the graph of figure 1 on a simple polygon

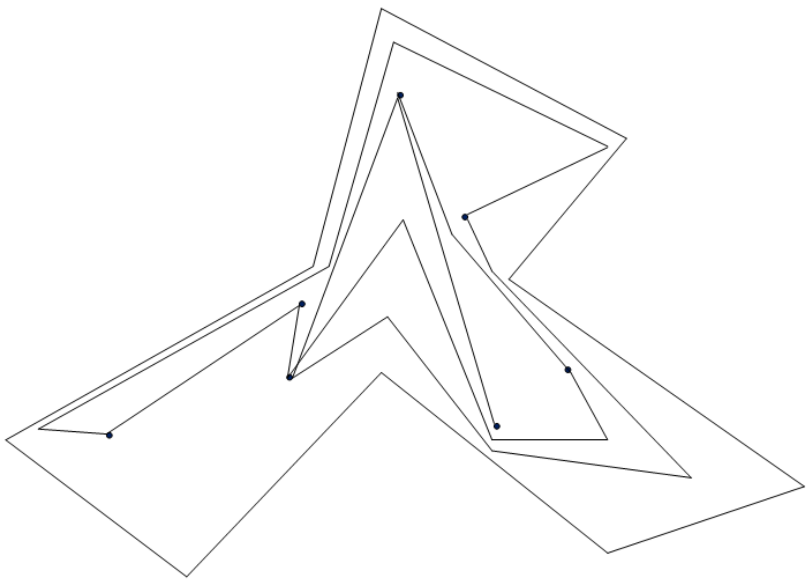

Fig 7. The final drawing of the graph of Figure 1 on a simple polygon

5. Conclusion

In this paper, we present a new algorithm for drawing a planar graph, which have an external Hamiltonian cycle, inside a simple polygon. Our algorithm can be used for general simple polygons but its main goal is to draw planar graphs inside multi branch simple polygons and use the entire surface of these polygons. This leads to a better node distribution than the previous algorithms.

As a future work, we can concentrate on restricted types of the graphs and polygons and present algorithms that especially work better for these classes of graphs and polygons.

References

[1] A. Bagheri, Drawing Planar Graphs inside Simple ploygons, Ph.D Thesis, Amirkabir University of Technology, Tehran, Iran, 2006.

[2] A. Bagheri, and M. Razzazi, How to Draw Free Trees inside Bounded Rectilinear Polygons, International Journal of Computer Mathematics, 81(11), Nov. 2004, pp. 1329-1339.

[3] A. Bagheri, and M. Razzazi, Drawing Free Trees inside Simple Polygons using Polygon Skeleton, Computing and Informatics, 23(3), 2004, pp. 239-254.

[4] A. Bagheri, and M. Razzazi, How to Draw Free Trees inside Bounded Simple Polygons, Journal of Universal Computer Science, 11(6), 2005, pp. 804-829.

[5] A. Bagheri, and M. Razzazi, On Drawing of Free Trees on 2D Grids which are bounded by Simple Polygons, Scientia Iranica, 13(4), 2006, pp. 387-394.

[6] A. Bagheri, and M. Razzazi, Drawing Complete Binary Trees inside Rectilinear Polygons, International Journal of Computer Mathematics, 87(14), 2010, pp. 3138-3148.

[7] N. Chiba, and T. Nishizeki, The Hamiltonian Cycle Problem is Linear-Time Solvable for 4-Connected Planar Graphs, J. Algorithms, 10, 1989, pp. 189-211.

[8] M. Kaufmann, and R. Wiese, Embedding vertices at points: Few bends suffice for planar graphs, Journal of Graph Algorithms and Applications, 6(1), 2002, pp. 115-129 .

[9] M. Formann, and F. Wagner, The VLSI layout problem in various embedding models, Graph-Theoretic Concepts in Comp. Science (16th Workshop WG'90), Springer-Verlag, Berlin/Heidelberg, 1992, pp. 130-139.

Received: May, 2011