Drawing Large Weighted Graphs using Clustered Force-Directed Algorithm

Jie Hua[1], Mao Lin Huang [2,1*]and Quang Vinh Nguyen[3]
[1]Faculty of Engineering and IT, University of Technology, Sydney, Australia
[2]School of Computer Software, Tianjin University, China
[3]MARCS Institute & School of Computing, Engineering & Mathematics, University of Western Sydney
*{*Mao.Huang@uts.edu.au*}

Abstract - Clustered graph drawing is widely considered as a good method to overcome the scalability problem when visualizing large (or huge) graphs. Force-directed algorithm is a popular approach for laying graphs yet small to medium size datasets due to its slow convergence time. This paper proposes a new method which combines clustering and a force-directed algorithm, to reduce the computational complexity and time. It works by dividing a Long Convergence: LC into two Short Convergences: SC1, SC2, where SC1+SC2 < LC. We also apply our work on weighted graphs. Our experiments show that the new method improves the aesthetics in graph visualization by providing clearer views for connectivity and edge weights.

Keywords--- **graph visualization; graph drawing; data analytics; information visualization; force-directed graph drawing; clustered graph drawing; weighted graph**.

I. INTRODUCTION

Graphs generated in real-world applications could be very large with thousands or perhaps millions of nodes, such as academic citation and collaboration networks and the World Wide Web (WWW). On the other hand, in real world, the relationships among data elements in many cases are not only just "connection", rather they have some domain specific attributes (or 'weight') need also to be graphically represented in the graph, such as costs, lengths or capacities, etc. Those graphs are called weighted graphs if a number (weight) is assigned to each edge. [1]

As the result of rapid increasing of the size in networks, how to draw a large graph with clear representations of data and its network structures is the challenge to the graph drawing community. The key issue here is not only to provide users with a comprehensive display of large graphs on the screen, but also a user-friendly navigable visual structure for users browsing through the structure to find a particular detail of the data. In the past, some attempts to overcome this problem have proceeded in two main directions:

- Clustering: Groups of related nodes are clustered into super-nodes. The user sees a summary of the graph with the super-nodes (clusters) and super-edges between the super-nodes (clusters). [2, 3, 4, 5]. E.g. K-mean clustering method, Markov Clustering method.
- Navigation: The user sees only a small subset of the nodes and edges at any one time, and facilities are provided to navigate through the graph [6, 7, 8, 9, 10, 11, 12].

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem [2, 13]. There are many different clustering methods and more commonly those algorithms on based on the following arrangements:

- Connectivity based clustering (hierarchical clustering)
- Centroid-based clustering (k-means clustering)
- Distribution-based clustering
- Density-based clustering

In practice, applying different clustering algorithms to the same clustered graphs might create very different final layouts. Force-directed layout algorithms use a physical analogy to draw graphs. A graph is viewed as a system of bodies with forces acting between the bodies. The algorithm seeks a configuration of the bodies with locally minimal energy, that is, a position for each body, such that the sum of forces on each body is zero. And the method is easy to understand, the results is normally good [14, 15, 16, 3, 17, 18].

However, force-directed methods can deal with only a limited number of nodes due to its slow convergence time. In this paper, we propose a new approach which

1550-6037/14 $31.00 © 2014 IEEE
DOI 10.1109/IV.2014.24

combines clustering method and the traditional force-directed algorithm, to speed up the convergence time by dividing a Long Convergence: LC into two Short Convergences : SC1, SC2, where SC1+SC2 < LC. Thus, we could greatly reduce the computation complexity (or convergence time) of force-directed graph drawing methods.

We also apply proposed method to drawing weighted graphs. The early outcome of our approach indicates improvement in computation time and graph aesthetics that have a clearer view of the properties associated with the weighted graph in terms of its connectivity and edge weights. The preliminarily experimental results also shown that the combination of the clustered graph drawing method and the force-directed layout algorithm could be used in large graph drawing.

II. METHODS

In our experiments, we used clustering method based on edge weight to group vertices for pre-handling; then applied forces within each cluster. Details are described in the following subchapters.

A. Decrease Progressively Clustering on Weighted Graph (DPCW)

The *DPCW* clustering is based on the connectivity of vertices and weight on each edge in the graph. The basic idea is that if a vertex $v_i$is assigned in a cluster c_j, then we intend to include all its connected vertices with the most weights in the graph in this cluster.

Suppose that $W = (w_0, w_2, ..., w_k)$ is the set of weights on every edge, w_k is the maximum weight, and w_0 is the minimum weight in W.

Assume that $G = (V, E)$ is a connected undirected weighted graph, where$V$ is the set of vertices and E is the set of edges among V. A cluster graph $C = (G', T)$ consists of graph $G'= (V', E')$ and a rooted tree T, where G' is a sub-graph of G. The DPCW algorithm can be described below:

a) If $(v_m, v_n) \in V$, where $e_i = (v_m, v_n)$ and its weight $w_i = w_k$, then we add two vertices v_m and $v_n$into the same cluster c_k^1;
b) If $(v_{m1}, v_{n1}) \in V$, where $e_{i1} = (v_{m1}, v_{n1})$ and its weight $w_{i1} = w_k$.
   1) If ($m = m_1$and $n \neq n_1$), then we add vertex v_{n1}into the cluster c_k^1;
   2) If ($m \neq m_1$and $n = n_1$), then we add vertex v_{m1}into the cluster c_k^1;
   3) If ($m = n_1$and $n \neq m_1$), then we add vertexv_{m1}into the cluster c_k^1;
   4) If ($m \neq n_1$and $n = m_1$), then we add vertex v_{m1}into the cluster c_k^1;
   5) If ($m \neq m_1$and $n \neq n_1$and$m \neq n_1$and $n \neq m_1$), then we add two vertices v_{m1} and v_{n1}into the same cluster c_k^2;
c) Repeat step (b) until every vertex satisfies the conditions described in (b) are included in clusters, and the cluster $c_k = \{c_k^1, c_k^2, ..., c_k^{xk}\}$;
d) Find the smaller weight $w_{k-1} \in W$, where $w_{k-1} < w_k$ and $w_{k-1} > \{w_0, w_2, ..., w_{k-2}\}$, set $w_k = w_{k-1}$, Repeat step (b) and (c) until every vertex satisfies the conditions described in (b) are included in clusters, and the cluster $c_{k-1} = \{c_{k-1}^1, ..., c_{k-1}^{x(k-1)}\}$;
e) Repeat step (d) until $w_i = w_0$and every weight in W has been handled;
f) The final clusters $C = \{c_k^1, \ldots c_k^x, \ldots, c_{k-1}^1, \ldots c_{k-1}^{x(k-1)}, \ldots, c_0^1, \ldots c_0^{x0}\}$.

B. Markov Cluster Algorithm (MCL)

The *MCL* is based on the Markov Chain method which calculates the random walkers' chance between every pair of nodes in the graph, and then the nodes could be grouped according to the connection possibilities among them [19].

The *MCL* adds the inflation operator for both strengthening and weakening of current [19] (Strengthens strong currents, and weakens already weak currents). The details are:

a) Input is an un-directed graph, power parameter e, and inflation parameter r.
b) Create the associated matrix
c) Add self-loops to each node (optional)
d) Normalize the matrix
e) Expand by taking the eth power of the matrix
f) Inflate by taking inflation of the resulting matrix with parameter r
g) Repeat steps e) and f) until a steady state is reached (convergence).
h) Interpret resulting matrix to discover clusters.

C. A Classical Force-Directed Algorithm

The force-directed algorithm aims to position nodes with as few crossing edges as possible by assigning forces among the set of nodes and edges for drawing graphs in an aesthetically pleasing way. The spring forces are used to keep all elements in reasonable distances: not too close and not too far.

The force-directed algorithms achieve this by assigning forces amongst the set of edges and the set of nodes. The entire graph is then simulated as if it were a physical system. In the force-directed algorithm, we need to calculate all the forces work on every element, and then place them to suitable position to avoid edge crossings. There are three steps for each iterative calculation.

a) Calculate the effect of attractive forces $f_a(d)=d^2/k$ between adjacent vertices;
b) Calculate the effect of repulsive forces $f_r(d)=-k^2/d$ between all pairs of vertices;
c) Finally stop the iteration if f_a and f_r tend to not be changed.

Where d is the distance between two vertices and k is the optimal distance between vertices.

III. OUR APPROACH

This section describes the details of our approach as:

a. The first step is to apply the force-directed algorithm on a given graph $G = (V, E)$;
b. Apply two clustering methods *DPCW* and *MCL* separately to build a clustering structure on the given graph G;
c. Apply the force-directed algorithm on clustered graph $C(G) = (G', T)$ with all its clusters 'close' as red dots in the layout till it's (force-directed drawing) convergence process completed and reaches the energy balance;
d. 'Open' all its clusters in the layout of $C(G)$;
e. Apply the forces on the elements within same clusters separately again to achieve the energy balance;
f. In the final step, compare the qualities of final layouts (edge crossing).

IV. CASE STUDY

A case is given to explain the relationship between final layout and edge weight, the data is based on a real company structure and dummy email communication amounts which can be found in the Table 3.

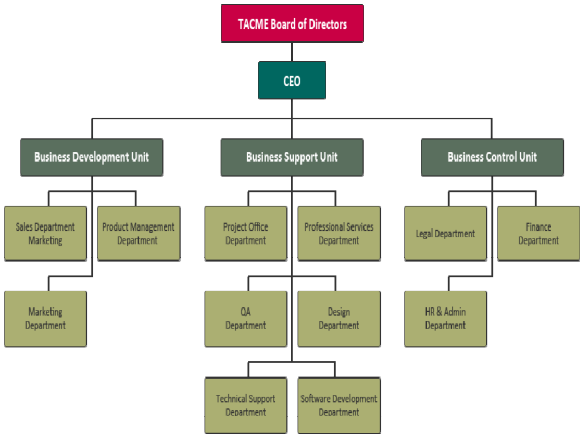

Figure 1. Structure of a company
(Sourced from http://www.tacme.com/corporate_structure.html)

Weighting Description:	
Quantity	**Weighting**
<10	1
11 – 50	2
51 – 100	3
101 – 200	4
201 – 300	5
301 – 400	6
> 401	7

Table 2. Email weighting description

ID	Name	Position
0	James	Director
1	David	Director
2	George	CEO
3	Ronald	Business Development Manager
4	John	Business Support Manager
5	Richard	Business Control Manager
6	Daniel	Sales Department Leader
7	Kenneth	Product Department Leader
8	Anthony	Marketing Department Leader
9	Robert	Project Office Leader
10	Charles	Professional Service Leader
11	Paul	QA Leader
12	Mark	Design Office Leader
13	Kevin	Technical Support Office Leader
14	Edward	Software Development Leader
15	Joseph	Legal Office Leader
16	Michael	Finance Office Leader
17	Jason	HR Office Leader

Table 2. HR details

Email Amount							
ID	**Emails /pm**	**Weighting**	**ID**	**ID**	**Emails /pm**	**Weighting**	**ID**
0	**5**	**1**	1	**9**	**546**	**7**	12
0	**6**	**1**	2	**9**	**23**	**2**	13
1	**5**	**1**	2	**9**	**145**	**4**	14
2	**25**	**2**	3	**10**	**256**	**5**	11
2	**36**	**2**	4	**10**	**222**	**5**	12
2	**53**	**3**	5	**10**	**190**	**4**	13
3	**150**	**4**	6	**10**	**56**	**3**	14
3	**213**	**5**	7	**11**	**78**	**3**	12
3	**298**	**5**	8	**11**	**112**	**4**	13
4	**345**	**6**	9	**12**	**98**	**3**	14
4	**123**	**4**	10	**15**	**88**	**3**	16
4	**212**	**5**	11	**15**	**128**	**4**	17
4	**453**	**7**	12	**16**	**238**	**5**	17
4	**156**	**4**	13	**17**	**5**	**1**	7
4	**278**	**5**	14	**16**	**15**	**2**	6
5	**300**	**5**	15	**16**	**23**	**2**	7
5	**78**	**3**	16	**16**	**54**	**3**	8
5	**256**	**5**	17	**16**	**18**	**2**	9
6	**78**	**3**	7	**16**	**23**	**2**	11
6	**145**	**4**	8	**16**	**41**	**2**	13
7	**139**	**4**	8	**16**	**13**	**2**	14
9	**34**	**2**	10	**16**	**27**	**2**	10
9	**134**	**4**	11	**9**	**546**	**7**	12

Table 3. Dummy email amount

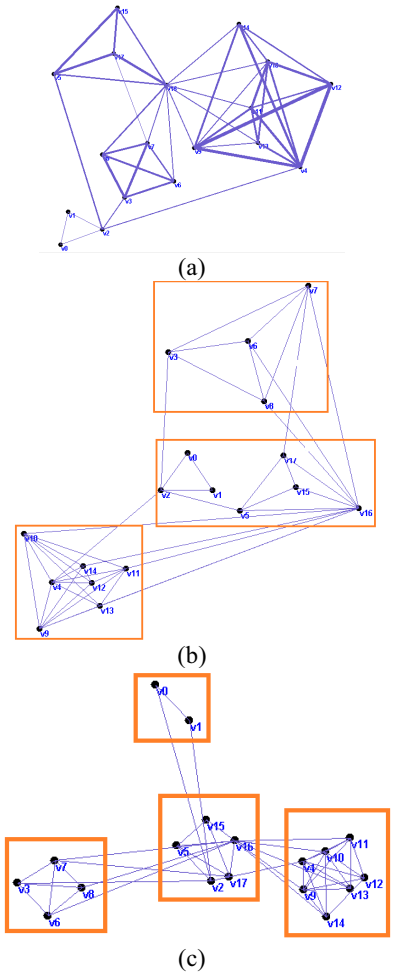

Figure 2 Example of final layout comparison on weighted graph Gm
(a) Initial graph G_m; (b) Final layout based on *MCL*; (c) Final layout based on *DPCW*.

In Figure 2 above, every orange rectangle indicates a cluster.
In Figure 2 (b), the clusters are $\{v_4, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}\}$, $\{v_0, v_1, v_2, v_5, v_{15}, v_{16}, v_{17}\}$ and $\{v_3, v_6, v_7, v_8\}$;
In Figure 2 (c), the clusters are $\{v_0, v_1\}$, $\{v_4, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}\}$, $\{v_2, v_5, v_{15}, v_{16}, v_{17}\}$ and $\{v_3, v_6, v_7, v_8\}$;
There is a slight difference between (b) and (c), but they all provided a reasonable layout based on the email communication amounts. And since the time complexity of *DPCW* is much better than *MCL,* it can reduce computational time significantly, the experimental result on running time is given in Figure 3 shown below.

V. EXPERIMENTAL EVALUATION

We created artificial 20 connected / undirected weighted graphs randomly range from 100 to 290 vertices, and from 272 to 1010 edges for the evaluations. We applied the *MCL* /*DPCW* and forces on each graph, compared the qualities of the final layouts (edge crossing). See Figure 3 for the comparison results on time complexity and edge crossing of our experiments.

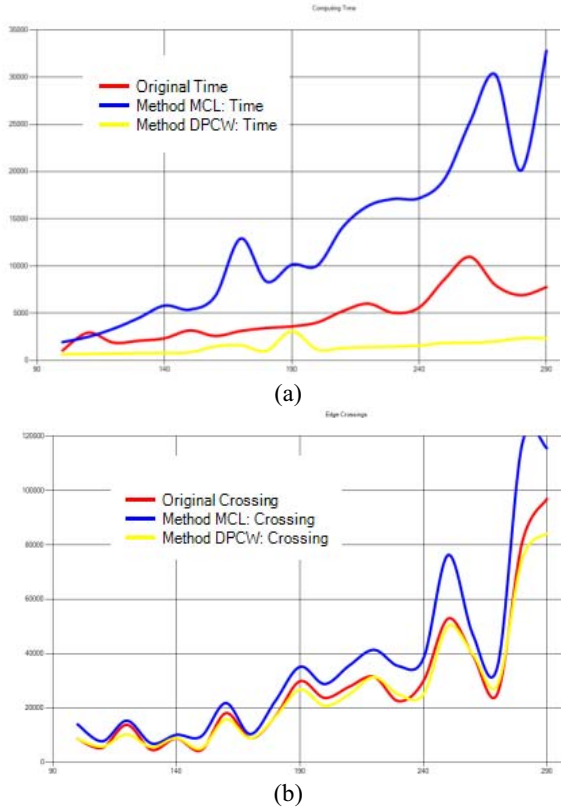

Figure 3.Thecomparisonofthefinal layouts on experiments
(a) Computing time; (b) Edge crossing.

From the results of the Figure 3 above we can see that the *DPCW* method provides layouts with a similar number of edge crossings to the traditional force-directed algorithm (see Figure 3(a)) yet much less computational time (see Figure 3(b)). DPCW also has much better than outputs the *MCL* method. However, the reduction in edge crossings is not significant, although the structures are based on logical values.

VI. CONCLUSIONS AND FUTURE WORKS

We have presented a new approach for potentially visualizing large graphs by combining clustering and force-directed algorithms. Our method works by dividing long convergences into short convergences so that the computational time can be reduced. Early experimental evaluations demonstrate its effectiveness in terms of reduction in computational time and some aesthetical improvement in the graph layouts. In our future works, we will apply our approaches to a wider and larger datasets and applications. A usability study will also be carried out to formally evaluate the effectiveness of the methods.

REFERENCES

[1] Fletcher, P., Patty, C.W. & Hoyle, H. 1990. Foundations of Discrete Mathematics. 1st ed. USA: Pws Pub Co.

[2] Wikipedia, P.2013, Cluster analysis, Syd, viewed 24 Feb. 2013, <http://en.wikipedia.org/wiki/Cluster_analysis>.

[3] Omote, H. & Sugiyama, K. 2007, Force-Directed Drawing Method for Intersecting Clustered Graphs, APVIS 2007, 6th International Asia-Pacific Symposium on Visualization 2007, 5-7 February 2007, Sydney, Australia, pp.85-92

[4] Huang, X. & Lai, W., 2005, Clustering graphs for visualization via node similarities. J. Vis. Lang. Comput. 17, 3 (June 2006), 225-253. Elsevier Ltd.

[5] Huang, M., Nguyen, Q. V., A space efficient clustered visualization of large graphs Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, 920-927

[6] Huang, M.L., & Nguyen, Q.V. 2007, Navigating Large Clustered Graphs with Triple-Layer Display, 11th International Conference Information Visualization, 2-6 July 2007, Zürich, Switzerland, pp. 684-692.

[7] Sarkar, M. & Brown, M.H. 1994, 'Graphical fisheye views', Communications of the ACM, Volume 37 Issue 12, pp. 73 – 83.

[8] Huang, M.L., Eades, P. & Wang, J. 1998, On-line animated visualization of huge graphs using a modified spring algorithm, Journal of Visual Language and Computing, no. vl980093, pp. 623-645.

[9] Huang, M.L., Eades, P. & Cohen, R.F. 1998, Webofdav – navigating and visualizing the web on-line with animated context swapping, Computer Networks and ISDN Systems, 30(1), pp. 638-42.

[10] Nguyen, Q.V. & Huang, M.L. 2005: EncCon: an approach to constructing interactive visualization of large hierarchical data. Information Visualization 4(1): 1-21.

[11] Qiu, M., Zhang, K., Huang, M., 2006, Usability in mobile interface browsing, Web Intelligence and Agent Systems, Vol.4 (1), Pages 43-59, IOS Press.

[12] Nguyen, Q. V. and Huang, M. L., A focus+ context visualization technique using semi-transparency, The Fourth International Conference on Computer and Information Technology, 2004. CIT'04, 101-108.

[13] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D., On Modularity Clustering, Knowledge and Data Engineering, IEEE Transactions on, pp. 172 - 188 Volume: 20, Issue: 2, Feb. 2008.

[14] Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. 1999, Graph drawing algorithms for the viisualization of graphs, Prentice-Hall, New Jersey, USA.

[15] Eades, P., A heuristic for graph drawing. Congress Numerantium, 42:149-160, 1984.

[16] Lin, C.C., Yen, H.C. 2005, A New Force-Directed Graph Drawing Method Based on Edge-Edge Repulsion, Ninth International Conference on Information Visualisation,
6-8 July 2005, London, UK, pp.329-324

[17] Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. 1999, Graph drawing algorithms for the visualization of graphs, Prentice-Hall, New Jersey, USA.

[18] Lin, C.C. & Yen, H.C. 2008, A new force-directed graph drawing based on edge-edge repulsion, 9th International Conference on Information Visualization IV2008, 6-8July, London, England, pp. 329-34.

[19] Van Dongen, S. 2000, 'Graph Clustering by Flow Simulation', PhD Thesis, University of Utrecht, The Netherlands.

[20] Hua, J. & Huang M.L. (2013). Improving the Quality of Clustered Graph Drawing through a Dummy Element Approach. In Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference. Macau, 6-8 Aug. pp 88-92.