Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 1, January 2018, pp. 67-76
DOI: 10.23940/ijpe.18.01.p8.6776

Graph Layout Performance Comparisons of Force-Directed Algorithms

Jie Hua[a,*], Mao Lin Huang[a], Guohua Wang[b]

[a]University of Technology Sydney, Sydney, 2007, Australia
[b]South China University of Technology, Guangzhou, 510641, China

Abstract

Due to force-directed algorithms' capabilities of producing aesthetically pleasing graph layouts, which follow metrics for graph drawing aesthetics, these layouts have become the most common methods in the practical data visualization area. However, evaluating the performance of relevant algorithms remains a challenge, since graph layout quality is largely relying on aspects such as human intuition, personal judgment and methods' pre-setting parameters. In addition, most aesthetics criteria of graph drawing conflict with each other. This study evaluated the performance measurements of four force-directed algorithms in terms of seven commonly applied aesthetic criteria based on practical raw data collected, and demonstrated the experimental framework. The early outcomes compared twenty final graph layouts and gave empirical evidences; the study may assist with future detailed force-directed algorithms selection based on users' specific requirements.

Keywords: graph drawing; graph layout; force-directed algorithms; aesthetic criteria; evaluation; performance comparison

(Submitted on October 14, 2017; Revised on December 5, 2017; Accepted on December 20, 2017)

1. Introduction

Traditional force-directed algorithms have been widely adopted for data analytics purposes in the graph drawing field, they treat graphs as physical systems, relevant forces such as spring force, gravity force etc. are applied onto each element (vertex/edge) to keep them in reasonable distances, and the process stops when zero/minimized energy is reached, or a stop condition is determined [3,4,8,9,12,17,18,22]. The concept was first introduced by Tutte based on barycentric representations. The essential spring layout methods which rely on spring forces are similar to those in Hooke's law, there are repulsive forces among all vertices and also attractive forces among adjacent vertices. Alternatively, spring forces can be computed based on their graph theoretic distances. Graphs drawn with these force-directed algorithms tend to be aesthetically pleasing, exhibit symmetries, and most likely to produce crossing-free layouts for planar graphs, although comes with high running time due to their big *O* notations [19].

One feature of the force-directed algorithms is that they may generate different layouts of the same dataset, depending on the combination of input parameters such as initial placement of vertex's and constants that define the physical forces schedule [19]. All these lead to challenges of how to compare graph layout performances among those common applied force-directed algorithms, and how to judge the final layout performance of particular graphs.

Four general-purpose graph drawing methods (Bend-Stretch, Giotto, Column and Pair) were compared by Battista et al., and factors such as edge crossing/length/bends etc. were evaluated. 11,582 graphs with 10 to 100 vertices were generated based on raw data from practical software engineering and database applications. Experiments provided a detailed quantitative evaluation of those four algorithms' performances, as well as the trade-offs between aesthetic criteria and running time, but force-directed approach was not involved in detailed experiments [5].

* Corresponding author.
E-mail address: Jie.hua@alumni.uts.edu.au

Gansner and North presented the application of two post-processing techniques to produce uncluttered layouts with non-point nodes, to improve readability in labelled nodes for force-directed models since when nodes must be labelled. Point vertices are replaced by non-point vertices; simple force-directed methods normally produce unreadable graphs. The method prevents unwanted intersections between node-node and node-edge, but keeps edge-edge intersections, and the comparisons were limited between the Voronoi and scaling though [10].

Brandenburg, Himsolt and Rohrer compared five force-directed algorithms (Fruchterman and Reingold, Kamada and Kawai, Davidson and Harel, Tunkelang, Ludwig and Meldau) with a wide-ranged dataset for general undirected graphs, following measurements factors such as running time, edge length ratio/standard deviation and edge crossing etc. Uniform edge length and node distribution were generated as well. However, less empirical evidence was concluded, and the experimental results could not have been recalled and checked due to the random character of the algorithms [2].

Hachul and Jünger investigated several methods such as GVA (grid-variant algorithm), HDE (High-Dimensional Embedding) and FM^3 etc. for large graph drawing on a big variety of artificial and real-world graphs, addressing the time complexity. However, criteria for graph layout performance judgment were ‘pleasing’ features with no empirical evidence [14].

Huang, Eades, Hong, and Lin argued that effectiveness could be improved when algorithms were designed by making compromises between seven aesthetics, rather than trying to satisfy one or two of them to the fullest. Their study was tested on two methods (classical spring algorithm and BIGANGLE), and outcomes indicated that BIGANGLE, which produces drawings with multiple aesthetics being improved at the same time, induces significantly better performance of humans in perceiving shortest paths between two nodes [15].

The performance comparisons of two different force-directed algorithms (classical spring algorithm and forceAR) in terms of six commonly applied aesthetic criteria were evaluated by Huang and Lin. The results indicated that not only the aesthetics that were supposed to be improved by the algorithm, but also other aesthetics that were important to the overall visual quality of drawings [13].

However, performance measurement on final layouts from most methods above was either mainly judged based on human intuition, personal judgement, and largely relying on methods’ input parameters such as initial layouts etc. [13,16]. Graphs tested in experiments were mostly generated randomly; the data did not reflect real world. In addition, there are conflicts between most of the aesthetics criteria related to the graph layout quality; thus, conducting evaluations on algorithm performance measurement has become a challenging issue in the graph drawing area, and the task of validating graph drawing aesthetics is highly important.

In this paper, a case study was conducted based on practical raw data collected from the Australian stock market, to compare four force-directed approaches, in terms of seven aesthetic criteria in graph drawing (time complexity was not involved), and the specific purposes of the study were as follows:

- Compare final graph layouts with multiple aesthetic criteria
- Further explore the affections among each other of the multiple aesthetic criteria
- Examine different algorithms’ advantages

2. Method

In this section, experimental methodology and related algorithms are introduced, as well as the experimental hypotheses. From the technique points, this study involves three aspects as follows:

- Algorithm description
- Graph models
- Aesthetic criteria for graph layouts

2.1. Force-directed Algorithms

Four algorithms were chosen in the evaluation framework:

- FR (Fruchterman and Reingold)

  The Fruchterman and Reingold algorithm is a traditional force-directed layout algorithm, which is a modification of the spring embedder model of Eades (Eades, 1984). It attempts to produce aesthetically pleasing, two-dimensional pictures of graphs by conducting simplified simulations of physical systems. In this method, the vertices are represented by steel rings and the edges are treated as springs, the algorithm considers a force between every two vertices, and forces continue changing through moving the nodes till the energy is minimized, which is the termination state of the whole 'mechanic system'. Although a constant that determines how far a node moves in a single step needs to be pre-defined, the final layout may not reach equilibrium at all. FR is simple, elegant, conceptually intuitive, efficient, and comes with uniform edge length and reasonable processing time [9]. It follows two principles mentioned below.

1) Vertices connected by an edge should be drawn near each other.
2) Vertices should not be drawn too close to each other.

In this algorithm, suppose f_a and f_r are the attractive and repulsive forces respectively, with d the distance between the two vertices and k is the radius of the empty area around a vertex, then

$$f_a(d) = d^2 / k \qquad (1)$$

$$f_r(d) = -k^2 /d \qquad (2)$$

Given a graph $G = (V, E)$, the combined force applied on vertex v is:

$$F(v) = \sum\nolimits_{(u,v)\in E} f_{a,\ uv} + \sum\nolimits_{(u,v)\in V*V} f_{r,\ uv} \qquad (3)$$

- FA (ForceAtlas);
The FA layout algorithm (Bastian, Heymann & Jacomy, 2009) is a spatial layout method under the category of force-directed algorithms, which aims at giving a readable shape to a network (spatialization), along with integration between different techniques such as Barnes Hut simulation, degree-dependent repulsive force, local and global adaptive temperatures. It is produced for real-world networks, such as web networks etc. It addresses providing a generic and intuitive way to spatialize networks. FA layout algorithm gives more weight to the quality of the layout than the speed with which it has been computed [18]. This is especially true in the case of large networks. FA algorithm offers real-time settings including speed, gravity, repulsion, auto-stabilize, inertia or size-adjust etc. [6]. FA's force model is similar to FA2's as described below.

- FA2 (ForceAtlas2);
FA2 is based on FA but offers more options and innovative optimizations that make it a very fast layout algorithm. Its implementation of adaptive local and global speeds brings good performances for a network of fewer than 100,000 nodes. It was empirically observed that FA2 was at its best with strongly clustered networks [18].

The attraction force f_a between two connected nodes n_1 and n_2 depends linearly on the distance $d(n_1, n_2)$; the repulsive force Fr is proportional to the produce of the degrees plus one $(deg+1)$ of the two nodes n_1 and n_2.

$$f_a(n_1, n_2) = d(n_1,n_2) \qquad (4)$$

$$f_r(n_1, n_2) = k_r \frac{(deg(n_1)+1)(deg(n_2)+1)}{d(n_1,n_2)} \qquad (5)$$

Given a graph $G = (V, E)$, the combined force applied on vertex v is:

$$F(v) = \sum\nolimits_{(u,v)\in E} f_{a,\ uv} + \sum\nolimits_{(u,v)\in V*V} f_{r,\ uv} \qquad (6)$$

- LinLog.
Noack proposed the LinLog energy models, which can naturally represent the cluster structure of graphs by grouping densely connected nodes and separating sparsely connected nodes. The models include node-repulsion LinLog and edge-repulsion LinLog, whose minimum energy layouts reflect the cluster structure of graphs with respect to two well-defined clustering criteria, and edge repulsion in energy models. LinLog avoids or reduces the bias towards grouping nodes with the high degree when used instead of or in addition to node repulsion [22].

Given a graph $G = (V, E)$, for a drawing p and two nodes $u, v \in V$ the length of the difference vector $p(u) - p(v)$ is called the Euclidean distance of u and v in p and denoted by $||p(u) - p(v)||$ [21].

$$U_{LinLog}(p) = \sum_{(u,v)\in E} ||p(u) - p(v)|| \quad + \sum_{(u,v)\in V*V} ln||p(u) - p(v)|| \tag{7}$$

Noack defines the energy model ((attraction, repulsion)-model) of a layout as the exponent taken by distance in the formulas used to calculate attraction and repulsion (the log is considered as the 0th power) [22]. The (attraction, repulsion)-model of FA (1, −1) has an intermediate position between Noack's LinLog (0, −1) and the algorithm of Fruchterman and Rheingold (FR) (2, −1). ForceAtlas2 (FA2) is based on FA but offers more options and innovative optimizations that make it a very fast layout algorithm. FA2's ability to show clusters is better than FR's but worse than LinLog's [18].

2.2. Graph Models

In experimental evaluation:

- All raw data were collected from the ASX (Australian Securities Exchange), including 5,088 stocks in the Australia stock market, which ranged from 02/01/1997 to 30/06/2017. Around 6.4 million data entries were included (before processed).
- After data processing step, cross-comparison of related changing rates on every same two trading days between every two stocks was calculated, and nearly 194 million raw data were generated. See [23] for details.
- Five groups of final graph models were generated for further experiments, which followed the GraphML (Graph Markup Language) format, an XML-based format for the description of graph structures designed to improve tool interoperability and reduce communication overhead [7].

In this study, all experiments were based on real raw data collected, and initial layouts for graph models were produced randomly.

2.3. Aesthetic criteria

In the graph drawing area, judgement of the quality of a drawing is subjective. Battista, Eades, Tamassia and Tollis claimed that there were several common aesthetics for general undirected graph drawing, which includes: symmetry display; edge crossing reduction; edge bends reduction, edge length uniform and vertex distribution uniform etc. [3].

Finkel and Tamassia compared angular resolution and edge separation as relevant aesthetic criteria. The angular resolution refers to the angles formed by pairs of edges incident on a vertex. The edge separation refers to the distance between an edge and another non-incident, non-intersecting edge [8].

Huang indicated that human graph reading performance could be affected by the size of crossing angles. Maximizing the size of crossing angles had been shown to be beneficial for graph comprehension [16]. Average size of crossing angles (angle size), standard deviation of crossing angle (angle dev.) etc. were adopted in experiments, see [13], [15] and [16].

In this study, the following aesthetic criteria were applied to graph layout performance measurement:

- Cross#: Edge Crossing Number.
  Reducing the number of edge crossings is considered one of the most important aesthetics in graph drawing. Edge crossing number negatively affects task performance with statistical significance and should be minimized whenever possible in drawing graphs [4]. Although Kobourov claimed that the number of edge crossings only significantly affected performance for small graphs and the impact was not that important for large graphs, the experiments were only done on graphs with less than 120 vertices and density fewer than 2.5 [20].

- Angle Size: Average size of crossing angles.
  Another attribute that may negatively influent the readability of a graph is the presence of how close edges are to each other, especially when these edges are adjacent [1]. The aesthetic criterion of maximizing the angles among incident edges can make graph layouts more readable, since there is some correlation between the involved angles and the visual distinctiveness of the edges [1,16].
  In this study, suppose θ is the original angle size, the finalized angle size is represented using ($90^o - \theta$). Thus, the smaller the finalized angle size is, the better the final layout is.

- Angle Dev.: The standard deviation of crossing angles.
  Angle Dev. is a measure that is used to quantify the amount of variation of a set of angles among incident edges. A lower standard deviation indicates that the angles tend to be closer to the expected angle value (maximized angle), while a higher standard deviation indicates that the angles are spread out over a wider range of angle values. A smaller difference implies a better angel dev.

- Angle Dev.(90°): The standard deviation of difference on crossing angles to 90°.
  Angle Dev.(90°) is a measure that is used to quantify the amount of variation in the differences of a set of angles among incident edges to 90°. A smaller difference implies a nearer angle to 90° and easier-to-recognize edge crossing.

- Edge Dev. Scale: EdgeDev./EdgeLength
  One of the main aesthetics in graph drawing is to keep edge lengths uniform [3]. Suppose EdgeDev. means standard deviation of edge length, a lower standard deviation represents more uniform edge length. EdgeLength indicates the average edge length. A smaller difference implies a better uniform edge length.

- Angular Res.: Angular resolution is measured as the average of differences between the smallest angle and the optimal angle for each vertex.
  A smaller difference implies a better angular resolution (a better layout) [8] (Suppose that vertex a has at least two incident edges. Let φ be the optimal angle ($360°/deg(a)$), θ be the angle formed by a pair of two neighbouring edges (a, b) and (a, c)).

- Angular Dev.: The standard deviation of angular res.
  A smaller difference implies a better angular deviation.

2.4. Procedure

The evaluation framework is proposed as follows:

- Stock raw data collection from the ASX
- Graph model conduction
  Eventually, five undirected/weighted graphs were generated.
- Force-directed algorithms selection;
  Four common force-directed algorithms were applied for evaluation.
- Graph layouts generation
  Applied selected force-directed algorithms onto five graph models
- Measured and compared layouts' aesthetic quality of the resultant drawings based on seven factors to find out the better approach that suits general-purposed aesthetic criteria requirements in graph drawing fields.

2.5. Tasks

The task was to experimentally compare the performances of the finalized graph layouts generated by four different force-directed algorithms using real data collected. Other than describing graph layout performance only by personal judgment, seven aesthetic criteria were involved to provide comparison details.

2.6. Hypotheses

In this study, the effects of seven aspects were explored on 20 finalized graph layouts. We hypothesized that although the aesthetic criteria in graph drawing area conflict with each other, some 'above-average' algorithms could be chosen, which could keep the balance among multiple aesthetic criteria. The benefits of each method could be analyzed as well; hence, detailed algorithm selection can proceed depending on specific requirements.

3. Results

3.1. Experimental Environment

- Hardware:

- Data process:
  - i7-4770 CPU @3.40GHz
  - 32.0 GB RAM
  - 250 GB SSD Hard disk
- Graph generation:
  - i5-4210 CPU @2.40GHz
  - 16.0 GB RAM
  - 1 TB Hard disk

- Software:
  - Windows 10 Pro (64-bit)
  - Visual Studio 2017

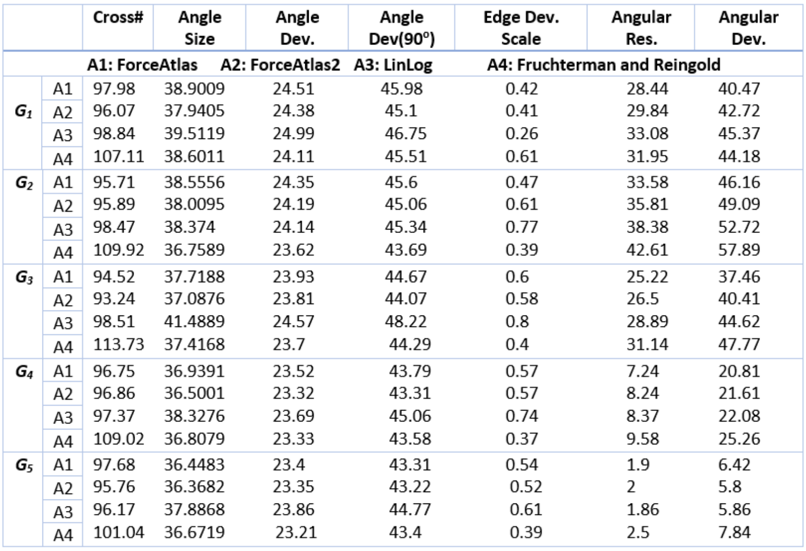
3.2. Graph models processed

Eventually, 2,203 stocks had been kept for further experiments and five relevant graph models were randomly produced to present the stock market network. See below for details:

- $G_1 = (V_1, E_1)$ ($|V_1|$=115, $|E_1|$=497);
- $G_2 = (V_2, E_2)$ ($|V_2|$=252, $|E_2|$=1668);
- $G_3 = (V_3, E_3)$ ($|V_3|$=317, $|E_3|$=3572);
- $G_4 = (V_4, E_4)$ ($|V_4|$=334, $|E_4|$=6654);
- $G_5 = (V_5, E_5)$ ($|V_5|$=339, $|E_5|$=11473);

3.3. Experimental results

We applied four force-directed algorithms on five graph models generated, twenty graph layouts were produced and compared based on seven performance measurements described before (edge crossing number and angle size etc.).



A1: ForceAtlas A2: ForceAtlas2 A3: LinLog A4: Fruchterman and Reingold

		Cross#	Angle Size	Angle Dev.	Angle Dev(90°)	Edge Dev. Scale	Angular Res.	Angular Dev.
***G*₁**	A1	97.98	38.9009	24.51	45.98	0.42	28.44	40.47
	A2	96.07	37.9405	24.38	45.1	0.41	29.84	42.72
	A3	98.84	39.5119	24.99	46.75	0.26	33.08	45.37
	A4	107.11	38.6011	24.11	45.51	0.61	31.95	44.18
***G*₂**	A1	95.71	38.5556	24.35	45.6	0.47	33.58	46.16
	A2	95.89	38.0095	24.19	45.06	0.61	35.81	49.09
	A3	98.47	38.374	24.14	45.34	0.77	38.38	52.72
	A4	109.92	36.7589	23.62	43.69	0.39	42.61	57.89
***G*₃**	A1	94.52	37.7188	23.93	44.67	0.6	25.22	37.46
	A2	93.24	37.0876	23.81	44.07	0.58	26.5	40.41
	A3	98.51	41.4889	24.57	48.22	0.8	28.89	44.62
	A4	113.73	37.4168	23.7	44.29	0.4	31.14	47.77
***G*₄**	A1	96.75	36.9391	23.52	43.79	0.57	7.24	20.81
	A2	96.86	36.5001	23.32	43.31	0.57	8.24	21.61
	A3	97.37	38.3276	23.69	45.06	0.74	8.37	22.08
	A4	109.02	36.8079	23.33	43.58	0.37	9.58	25.26
***G*₅**	A1	97.68	36.4483	23.4	43.31	0.54	1.9	6.42
	A2	95.76	36.3682	23.35	43.22	0.52	2	5.8
	A3	96.17	37.8868	23.86	44.77	0.61	1.86	5.86
	A4	101.04	36.6719	23.21	43.4	0.39	2.5	7.84

Figure 1. Experimental results

Figure 1 showed the final testing results. For example, the edge crossing number on G_1's final layout after applied FA was 97.98 * 10k, and it was 107.11 * 10k after applied FR, that FA is 'better' than FR here in this simple case. In G_1, between FA2 and FR, FA2 produced 10.3% less edge crossing and 6.7% less angular resolution; in G_4, it presented 5.2% less edge crossing, 0.8% less average angle size and 20% less angular resolution as well. In most cases from testing, FA2 produced better layouts, which kept balance in those aesthetic criteria factors. FR had the worst edge crossing.

From Figure 1, some facts could be seen that:

- In most cases, FA and FA2 produced fewer edge crossings, and FR presented worst.
- Regarding the angle size, LinLog generated 'bad' layouts comparing to other three methods.

- In terms of the angle dev., FR's results were a bit better which came with smaller standard deviation of crossing angles.
- For the Angel Dec.(90°) factor, the results were similar to the angle dev.'s, that showed LinLog was not good at generating crossing angles closer to 90° and FR excelled at it.
- In regard to the Edge Dev. Scale, FR produced uniform edge length in most cases (large graphs), but failed in the smallest graph G_1.
- For the angular res, FA and FA2 tend to have the similar 'better' angular resolution, and FR was the worst especially when dealing with large graphs.
- Finally, the angular dev.'s results indicated the same effects as the angular res's due to their feature's similarity.

The final comparisons were also shown in Figure 2. It offered a parallel coordinates layout in which five color groups indicated five different graph models, and the five highlighted blue lines presented aesthetic measurements of final layouts from the FA2 method. Scale adjustments had been done on final testing data to avoid group overlaps, which offers a 'clear' view of measurement.

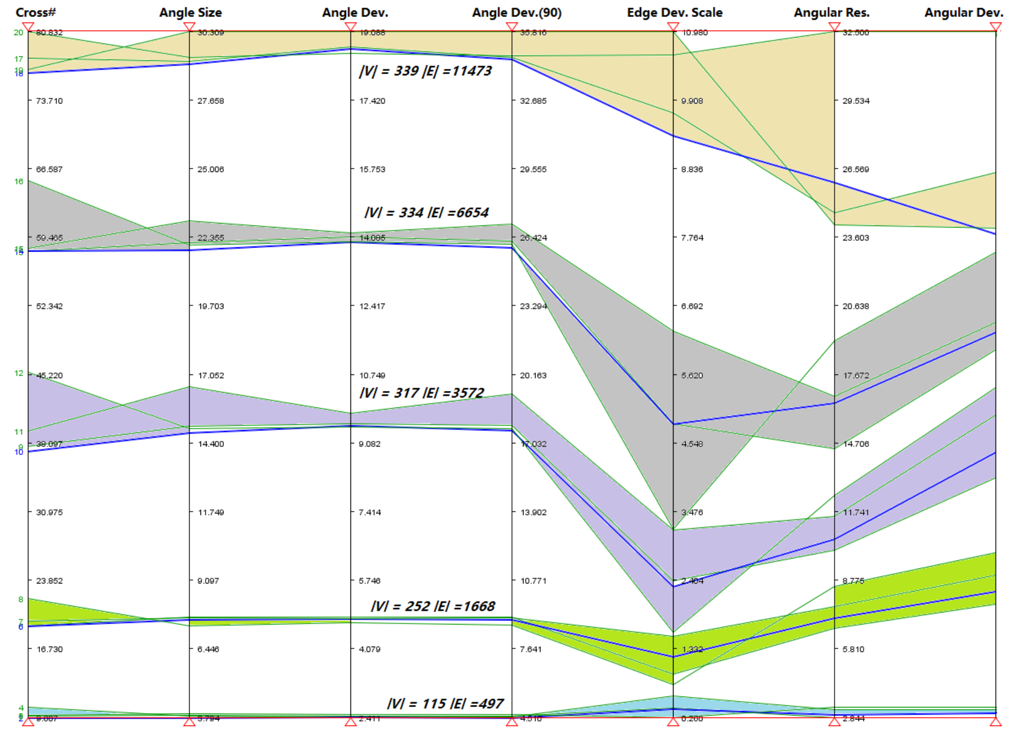


Figure 2. Comparisons of aesthetic measures among four algorithms

4. Discussion

Several attempts have been tried in regard to the graph layout performance measurement although the experiments were all limited to small datasets. For example, Argyriou's tests on angular resolution and crossing resolution were processed within graphs with less than 99 vertices and 135 edges; in Kobourov's experiments regarding the edge crossing, graphs involved contained no more than 120 vertices. Huang's evaluation was tested on a graph with 17 nodes and 29 edges. To fulfil the purpose of examining graph layout performance with more practical significance, this approach was employed to allow us to conduct the analytics experiment in a more realistic environment on a large dataset collected from the Australian stock market. The differences between this approach and other methods were as follows:

- This approach applied a real practical large-scale dataset.
- This approach adopted four force-directed algorithms in experiments.
- This approach provided a comprehensive comparison of seven factors.
- This approach compared results among twenty final graph layouts to examine the algorithms' performances.

The early outcomes of experiments showed:

- FA2 satisfied aesthetic criteria in graph drawing most. It excelled in edge crossing reduction, angle size maximization and angle dev./dev.(90^o), and generated layouts well in edge dev.scale, angular res./dev.
- FR method could offer more uniform edge length.
- Algorithms that produced fewer edge crossings normally generated worse edge dev. scale (uniform edge length).
- In addition, layouts with less angle dev. tended to have more edge crossings.

The hypnosis was confirmed from those findings in this study. It was found that if specific requirements were ignored, some ‘above-average’ algorithms could be selected. They may keep a balance among those aesthetic criteria in graph drawing, although on the other hand, those balanced methods may not excel at some particular aspects. For instance, FA2 acted well in reducing edge crossing, but produced less uniform edge length. Detailed algorithm selection still needs to be done via specific needs analytics. Furthermore, to provide users with ‘clear’ views of final layouts, navigation features etc. need to be conducted, see [11].

5. Conclusions and future works

Based on practical real data collected and finalized, our experiments compared four common applied force-directed algorithms and followed seven aesthetics criteria. The early outcomes showed that FA2 provided above ’average’ performance layouts, came with less edge crossing and angular resolution, etc. Those experimental results offer detailed measurements with empirical evidence other than only personal judgement.

In our experiments, only highly structured data were involved, and seven aesthetics factors have been measured, which may affect the final accuracy of the experiment outcomes. Since most graph drawing aesthetics criteria conflict with each other, detailed force-direct algorithm selection is still relying on specific requirements analytics. In our future work, more factors (time complexity etc.), data types and graph layout comparisons will be considered.

References

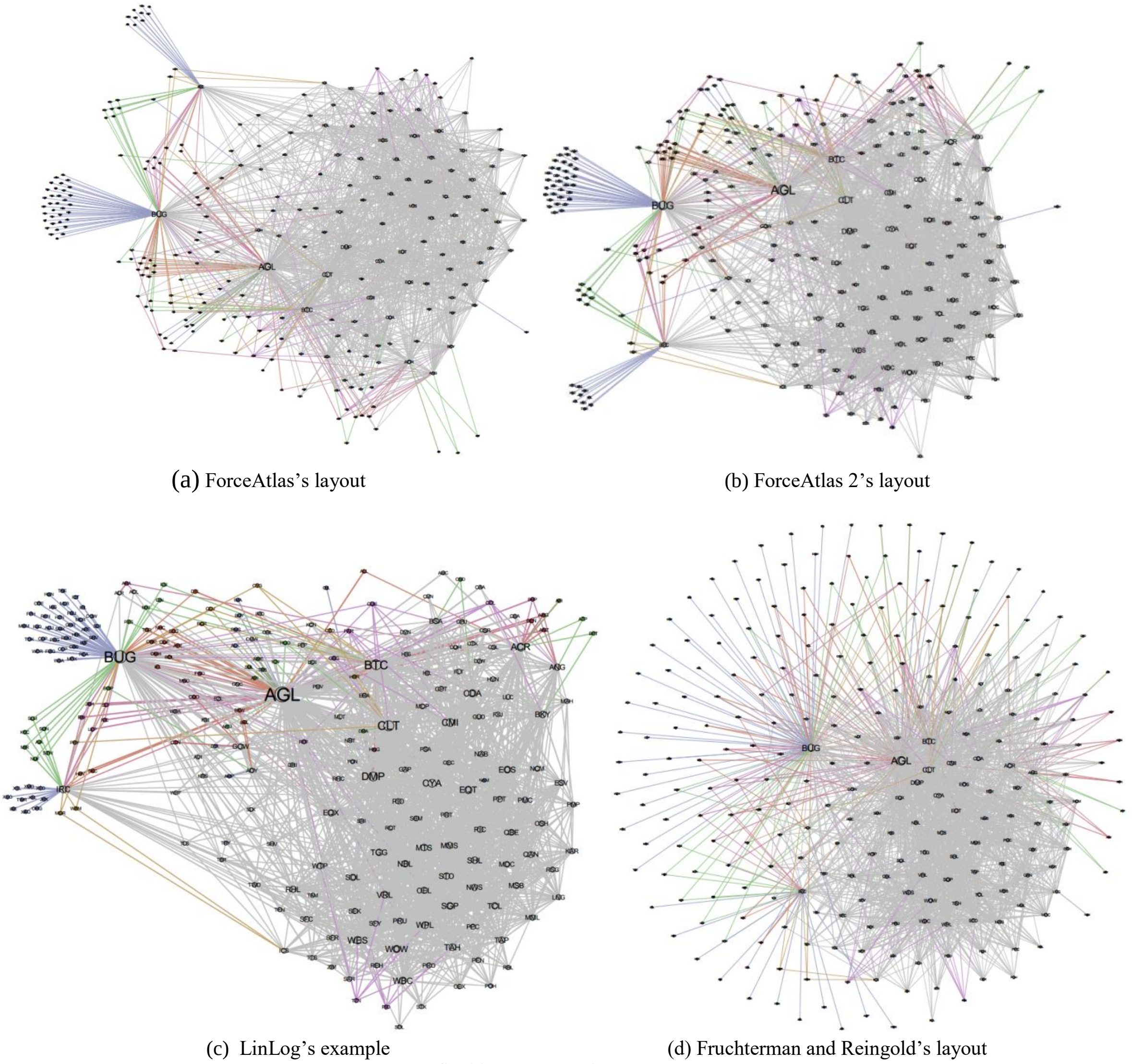
1. E. N. Argyriou, M. A. Bekos, and A. Symvonis, “Maximizing the Total Resolution of Graphs,” Comput. J., vol. 56, no. 7, pp. 887–900, 2013.
2. F.J. Brandenburg, M. Himsolt, and C. Rohrer, “An Experimental Comparison of Force-Directed and Randomized Graph Drawing Algorithms,” Proc. Symp. Graph Draw. (GD 1995), pp. 76–87, 1996.
3. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, “Algorithms for Drawing Graphs an Annotated Bibliography,” Comput. Geom. Theory Appl., vol. 4, pp. 235–282, 1994.
4. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, “Graph Drawing: Algorithms for the Visualization of Graphs,” 1999.
5. G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu, “An Experimental Comparison of Four Graph Drawing Algorithms,” Comput. Geom., vol. 7, no. 5–6, pp. 303–325, 1997.
6. M. Bastian, S. Heymann, and M. Jacomy, (2009) “Gephi: An Open Source Software for Exploring and Manipulating Networks,” Third International AAAI Conference on Weblogs and Social Media. doi: 10.1136/qshc.2004.010033.
7. U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. S. Marshall. “GraphML Progress Report: Structural Layer Proposal,” Proc. 9th Intl. Symp. Graph Drawing (GD ’01), LNCS 2265:501–512. Springer, 2001.
8. B. Finkel and R. Tamassia, “Curvilinear Graph Drawing Using the Force-Directed Method,” in the 12th International Symposium on Graph Drawing (GD’04), 2004, pp. 448–453.
9. T. M. J. Fruchterman and E. M. Reingold, “Graph Drawing by Force-Directed Placement,” Software Practice and Experience., vol. 21, no. 11, pp. 1129–1164, 1991.
10. E. R. Gansner and S. C. North, “Improved Force-directed Layouts,” in Proceedings of Graph Drawing 98, 1998, pp. 364–373.
11. J. Hua, M.L. Huang, M. Zreika, and G.H. Wang, (2017). “Applying Data Visualization Techniques for Stock Relationship Analysis,” In The 3rd International Conference on Fuzzy Systems and Data Mining. Hualien, Taiwan, 24-27 November 2017. iOS press. In Press.
12. M. L. Huang, P. Eades, and J. Wang, “On-Line Animated Visualization of Huge Graphs Using a Modified Spring Algorithm,” J. Vis. Lang. Comput., vol. 9, no. 6, pp. 623–645, 1998.
13. M. Huang, W. Huang, and C. Lin, “Evaluating Force-Directed Algorithms with a New Framework,” in the 27th ACM Symposium on Applied Computing (SAC’12), 2012, pp. 1030–1032.
14. S. Hachul and M. Jünge, “An Experimental Comparison of Fast Algorithms for Drawing General Large Graphs,” in Proc. 13th Int’l Symp. Graph Drawing (GD ’05), 2005, pp. 235–250.
15. W. Huang, P. Eades, S. H. Hong, and C. C. Lin, “Improving Force-Directed Graph Drawings by Making Compromises Between Aesthetics,” Proc. - 2010 IEEE Symp. Vis. Lang. Human-Centric Compute. VL/HCC 2010, pp. 176–183, 2010.
16. W. Huang, and M. Huang, “Exploring the Relative Importance of Number of Edge Crossings and Size of Crossing Angles: A Quantitative Perspective,” Int. J. Adv. Intell., vol. 3, no. 1, pp. 25–42, 2011.
17. Y. Hu, “Efficient and High Quality Force-Directed Graph Drawing,” Math. J., vol. 10, no. 1, pp. 37–71, 2005.
18. M. Jacomy, S. Heymann, T. Venturini, and M. Bastian, “Force Atlas 2, A Graph Layout Algorithm for Handy Network Visualization,” … Http//www. Medialab. Sci. Fr/ …, pp. 1–21, 2011.

19. S. G. Kobourov, "Force-directed Drawing Algorithms," Handb. Graph Draw. Vis. (Discrete Math. Its Appl., pp. 383–408, 2013.
20. S. G. Kobourov, S. Pupyrev, and B. Saket, "Are Crossings Important for Drawing Large Graphs?" Graph Draw., pp. 234–245, 2014.
21. A. Noack, "An Energy Model for Visual Graph Clustering," Proc. 11th Int. Symp. Graph Draw. (GD 2003), LNCS 2912, pp. 425–436, 2003.
22. A. Noack, "Energy Models for Graph Clustering," J. Graph Algorithms Appl. JGAA, vol. 11, no. 112, pp. 453–480, 2007.
23. M. Zreika, J. Hua, and G.H. Wang, "Applying Data Processing Method for Relationship Discovery in the Stock Market," in the International Conference on Data Science and Business Analytics (ICDSBA 2017), 2017, Springer. In Press.

Appendix

This appendix shows some examples in regard to the final graph layouts produced by four force-directed algorithms, and weighted degree metric was applied on:

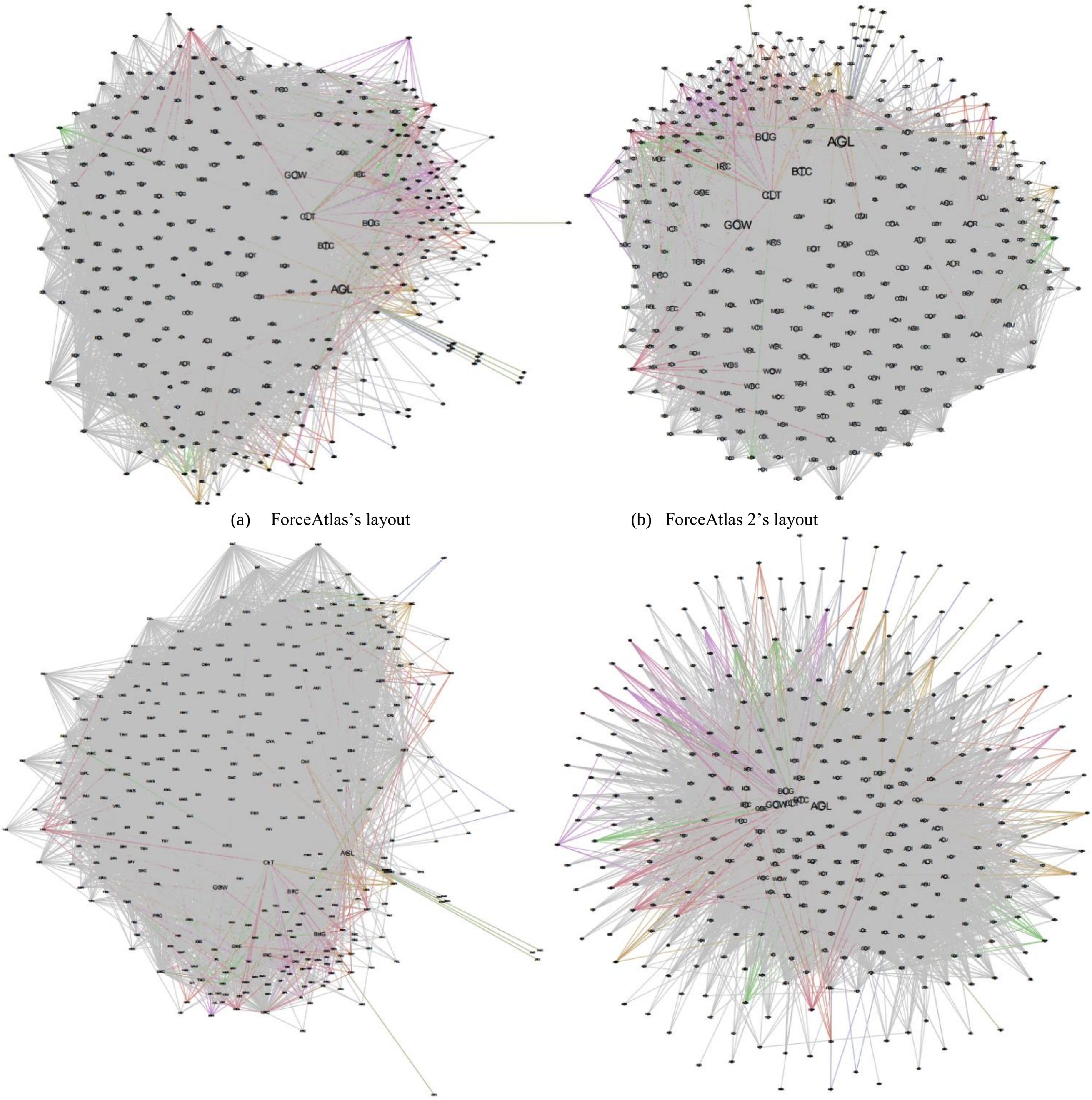
- Node colour: More weighted degree on nodes come with darker colours.
- Node size: More degree, larger size. For example: AGL is normally the largest node in layouts due to its high weighted degree value.
- Node partition: Nodes are clustered based on weighted degree values, and nodes with less values tend to be grouped into grey colours.



(a) ForceAtlas's layout (b) ForceAtlas 2's layout

(c) LinLog's example (d) Fruchterman and Reingold's layout

Figure 3. G_2's final layout examples (V_2|=252, |E_2|=1668)



(a) ForceAtlas's layout

(b) ForceAtlas 2's layout

(c) LinLog's example

(d) Fruchterman and Reingold's layout

Figure 4. G_4's final layout examples ($|V_4|$=334, $|E_4|$=6654)