

Research on Force-directed Algorithm Optimization Methods

Wang Lingling¹, Wang Xianshui¹, Wang Qiuquan², u Mei¹

¹Jiangcheng College, China University of Geosciences, Wuhan Hubei 430200, China

²West Section Headquarters, China Transportation Construction Co., Ltd, Xian 710065, China
winniay@yeah.net

Abstract - In view of the fundamental Forces-directed iteration the shortcoming of the slow speed, In this paper, using the method of Hyper node and Multilayer method to improve the algorithm and applied to the large-scale graphics drawing automatically, Hyper node Method is the distance of the vertex as a Hyper node to the whole processing, with the Octree data structure to define and iterate over Hyper node. Multilayer method using maximum independent edge set or maximum independent set of points to simplify the graphics on a large scale, by Forces-directed of solving optimization simplified diagram and then find the original inverse iteration. Experiments show that the method of Automatic drawing to draw true and vivid, drawing speed is greatly increased and well expressed the people's automatic drawing intentions with Multilayer method and Hyper nodes.

Index Terms - Force-directed, scheduling network, Automatic drawing, Hyper node, Multilayer method

1. Introduction

Force-directed graph drawing algorithms is used to convert the information of mathematical information of graphics into two-dimensional or three-dimensional geometric information, based on the relative positions of nodes and edges, automatically draw out the graphics, and thus realize the visualization of graphics. Currently, there are many algorithms for different purposes of drawing, straight line, rectangle method, grid method, circular method, discounting method are based on mathematical methods, according to the mathematical characteristic of graphics, they obtain graphic coordinates of the vertices by equation.

Force-directed methods in graph drawing date back to 1960s, and it achieves a further development in 1980s. It is particularly suitable for automatic drawing undirected graph. In 1991, Fruchterman, Thomas M. J, etc proposed Graph Drawing by Force-Directed Placement [1]. In 2000, Wuhan university of Huang Jingwei explore how to turn the general undirected graph drawing into function optimization, using genetic algorithms to find the approximation of optimal solution of the objective function ,thus develop a general framework of undirected graph automatically drawing algorithm[2]. In 2010, China university of geosciences of ZhangQin apply genetic algorithm to automatic drawing, it imitates biological evolutionary theory, evolutionary genetic algorithm will be randomly distributed shaped for the optimal structure [3]. In the same year, SunQiang succeed in applying force-directed methods to the optimization of peak power consumption, Xinhua studied the distribution network boot line diagram automatic mapping with the use of force-directed methods. In 2012, Chernobelskiy, R show that approaches

produce near- Lombardi drawing, with one being slightly better at achieving near-perfect angular resolution[4] . Kobourov, Stephen G. Embedders put forward to that Force-Directed Graph Drawing Algorithms [5].Bannister, M. J show that force-directed layout algorithms produce graph drawing by resolving a system of emulated physical forces [6].

The iterative forces-directed algorithm of spring electrical model can give better painting for small-scale (for example, dozens of vertices).However, in view of large-scale map (thousands of vertices), common iterative algorithm (iterative speed slow and easy to fall into local optimal) two weaknesses.

So, in view of the fundamental forces-directed iteration the shortcoming of the slow speed, this paper, using the method of Hyper node and Multilayer method to improve the algorithm and apply them to the large-scale graphics automatically drawing, explores how to draw a graph which meet the following four criteria: uniform vertex distribution, avoiding overlap or near-overlap ,uniform edge length and reflect the inherent symmetry to show the relationship between vertices, hence, find the law between them and get more useful information.

2. The Principle of Force-directed Algorithms

A. Spring model

The thinking of spring model is that assume that each vertex is a small ball. For that they do not know the optimal coordinates of the ball, the ball randomly distributed in these two-dimensional plane (three-dimensional graphics can also use this method, but this paper mainly discusses the automatic generation of two-dimensional structure of the problem).there exists spring between balls. Ideal length of springs is proportional to the distance between balls .once the distance between balls is not equal to ideal length of springs, the spring in system will be stretched or compressed and the system will accumulate energy. If the energy of system is not zero, the balls will move because of the attractive and repulsive forces. Once the balls do not move any more, the energy of system is zero and the system is in an equilibrium state. At this point that the coordinate position of the ball is in the optimal position of the vertex structure.

Based on this thinking, we can build a mathematical model for a spring system:

1. $x(i)$ represents the coordinates of node i ;

2. $d(i,j)$ represents the ideal spring length between nodes i and j (the graph-theoretic distance between nodes i and j);

3. $\|x(i) - x(j)\|$ represents actual distance between nodes i and j .

Then the energy of spring between nodes i and j can be described like this:

$$\left(\|x(i) - x(j)\| - d(i, j)\right)^2 \quad (1)$$

As can be seen from the spring energy equation, if the distance between nodes i and j is equal to ideal length of springs, the energy of spring will be zero. If the distance between nodes i and j is longer than ideal length of spring (the spring is stretched), or the distance between nodes i and j is shorter than ideal length of spring (the spring is compressed), the energy of spring will be zero. The more differences between the actual distance between nodes and the idea length of spring, the greater the energy of the spring will be. So, speaking for the entire spring system, its energy is the sum of all the spring energy between all nodes, namely:

$$Energy = \sum_{i \neq j} \left(\|x(i) - x(j)\| - d(i, j)\right)^2 \quad (2)$$

In that way, drawing optimal structure is then equivalent to mathematical unconstrained optimization problem, means that find the coordinate $x(i)$ of all nodes, making the formula above has minimum energy. The coordinate $x(i)$ in this situation is the optimal solution of optimization problem above.

The force of each vertex can be calculated like this: If the distance between nodes i and j is longer than ideal length of spring (the spring is stretched), vertex i will be attracted by the spring:

$$AF(v) = \left(\|x(i) - x(j)\| - d(i, j)\right) * \left(\frac{x(j) - x(i)}{\|x(i) - x(j)\|}\right) \quad (3)$$

If the distance between nodes i and j is shorter than ideal length of spring (the spring is compressed), vertex i will be repulsed by the spring:

$$RF(v) = \left(\|x(i) - x(j)\| - d(i, j)\right) * \left(\frac{x(j) - x(i)}{\|x(i) - x(j)\|}\right) \quad (4)$$

In fact the two forces are the same, just the direction in opposite. So the force $F(i)$ of vertex i can be described like this:

$$F(i) = \sum_j \left(\|x(i) - x(j)\| - d(i, j)\right) * \left(\frac{x(j) - x(i)}{\|x(i) - x(j)\|}\right) \quad (5)$$

The iterative process of force-directed algorithms is:

Step1 assume initial step $t=1$

Step2 the vertex is randomly distributed in two-dimensional (or three-dimensional) space, the coordinate of vertex i is $x(i)$

Step3 calculate the total force $F(i)$ of each vertex i

Step4 calculate the position changes ($d x(i) = t * F(i)$) of each vertex i

Step5 if all the position changes $d x(i)$ are small, print x and stop

Step6 update x and t : $x(i) = x(i) + d x(i)$, $t = 0.9 * t$

Step7 go step3

The disadvantage of force-directed algorithm based on spring model is that it must calculate the graph-theoretic distance between any two vertices, If there are 1000 vertices, we will calculate the graph-theoretic distance about $1000 \times 999 \div 2 = 499500$ times, the time complexity is $O(V^2)$. While calculating the graph-theoretic distance is more time-consuming than calculating the geometric distance, thereby, computation of force-directed algorithm based on spring model will grow rapidly along with the increase of vertices in the graph.

B. Spring electrical model

In order to solve the problem that the spring model is time-consuming in calculating the graph-theoretic distance, the force-directed algorithm based on spring electrical model arises. The physical concepts -electronic repulsion is brought to graph-theoretic distance, It is not necessary to calculate the graph-theoretic distance between each vertex any more.

The thinking of spring electrical model is: assume the vertexes are balls which are with the same charge and repulse each other, the repulsion force is inversely proportional to the geometric distance of the ball; and if two vertexes are connected by an edge, the corresponding balls will be connected by a spring whose ideal length is zero, the spring makes ball attracted, repulsion forces are inversely proportional to the geometric distance of the ball. If the ball's repulsion force is equal to the attraction force, the energy of the whole system will be zero, and the system will in equilibrium states. Otherwise the energy of the whole system will not be zero, and the system will not in equilibrium states. The ball will not move under the guidance of attractive and repulsive forces until total forces is zero and the ball comes into balance, the energy of system is zero.

Let $x(i)$ represents the coordinates of the vertex i , $\|x(i) - x(j)\|$ represents the actual distance between vertexes i and j , then the electronic repulsion forces between any vertexes i and j is:

$$Re\ pulsive\ Force = -CK^2 / \|x(i) - x(j)\| \quad (6)$$

That is repulsive forces are inversely proportional to the distance: the closer the distance, the greater the repulsion force will be. The attraction force between vertexes i and j that are not connected by an edge is:

$$Attractive\ Force = \|x(i) - x(j)\|^2 / K \quad (7)$$

That is attraction forces are proportional to the square of the distance: the greater the distance, the greater the attraction.

If the ball's repulsion force is equal to the attraction force, the energy of the whole system will be zero, otherwise, the greater force of each vertex, the higher the energy of the entire system. In that way the optimal structure problem is then equivalent to minimum energy problem. Likewise, this optimization problem can also apply Iterative algorithm which is same in the before model to find the optimization.

In the spring model, only spring force a physical phenomenon is introduced, that is easy to understand and facilitates mathematical calculations, since calculating the graph-theoretic distance is more complex than calculating the geometric distance, so in the same iteration conditions, the algorithms based on spring electrical model are more efficient than the algorithms based on spring model.

3. Optimization for Force-directed Algorithms

Guidance algorithm in these papers the basic force iterative slow shortcomings, this paper puts forward and multi-layer method by the use of super node basic force guidance algorithm is improved.

A. Multilayer method

The thinking of multilayer method is putting the complexity into simplicity: If the original graph has tens of thousands of vertices, we can reduce it to an abstract illustration with thousands of points at first. Then simplify the once abstract illustration with thousands of vertices as secondary abstract illustration hundreds of vertices, and then come to the graph with dozens of points. Once the original graph is simplified as abstract illustration with dozens of points that we can deal with, then we can use common iterative forces-directed algorithms to get the optimal structure of this abstract illustration, then use the common iterative forces-directed algorithms again to get its booted drawings after reverse adjustment, upside reversal like this, finally we can get the optimal structure of large-scale map efficiently.

The multilayer method is mainly recursion and has following three steps:

Step1 simplify the graph

Step2 get the optimal structure

Step3 reverse adjustments and get the optimal structure of large-scale map.

There are many methods to simplify large-scale map into small-scale graphs which have similar structure, commonly used method is edge collapse method. We should first understand the concept of the biggest independent edges set.

The independent edges sets consist of edges which have no common vertices. The independent edge set will not be independent if put an edge which not in the set into the set, then this independent edge set becomes the biggest independent edges set.

For example, in figure 1(a), {1-2, 3-4, 5-6, 7-8} is the biggest independent edges set, because those edges (with bold lines indicate) not share vertices with each other, so it is

biggest independent edges set. And if put any edges into it, those edges will not be independent edges set any more.

Once there be a biggest independent edges set, we can treat the edges in the biggest independent edges set as a node .The figure 1(b) is the simplified graph by simplify the bold lines (biggest independent edges set) in the figure 1(a) as a node. In this way, we can simplify the graph with nine nodes and nine edges as a graph with five nodes and five edges.

We can get the optimal structure of fig.1(a) by apply iterative forces-directed algorithm to fig.1(b).It is observed that the two nodes which connected by the biggest independent edges set have the same the same coordinates, extend those nodes and apply iterative forces-directed algorithm again, then we can get the optimal structure fig.2 (b) of fig. 1(a).

For the large problems, one time simplification is not enough. It usually requires repeated simplification until the graph has dozens of vertices. Even the graph is simplified to just a few dozen vertices, the simplified graph still captures the basic structure of the original graph .so the global optimum drawing of the simplified graph after simplification also reflect the more complex graph and the global optimum drawing of original graph.

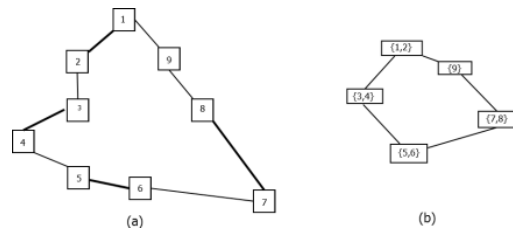


Fig. 1 (a) The Biggest Independent Edges Set
(b)The Figure after Simplifying the Edges in the Biggest Independent Edges Set

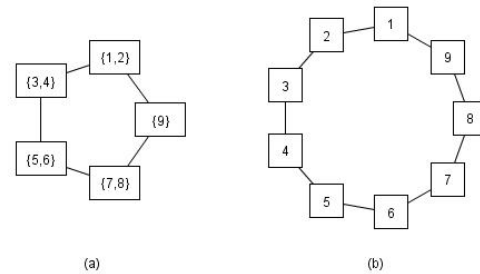


Fig .2 (a) The Optimal Structure of the Simplified
(b) The Optimal Structure of original figure

B. Method of hyper node

Whether spring model or spring electrical model both need to calculate the graph-theoretic distance or the geometric distance between any two nodes. If the graph has n vertices, it needs to calculate the distance about $n(n-1)/2$ times for each iterative. So for the large-scale map, the common iterative algorithms are too slow. To solve this problem is to use hyper

node. This approach is first used in kinetic and astrophysics. It only applies to electric spring model.

In electric spring model, the spring attraction forces only exist between the nodes which are connected by edges, the calculation is not large. The electronic repulsion forces exist between any two vertices, so if the graph has n vertices, it needs to calculate the repulsion forces about $n(n-1)/2$ times. If you want to calculate the repulsion forces between the vertex A and any other vertices, the thinking of hyper node is that: for the close vertices, the electronic repulsion forces can only be calculated one by one. However, for the group of vertex is too far away, as long as the size of the box contains all the vertices is smaller than the distance between vertex A and any other vertices, we can treat those vertices as a hyper node. The mass of the hyper node is the sum of the mass of all the vertices. This way maybe exists error, but it is not big.

4. The Algorithm Validation And Experimental Results

In this paper, we validate forces-directed algorithm which is optimized by connect the vertices which are in different types and sizes. In the drawing results, we do not use any graphics to represent vertices, the intersection of the edges is the position of the vertices, nor do we mark the number of vertices. Because in large-scale maps, marking the number of vertices and drawing the vertices graphics will not affect the display of the entire graphic. In order to keep the identity, the focus of the graphical results lies in showing the construct information between vertices and the rules it contains.

A. Experiment one Small-scale structure graph

Table I is unconnected graph ordered pairs with twelve vertices, table 2 is connected graph ordered pairs with twenty vertices, apply the forces-directed algorithm which is optimized to them for automatic drawing, figure 3 is the unconnected optimal structural renderings with twelve vertices, figure 4 is the connected optimal structural renderings with twenty vertices.

TABLE I Connected Ordered Pairs with Twelve Vertices

Number of pairs	Starting nodes	Ending nodes
1	1	2
2	1	7
3	2	3
.....
11	6	1
12	6	12

TABLE II Connected Ordered Pairs with Twenty Vertices

1-2 2-3 3-4 4-5 5-6 6-7 7-8 8-9 9-10 10-11
11-12 12-13 13-14 14-15 16-16 16-17 17-18 18-19 19-20 20-1
1-3 3-5 5-7 7-9 9-11 11-13 13-15 15-17 17-19 19-1

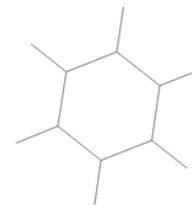


Fig.3 Optimal Structure with Twelve Vertices

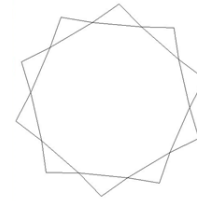


Fig.4 Optimal Structure with Twenty Vertices

Seeing from the experiment results of figure 3 that we can know, the forces-directed algorithm can meet the four pre-established criteria they are uniform vertex distribution, avoid overlap or near-overlap, uniform edge length and reflect the inherent symmetry in small-scale maps.

B. Experiment two medium-scale structures

Figure 5, 6 are two vertex connect ordered pairs in medium-scale. Figure 5 shows sixty vertex ordered pairs, it imitates the molecular structure of C60; figure 4-4 is a grid of 10x10 with one hundred vertices. Figure 7 is the result by applying iterative forces-directed algorithm to those connected ordered pairs for drawing.

1-2 2-3 3-4 4-5 5-6 6-1 6-7 7-8 8-9 9-10 10-6
10-11 11-12 12-4 1-13 13-14 14-7 3-15 15-16 16-17 17-2 9-18 18-19 19-20
20-8 17-21 21-22 22-13 23-15 23-24 24-12 20-25 25-26 26-14 26-27 27-28
28-22 18-29 29-30 30-11 30-31 31-32 32-24 31-33 33-34 34-29 34-35 35-36
36-19 36-37 37-38 38-25 38-39 39-27 21-40 40-41 41-28 41-42 42-43 43-39
43-44 44-45 45-53 53-48 49-54 54-50 54-55 55-56 56-52 58-55 33-60 56-60
58-53 60-57 57-59 59-35 57-58 59-46

Fig.5 Vertex Ordered Pairs of C60 with Sixty Vertices

1-2 2-3 3-4 4-5 5-6 6-7 7-8 8-9 9-10
11-12 12-13 13-14 14-15 15-16 16-17 17-18 18-19 19-20
21-22 22-23 23-24 24-25 25-26 26-27 27-28 28-29 29-30
31-32 32-33 33-34 34-35 35-36 36-37 37-38 38-39 39-40
41-42 42-43 43-44 44-45 45-46 46-47 47-48 48-49 49-50
51-52 52-53 53-54 54-55 55-56 56-57 57-58 58-59 59-60
61-62 62-63 63-64 64-65 65-66 66-67 67-68 68-69 69-70
71-72 72-73 73-74 74-75 75-76 76-77 77-78 78-79 79-80
81-82 82-83 83-84 84-85 85-86 86-87 87-88 88-89 89-90
91-92 92-93 93-94 94-95 95-96 96-97 97-98 98-99 99-100
1-11 2-12 3-13 4-14 5-15 6-16 7-17 8-18 9-19 10-20
11-21 12-22 13-23 14-24 15-25 16-26 17-27 18-28 19-29 20-30
21-31 22-32 23-33 24-34 25-35 26-36 27-37 28-38 29-39 30-40
31-41 32-42 33-43 34-44 35-45 36-46 37-47 38-48 39-49 40-50
41-51 42-52 43-53 44-54 45-55 46-56 47-57 48-58 49-59 50-60
51-61 52-62 53-63 54-64 55-65 56-66 57-67 58-68 59-69 60-70
61-71 62-72 63-73 64-74 65-75 66-76 67-77 68-78 69-79 70-80
71-81 72-82 73-83 74-84 75-85 76-86 77-87 78-88 79-89 80-90
81-91 82-92 83-93 84-94 85-95 86-96 87-97 88-98 89-99 90-100

Fig.6 Vertex Ordered Pairs of C60 with One Hundred Vertices

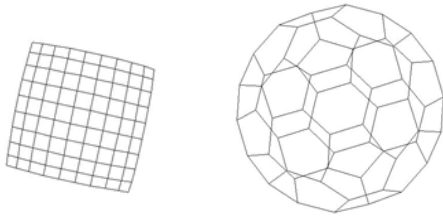


Fig.7 Left the Result of Figure 5; Right the Result of Fig. 6

Seeing from the left of figure 7 we can know, the essence of forces-directed algorithm is that the guided graph turn into optimal structure under the effect of force, then the connected graph always tend to be a rounded or spherical under the effect of force ,the left of fig.7display the a three-dimensional spatial effects of C60. If you are careful enough, you will find the size of the grid in the right of figure 8 is not the same. This due to the fact that when we calculate the repulsive forces between vertexes which are from each other, we use the hyper nodes to get approximate solutions ,the repulsive forces between distant nodes are not big enough to support the size of grid to be same with centers.

C. Experiment three large-scale structures

In the experiment three, we use the ordered pairs with 900 vertexes to draw a garland whose size is 30×30 , the text of the ordered pairs in this experiment is relatively large, we write the ordered pairs with 900 vertexes in the text by a simple small applet, so we no longer paste the text screenshot here. Figure 8 is the optimal structure of the garland whose size is 30×30 .

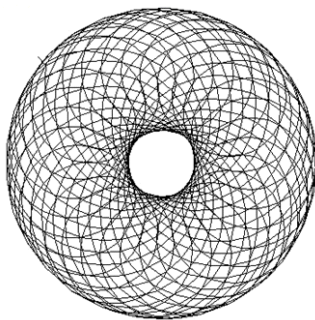


Fig.8. A Garland Whose Size is 30×30

However, because of different purposes, different standards, the optimal structure of same graph are not unique. The optimal structure of some demands may not meet mechanical condition .In the physical, such a structure cannot be obtained by means of forces-directed algorithm. Such as a tree structure, equilateral graphics, etc. Since the role of mechanics, the whole graph may tend to be a circular structure under the force of forces-directed algorithm; it

performs excellently in the undirected connected graph. The forces-directed algorithm is not only applied to two-dimensional space, in three-dimensional space can also be a good application. For this reason, some graphs may come to pseudo three-dimensional effect under the force of forces-directed algorithm. Sometime it is less intuitive than other plane algorithm.

Those deficiency are caused by the characteristics of forces-directed algorithm itself, we cannot rely on a particular algorithm to solve all problems, but we can improve the operational efficiency through dynamic optimization of the parameters.

5. Conclusions

If we just use the normal forces-directed algorithm, we can only get the optimal structure of the graph with 20 vertexes, when the vertex increases, vertex coordinates may be trap into local optimum, instead, we can get the optimal structure of the graph with 1000 vertexes if we use multilayer method, but running speed will be slower. On this basis, we introduce the Octree data structure of hyper node; this may improve the operational efficiency greatly. The automatic drawing experiment of ordered pairs in small-scale ,medium-scale, large-scale structure test the effectiveness of multilayer method and hyper node ,reached the pre-established four drawing standards, greatly improve the speed performance of the algorithm, Fully express automatic drawing intentions of the people's .

References

- [1] Fruchterman, Thomas M. J.; Reingold, Edward M. (1991),"Graph Drawing by Force-Directed Placement", Software – Practice & Experience (Wiley) 21 (11): 1129–1164.
- [2] Huang Jingwei, Kang Lishan, Chen Liuping. A new undirected graph drawing algorithm. Journal of Software. 2000, 11(1):138~142.
- [3] ZhangQin, ZhuLi, Xia ZhaoJun. The drawing for work flow diagrams based on genetic algorithms. Computer and digital engineering. 2010 No.4 PP125-127.
- [4] Chernobelskiy R, Cunningham K I, Goodrich M T, et al. Force-Directed lombardi-style graph drawing //Graph Drawing. Springer Berlin Heidelberg, 2012: 320-331.
- [5] Kobourov, Stephen G. Embedders and Force-Directed Graph Drawing Algorithms.2012 Spring.
- [6] Bannister M J, Eppstein D, Goodrich M T, et al. Force-Directed graph drawing using social gravity and scaling //Graph Drawing. Springer Berlin Heidelberg, 2013: 414-425.
- [7] Hu YiFan. Optimal Structure——find new regular from the complex relationship. Shanghai: Shanghai Science Press, 2007.12.
- [8] SunQiang. The force- guided scheduling algorithm of peak power optimization. Journal of University of Electronic Science and Technology.2010, 39(1):302-1.
- [9] LiXin. The algorithm research for solving unconstrained optimization problems. Xi'an University of Electronic Science and Technology, 2009.
- [10] XinHua. The algorithm research for automatic mapping of distribution network line diagram. Yunnan Power Technology Forum Proceedings in 2010(L), 2010.