IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No 2, July 2013
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org

New algorithm for automatic visualization of metro map

Somayeh Sobati moghadam
Electrical and computer engineering faculty, Hakim Sabzevari University, Sabzevar, Iran.

Abstract
The metro map is a type of diagram that illustrates transportation network. The automated method for visualisation of graphs with metro map layout is a challenge. In this paper two method for metro map automatic visualization is proposed. These methods use theory of graph and a grid for node coordination. In second method a temperature for graph is considered and the position of node is changed according to simulated annealing. The results show that these algorithms can produce good Metro Maps. The first algorithm has a good result for the graphs with many connections and the second, is better for very dense graphs.
Keywords: *Metro map layout, simulated annealing, graph***.**

1. Introduction

The metro map is a type of diagram that is used for illustrating transportation networks. Designing a schematic map today is still a challenge. Metro map visualizes the interconnections of rail road networks, and people are able to use it quickly. Many people around the word know how to read route maps for metros and buses [1]. The *metro map* can be presented as a graph; it consists of a set of lines which have intersections or overlaps. In all metro maps it is common to consider the graph in which, metro stations are considered as vertices and their interconnections as edges. The lines are straightened and restricted to horizontals, verticals and diagonals at 45°. For forming a good metro map layout, a set of aesthetic criteria should be defined. Traditionally, metro maps are drawn manually [2] and there was no automatic way to produce this diagram; the cartographer must decide where to put the stations and how to draw the lines in the diagram. In complex and large networks it is not simple. In the other hand this kind of diagram could be used as metaphor for abstract entities. The metro map diagram is easy to use and understanding hence it can be use for illustrate the data and relationship between them. For example in [3] it is used for organizing web-based learning resources. There are different types of this usage [4, 5, and 6]. First, we have to know what kind of map is suitable for metro map and what is a “nice” metro map? The map should be as readable and clear as possible without displaying unnecessary details [4]. The lines in the map must be displayed horizontally, vertically or diagonally at 45◦.This final layout is called “octilinear “layout. The final map must have as minimum bends as possible without edge crossing and overlapping of labels. Hong and al. [1] proposed five methods. For automatic drawing metro map; they used a modified embedded spring system. Stott and Rodger [1] are investigating another approach with considering multi criteria optimization. Nollenburg and Wolff [7], is proposing an automated method by the definition of hard and soft constraint.

2. Preliminaries

In this section the main concepts in graph drawing is reviewed. In the state of the art the related work is described and in the final section the new algorithm is proposed.

2.1 Graphs

Graphs are important because they can be used to represent essentially *any* relationship between entities. Graphs visualize the information for the users, and provide important information about the objects. For example, graphs can model a network of roads, with cities as vertices and roads between cities as edges. A nice layout of graph aids user to find immediately the information that he is looking for.
A graph $G= (V, E)$, consists of a set of vertices V, $|V|=n$, and a set of edges E, $|E|=m$. An edge is a pair $e= (u, v)$, $u, v \varepsilon V$. Two adjacent vertexes are connected by an edge. The degree of a vertex is the number of edges that incident to this vertices
A path in G is a sequence of distinct vertices of G like $(v_1, v_2 ... v_t)$ such that $v_i v_{i+1} \in$ E for $1 < i < t$.
A graph is called planar if it can be drawn in the plan without edge crossing.
In this paper, we need to use another definition of a graph based on lines. So, our problem, a graph $G= (V, E, P)$, will consist of a set of vertex V, a set of edges E, and a set of paths P. In any path there are many nodes with degree two.

2.2 Graph layout

Graph drawing applies topology and geometry to draw two and three-dimensional representation of graphs. Very different layouts can correspond to the same graph. There are different graph layout strategies. Fig.1 shows an example of a graph layout. In *straight line* drawing each edge is drawn as the straight line between the vertices. *Orthogonal layout* the edges are drawn as polygonal chains of horizontal and vertical line segments. We will use the *octilinear* layout for metro map drawing. In octilinear layout all edges are horizontally, vertically or 45°. The advantage of using this layout in comparison of orthogonal layout is that the maximum possible vertex degree increases from 4 to 8. Some drawings are better

IJCSI
www.IJCSI.org

than the others. Aesthetic criteria attempt to characterize readability of a layout. Various attempts have been made to specify the readability of a layout that comprises:

- minimize ***crossings***
- minimize ***area***
- minimize ***bends*** (in orthogonal drawings)
- minimize ***slopes*** (in polyline drawings)
- maximize ***smallest angle***
- maximize display of ***symmetries.***

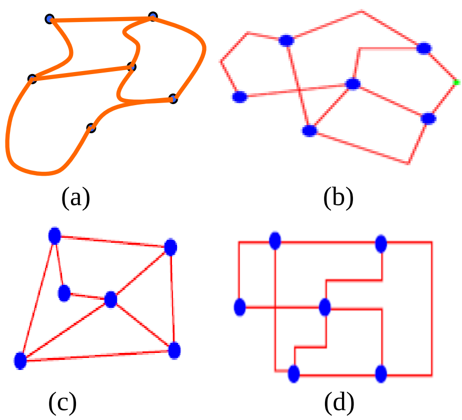

Fig.1 Four layouts of the same graph G. In (a) drawing with simple open curves. In (b) polyline drawing, (c) straight line, (d) orthogonal drawing of G.

In the other hand there are graphical properties for readability of a layout. For example the shape and size and color of vertices; the shape (curve, line), thickness and color of edges. In any algorithmic method of graph drawing these properties are left to the decision of user. Another property that must be taken into account is the graph labeling. The size and place of labels are important to minimize overlaps.

2.3 Simulated annealing

Simulated annealing is a very general optimization method, which stochastically simulates the slow cooling of a physical system. The idea is that there is a cost function H which associates a cost with a proposing change and their accepting or rejecting each change. Having proposed a change we may evaluate the change in H. The proposed change may be accepted or rejected by the criterion; if the cost function decreases the change is accepted. The system soon reaches a state in which none of the proposed changes can decrease the cost function. The simulated annealing is used in our method. In second method, a temperature is calculated for the system, in each iteration, if it is not zero the state of system changes until the temperature reaches at zero.

3. Related works

Automatic drawing a graph with a desired layout is a challenge. For octilinear drawing some methods are proposed. Each method has several advantage and disadvantage. Nollenberg and Wolff [7] have investigated an algorithm for automatically drawing metro maps. A planar graph G with maximum degree 8 and a set of paths or cycles, L, is considered. A list of soft and hard constraints is specified. Then they have shown that it is NP-hard to decide whether a drawing of G can satisfying all hard constrains. They have presented a mixed- integer linear program (MIP) which finds a drawing that fulfills all hard constraints. This program optimizes a weighted sum of costs corresponding to the soft constraints. They have shown how to include vertex labels in the drawing [7]. The authors have implemented this approach for six real-world examples. Stott and Rodgers in [1] have proposed an automatic method for drawing metro maps. They have considered the metro map as a graph. The graph is embedded on an integer square grid. It is important to mention that more than one node does not share the same grid intersection. In the case of contention for a particular intersection, the node being snapped should be moved to the nearest grid intersection that is vacant [1].

They have implemented a total of eight criteria. Each criterion measures some geometric property of the map such as the length of edges or edges crossings and is weighted. The nodes and labels are repositioned such that the total of the weighted criteria is always reduced. Hill climbing algorithm is based on the total of the weighted criteria, *t*, which *t* has given by:

$$t = w_1c_1 + w_2c_2 + w_3c_3 + w_4c_4 + w_5c_5 \qquad (1)$$

Which c_1, c_2, c_3, c_4, c_5 denote respectively the number of edge crossings, the edge length, the angular resolution, the line straightness and four-gonality respectively. An initial value of *t*, is t_0.The nodes are moved to the location that has the smallest value of *t*. When selecting a position to move a node there are a number of points that need to be considered: the total value of the weighted criteria, *t*; whether or not another node occupies that grid intersection; whether moving the node would occlude other nodes or edges; how far to move the node; whether the distance to move the node is reduced with each iteration (cooling); and whether the cyclic ordering of edges incident to a node would change [1]. In each movement for a node, the initial value of *t*, t_0, is calculated. A set of locations, *T*, is remembered for each movement where $t < t_0$. The node is moved to the location in *T* that has the smallest value of t_0.

Force directed algorithms are used for graph drawing but they introduce a lot of edge crossing. F. Bertault [8] has proposed an algorithm based on a force-directed approach. A force F(*v*), for each node *v*, is computed. Then each node is moved in the direction of F(*v*). Three kinds of forces are considered: the attraction forces between nodes, the repulsion forces between each pair of nodes, and the repulsion forces between nodes and edges [8]. For each node, *v* a zone *Z (v)* is associated that preserves the crossing properties between edges. This zone is divided into eight zones. For each node the size of the zones is updated. PrEd algorithm produces drawings which are more aesthetically pleasing that the initial layout and the final layout of a given graph has the same embedding as the original graph [9].Hong and Merrick [9] have investigated the new problem of automatic metro map layout. They have defined a set of aesthetic criteria and presented a method to produce the layout automatically. They have combined several differ[illegible] layout methods. The algorithms that they have utilized are GEM algorithm, PrEd algorithm and a magnetic spring algorithm. They have added magnetic spring forces to the spring embedded model. Edges are drawn straight-line, without bends. First, each graph is simplified by removing the nodes of degree two. In their methods the following criteria are used for producing a good metro map layout:

C1: Each line is drawn as straight as possible.
C2: No edge crossings are present.
C3: No overlapping of labels occurs.
C4: Lines are mostly horizontal or vertical, with some at 45°.
C5: Each line is drawn distinctly, with a unique color [9].

They have designed layout methods based on the first four these criteria. They have investigated five different methods:

(1) Method 1: The GEM algorithm.

(2) Method 2**:** The GEM algorithm with edge weights
- Simplifying the metro map graph G'
- Producing a layout L' of G' using GEM algorithm with edge weighted
- Reinserting the removed vertices.

(3) Method 3: Modifying PrEd algorithm with edge weights
- Simplifying metro map graph
- Producing an initial layout of G' using GEM algorithm
- Producing a better layout using PrEd algorithm by edges weights
- Reinserting the removed vertices.

(4) Method 4: Modifying PrEd algorithm with edge weights and orthogonal magnetic spring algorithm
- Simplifying the metro map
- Producing an initial layout using GEM algorithm
- Producing a better layout using PrEd algorithm by including edge weights an orthogonal magnetic field forces
- Reinserting the removed vertices.

(5) Method 5: Modifying PrEd algorithm with edge weights, orthogonal and 45° magnetic forces
- Simplifying the metro map graph
- Producing an initial layout using GEM algorithm
- Producing a better layout using PrEd algorithm with edge weights, orthogonal and 45°magnetic forces
- Reinserting the removed vertices.

All of this method has their advantage and disadvantage. The Hong and Merrick method do not present a good layout. In Stott and Rodgers method, there are several edges which aren't octilinear. The Nollenburg and Wolff method has a good final layout but their method is difficult to apply.

4. New Algorithms

In this section, we introduce the metro map layout problem and propose a new method for automated drawing of metro map. These methods are simple and produce the result in acceptable time.

4.1 The Metro Map Layout Problem

Let *G= (V, E, P),* be the plane input metro map graph, consisting of a set of vertices *V;* $|V|=n$, and a set of paths *P*. We consider that *G* is planar. If it is not planar, we planarize it by adding some dummy vertices. For simplification, we consider the maximum degree of vertices is 8. The main problem of this map layout is defined as follows:

Input : a metro map graph *G* with a set of paths.
Output: an octilinear layout of *G*.

First, we explain the preprocessing step. The degree of each node *v* in graph is calculated; according to the number of crossing edges. For each path; $P_i= \{v_t | 1<t<n\}$; a degree of path is calculated; by calculating the sum of degree of its vertices.

$$Degree\ P_i=\sum_{v_t \in P_i} \deg v_t \qquad (2)$$

The path that has the maximum degree is called the *Maximum path* in the graph. We simplify *G* by removing the vertex of degree-2. These nodes do not contribute to the initial embedding. The resulting graph contains intersection nodes and nodes with degree one (extension). In our approach, the graph is embedded onto an integer square grid (fig.2.). The size of the grid is defined by *g*, that *g* is calculated by:

$$g=2*(\text{long of } Maximum\ path) \qquad (3)$$

This guarantees the sufficient space for embedding the graph in the grid. The *Maximum path* is put in the middle of grid. For each node, 8 directions are considered. The 8 directions is shown in fig.3. Two methods for embedding the graph in the grid, is proposed. In method 1, the nodes are put in grid according to the 8 directions around each node. In the method 2 each node has a weight corresponding to the number of degree-2 nodes that are removed. The temperature of the system is defined as weights. In each iteration in this method the temperature is calculated and if it is not zero the position of the weighted nodes is updated. Each time that a node is put in grid this step is reiterated. This guarantees the uniform edges.

4.2 Method 1

For each node of the *Maximum path*, all of its neighbors are put in the directions (for each node of *Maximum path* the direction 1 and 3 are already occupants).Then for the new neighbors, this step is repeated and this step is repeated until all of nodes, are put in grid. After this step, all nodes of degree two are inserted. Finally, each node has coordinated in grid and all lines are horizontally, vertically or 45°.

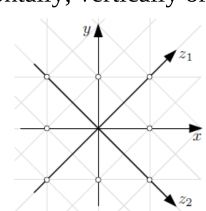

Fig.2 the grid for embedded the graph.

4.3 Method 2

Method 2 is used a preprocessing step, after removing the degree-2 nodes; consider nodes weight according to the number of the degree-2 nodes that is removed. In each iteration, after having put the nodes in grid, the temperature of graph is updated this is the sum of weights of the node in grid. If it is not zero, the position of nodes is update. There is a repulsive force between two nodes that have a weight. This repulsive force is computed and the position of nodes weighted, is changed. Their position change according to the number of nodes that are deleted. This step is repeated until all nodes are put in grid and the temperature of system is zero. The result is a graph with octilinear layout, and the length of edges is uniform.

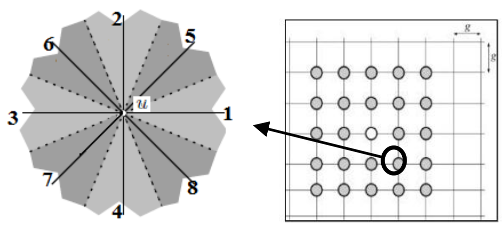

Fig.3 the 8 direction for each nod in grid.

```
Procedure Method1 is
-- Input:
--       G= (V, P) graph where
--        V= set of nodes (URL)
--       P=set of paths
--      max_path=the path with maximum weight
--    l_max_path=long of maximum path
--   g    the size of each square in grid
-- Output: for each v∈V, a coordination

For all   v∈V do
    Calculate the degree of v;
For all   p∈P do
    Calculate the degree of p;
Put the max-path In middle of grid
While all v∈V  is put in grid
For all v∈max_path do
    For all neighbors of v do
        put the neighbor in 8 direction around v;
end;
```

```
Procedure Method2 is
-- Input:
--        G= (V, P) graph where
--         V= set of nodes (URL)
--        P=set of paths
--       max_path=the path with maximum weight
--     l_max_path=long of maximum path
--   g    the size of each square in grid
-- Output: for each v∈V , a coordination

For all   v∈V do
Calculate the degree of v;
   Calculate the weight of v;
For all   p∈P do
Calculate the degree of p;
Put the max-path in middle of grid
Compute the temperature;
Choose a weighted vertex to update;
Update position;
While all v∈V  is put in grid and T>0 do
For all  v∈max_path do
    For all neighbors of v do
        Put the neighbor in 8 direction around v;
      Choose a vertex v, to update;
    Compute its impulse;
  Update its position and temperature;
end;
```

Fig.4 the main loop of method1 and method2.

5. Conclusion

The Metro Map Metaphor can be used for visualising abstract information. A Metro Map can be shown as a graph. The automated method for visualisation of graphs is a challenge. In this work we have studied the problem of automatically drawing metro maps. We have proposed two methods for automated drawing metro maps. In each method, the graph is put in a grid. Then each node has a coordinate and the final graph has an octilinear layout. In second method, the graph is weighted and a temperature is used. In each iteration, the temperature decreases. The final results show that the second method has a good result for the dense graph; and the first method has satisfactory results for the graphs that have many intersections. The examples show that the results have a good octilinear layout, but in some cases there are some label overlaps.

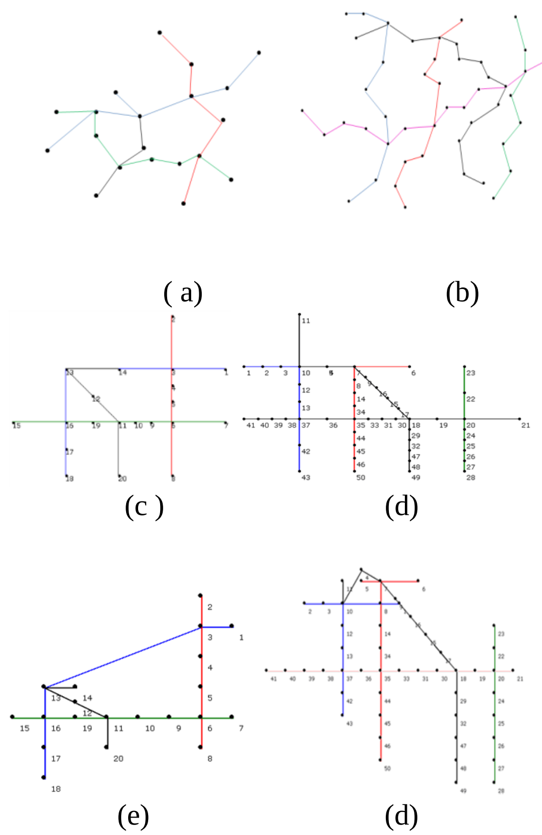

Fig.5 The result of method 1 and 2, (a)and (b) the initial graph, (c) and (d) the result of method 1, (e) and (d) the result of method 2.

References

[1] J. M. Stott and P. Rodgers ,J. C. Martínez-Ovando, S G. Walker , “Automatic Metro Map Layout Using Multicriteria Optimization”, journal of visualization and computer graphics, pp. 101-114, 2011.

[2] S. H. Hong, D. Merrick, and H.A.D. do Noscimento, "Automatic Visualisation of Metro Maps," *J. Visual Languages and Computing*, vol. 17, pp. 203-224, 2006.

[3] T. Bang, K. Gronbak, P.S. Hansen, Using a metro map metaphor for organizing web-based learning resources, In Proc. of ED-MEDIA 2002, pp. 647–652, 2002.

[4] K. V. Nesbitt. Getting to More Abstract Places with the Metro Map Metaphor,2003.

[5] E. S. Sandvad, K. Gronbak, L. Sloth and J. L. Knudsen, Metro Map Metaphor for Guided Tours on the Web: the Webvise guided Tour system, *Proc. of International Conference on World Wide Web*, pp. 326-333, 2001.

[6] T. Bang. Knowledge Sharing in a Learning Resource Centre by Way of a Metro Map Metaphor. 67th IFLA Council and General Conference, Boston, 2001.

[7] M. Nollenburg and A. Wolff. A Mixed Integer Program for Drawing High Quality Metro Maps. In Proc 13th International Symposium on Graph Drawing. Verlag Berlin Heidelberg, Springer, 2005.

[8] F. Bertault, A Force-Directed Algorithm that Preserves Edge Crossing Properties, *Graph Drawing 99*, LNCS 1731, pp. 351-358, Springer Verlag, 1999.

[9] S. Hong, D. Merrick, Automatic Visualization of Metro Maps. Journal of Visual Languages & Computing, vol. 17, PP. 203-224, 2006.

[10] P.Chaleat, D. Charnay. Programmation Html, JavaScript, Paris, Eyrolles, 1998.

[11] R. Davidson and D. Harel. Drawing Graphs Nicely Using Simulated Annealing. Technical ReportCS89-13, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, Israel, ACM, 1993.

[12] A. Frick, A. Ludwig and H. Mehldau, A Fast Adaptive Layout Algorithm for Undirected Graphs, Proc. of Graph Drawing 94, LNCS 894, pp. 388-403, Springer Verlag, 1995.

[13] H. de Frayssex, J. Pach, and R. Pollack. How to Draw a Planar Graph on a Grid. Combinatorial, pp. 41-51, 1990.

[14] T. M. J. Frachterman and E. M. Reingold. Graph Drawing by Force-Directed Placement. Software Practice and Experience, pp. 1129-1164, 1991.

[15] S. Hong, D. Merrick, H. A. Nsacimento. The Metro Map Layout Problem. Australasian Symposium on Information Visualization, Christchurch, 2004.

[16] W. He and K. Marriott. Constrained graph layout. An international journal, pp.289-314, 1998.

[17] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, ‘Optimization by simulated annealing’, *Science*, 220, (4598), pp. 671–680, 1983.

[18] R. Laurini. Information Systems for Urban Planning. A Hypermedia Cooperative Approach, Taylor and Francis, 2001.

[19] R. Laurini, D. Thompson. Fundamentals of Spatial Information Systems. Academic Press, 1993.

[20] P. Lacomme, C. Prints, M. Sevaux. Algorithmes de Graphes. Ed. Eyrolles, 2th, 2003.

[21] S. Maccari, S. Martin. HTML & JavaScript. Paris, Micro Application, 2004.

[22] T. Masui, Evolutionary Learning of Graph Layout Constraints from Examples. ACM, November 2-4, pp. 103-108, 1994.

[23] D. Merrick and J. Gudmundsson. Path simplification for metro map layout. In Pro Graph drawing, pp. 258-269, 2007.

[24] O’Reilly book network, http://www.oreilly.de/artikel/routemap.pdfi.

[25] R. Tamassia. On Embedding a Graph in the Grid with the Minimum Number of Bends. SIAM Journal of Computing. Vol. 16. No. 3. pp. 421-444. June 1987.

Somayeh Sobati Moghadam is lecturer in Hakim Sabzevari University. She received her B.Sc. degree in Applied Mathematics in computer in 2001 from Amir Kabir University of technology and M.Sc. degree in IT in 2007 from INSA-Lyon- France. From 2010 she is lecturer in Hakim Sabzevari University. Her research interests include Information visualization, graph theory, GIS and network security.