Physica A 539 (2020) 122948

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

PLANET: A radial layout algorithm for network visualization

Ge Huang [a], Yong Li [b,*], Xu Tan [c,*], Yuejin Tan [a], Xin Lu [a,d,e]

[a] College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[b] College of Economic and Management, Changsha University, Changsha 410022, China
[c] School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 518172, China
[d] School of Business, Central South University, Changsha 410083, China
[e] School of Mathematics and Big Data, Foshan University, Foshan 528000, China

ARTICLE INFO

Article history:
Received 8 January 2019
Received in revised form 18 July 2019
Available online 26 September 2019

Keywords:
Network visualization
Radial layout
Hierarchical structure
Tree layout

ABSTRACT

Tree layouts are among the key approaches for network visualization, and are of particular importance for exploring the hierarchical structure of networks. However, visualizations of the overall network structure and its hierarchical relationships can rarely be optimized simultaneously. This paper presents a radial layout algorithm called PLANET that enables users to explore the network structure from a root node, while maintaining readability. In order to distribute the nodes evenly and minimize edge crossings, we define a list of angle assignment rules for displaying child nodes which can automatically maximize the tunable angles between parent and child nodes, and to uniformly divide the angles of child nodes. Using these rules, the structural properties of the network such as hubs can be properly conveyed, and the readability of nodes that are far from the root can be guaranteed. Our experimental results show that PLANET is comparable to similar algorithms in terms of execution time, and gives better performance in terms of node distribution, variance of edge length and number of edge crossing; these advantages become greater for networks with large diameters.

1. Introduction

Many complex systems in nature and human society can be represented by networks, such as the biological food-chain network, social networks, academic citation networks and so on. The issue of how to efficiently browse, understand, excavate and navigate a large amount of network structure data has become pressing within the field of information science. As an important technique in the study of complex networks, network visualization can clearly show the structure of a network and its internal and external relationships by displaying the network data graphically, thus assisting users in achieving a better understanding of the complex network and system, and in excavating valuable information hidden in the structure.

There are several important issues related to the study of network visualization. The first is the presences of edge crossings in the graph, which directly affect its readability. Minimizing the number of edge crossings is a common optimization goal for many visualization algorithms [1,2]. The second issue is that the layout algorithm should be completed within a reasonable time. Most popular layout algorithms encounter the problem of computational complexity in the visualization of large networks; for example, the time complexity of the force-directed method is $O(N^3)$. Furthermore, although a layout

* Corresponding authors.
E-mail addresses: stone_liyong@sina.com (Y. Li), tanxu_nudt@yahoo.com (X. Tan).

https://doi.org/10.1016/j.physa.2019.122948
0378-4371/

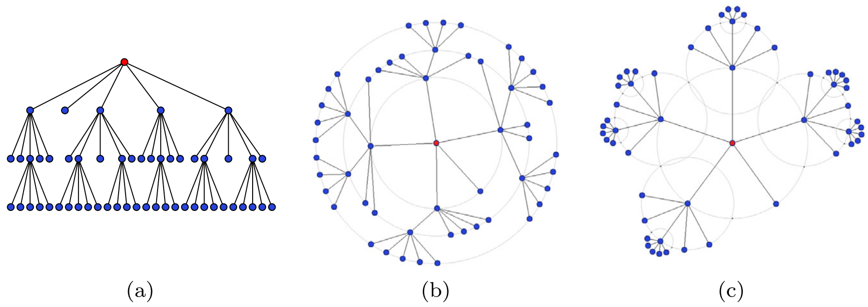


Fig. 1. Schematic chart for visualization of (a) the Reingold–Tilford algorithm, (b) the root-centric radial layout algorithm [5] and (c) the parent-centric radial layout algorithm [5].

algorithm displays all of the nodes and edges in the drawing, it may be hard to recognize the hierarchical structure from the results.

In planar graph visualization techniques, trees can be laid out properly on a plane with no edge crossings, in linear time. This can clearly reflect the intrinsic hierarchy of the data and is straightforward to understand. For this reason, several studies have extracted spanning trees from networks and visualized these rather than the networks themselves [3,4].

In this paper, we present a parent-centric layout algorithm for networks (PLANET), a visualization technique that can help users to explore a network by starting from a node and radially expanding this to the entire network. Our approach involves transforming a graph into a spanning tree or a tree with links, and using angle assignment rules for child nodes to produce drawings with uniform node distribution and minimal edge crossings. We call this approach PLANET, since each child node in the algorithm is represented in a polar system centered at its parent node, in the same way as a planet revolving around a fixed star while surrounded by its own satellites in a celestial system. PLANET allows users to view associations or hierarchies of particular entities and makes it easy to find the influence of a node in networks by making it as the root node, which is useful for visualizing data that contains many links and highlighting influential nodes in networks.

The main advantage of PLANET is its capability to show more clearly hierarchical relationship without compromising the clarity of the overall network structure. Another important feature of PLANET is that it produces a uniform layout in limited display size by specifying the angular extent of each node. The edge crossing in the graph produced by PLANET is less than that by similar tree layout techniques [5]. Besides showing basic visualization results common in many existing tree layout systems, PLANET also allow user to adjust the size of nodes, the length and thickness of edges.

The remainder of this paper is organized as follows. In Section 2, we give an overview of tree visualization techniques. Section 3 describes the proposed PLANET algorithm in detail. In Section 4, we apply the new algorithm to the visualization of various types of networks, and compare the visualization performance and measurements with classical methods. In Section 5, we discuss the advantages and disadvantages of the PLANET algorithm, and possible directions for future work. Finally, we summarize the results and present a conclusion in Section 6.

2. Related work

Since the 1980s, there have been a number of studies of hierarchical structure visualization in networks, which have been developed to describe how objects are ranked and ordered together in an organization or system, such as the evolution of natural species, management of computer file storage systems, etc. [6]. These visualization methods include tree drawings, tree and link drawings and treemaps. The Reingold–Tilford layout is one of the most well-known tree drawing algorithms, which was designed for binary trees [7] and was later extended to trees of any degree [8] (Fig. 1(a)).

Traditional tree layout algorithms usually place nodes from top to bottom or from left to right, and although this is a simple and widely used method, the utilization of space is not effective. In response to this limitation, researchers developed the radial tree [9]. This is a circular tree that can make better use of the drawing area and create more aesthetically pleasing results than traditional trees (Figs. 1(b)and 1(c)).

Most of the existing radial layout algorithms place the root node at the center of the coordinate system, and the other nodes then radiate outward in evenly separated concentric rings, with one ring for each set of ith order neighbors of the root node [10]. The distance of a non-root node from the root node determines the ring in which the non-root node is located (Fig. 2(b)). Although sibling nodes (child nodes of a given parent node) belong to the same ring, the distance between these sibling nodes and the parent node are different. Pavlo et al. extended the traditional radial layout by placing each node on an annulus wedge centered on its own parent node rather than on a concentric ring centered on the root node. This approach naturally reflected the self-similar structures of tree branches, producing visually pleasing layouts (Fig. 2(c)) [5]. However, as the tree grows higher, the annulus wedges of subsequent nodes quickly decrease in size, remote edges may cross or become too short to see, and parent and child nodes may overlap. To help users explore

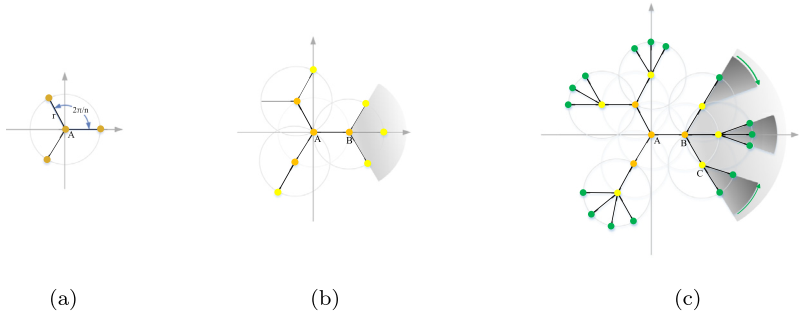


Fig. 2. The layout process in PLANET. In (a), starting from the positive axis, the child nodes of the root node A are uniformly placed counterclockwise on the ring, with A as the center and r as the radius. In (b), node B, one of child nodes of node A, is given the same angular space (light shading) as distributed in (a) for its subtree. In (c), node C, one of three child nodes of node B, is assigned one third of the angular space for its descendant nodes, as shown in the darker shaded area.

the network more effectively, Pavlo et al. developed an interactive exploration tool in which users can change the layout by selecting a new node as the root node. To visualize the hierarchical structure of large networks, some researchers have studied high-dimensional radial layouts using a three-dimensional tree [11] or a hyperbolic tree [12].

One of the problems with using a tree to represent a network is that it ignores the other edges in the network, so a more common way to visualize a network is to use a tree-and-link layout. After a spanning tree has been extracted from the network, we can use tree layout techniques to display it, and then add the rest of the edges. This method has been integrated in several popular network visualization tools (e.g. [13,14]).

Another approach to network visualization uses a treemap representation. A treemap is a set of nested rectangles that visualize data with hierarchical relationships [15]. Nodes are represented as rectangles, where the sum of the area of all rectangles represents the size of the whole system, and the size of each small rectangle is proportional to a property of the node. This layout makes full use of the layout space and does not cause overlap due to a large number of nodes. However, since non-leaf nodes are implicitly represented by nested child nodes within their parent node, it is hard for users to identify the hierarchical structure in this approach.

3. Algorithm

We assume that the network $G = (V, E)$ to be laid out is coded using an adjacency matrix, with V and E the sets of nodes and edges, respectively. PLANET produces a drawing of G based on a spanning tree rooted at s. The major goals for the design of our algorithm are that the nodes should distributed evenly in the drawing area, the number of edge crossings should be minimized, and that the hierarchical properties of the structure should be highlighted while maintaining readability of the overall network.

3.1. Aesthetic metrics

Aesthetic criteria enable a quantitative comparison of the aesthetic quality of different layouts. Battista et al. listed some common graph drawing aesthetics: edge crossing, area, edge length, edge bends, symmetry, angular resolution etc., but they also stress that it is often difficult or impossible to simultaneously optimize two or more [16]. In this work, we chose three aesthetic metrics (node distribution, edge length and edge crossing), because they have a normalized form and can be computed in a reasonable amount of time, to quantify the performance of the developed network visualization algorithm.

Standard deviation of node distribution. A good layout strategy should make full use of drawing space, enable nodes distribute evenly within maximum space. To measure the uniformity of nodes, we calculated the standard deviation of node distribution. Suppose the network is divided into n rectangular regions of the same size, the standard deviation of node distribution of a graph is

$$nd = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \bar{s})^2}, \tag{1}$$

where n is the number of sub-regions divided from drawing area, s_i is the number of nodes in the ith region, $\bar{s}$ is the average number of nodes in all regions.

Coefficient of variance of edge length. Since uniform edge lengths is an effective metric for measuring the aesthetic quality of a layout, the coefficient of variance of edge length has been used to quantify this criterion in several studies [17,18].

The coefficient of variance of edge length is defined as

$$cv = \frac{l_\sigma}{l_\mu} = \sqrt{\frac{\sum_{e\in E}(l_e - l_\mu)^2}{|E| \cdot {l_\mu}^2}}, \tag{2}$$

where l_σ is the standard deviation of edge lengths and l_μ is the average edge length.

Number of edge crossing. In graph theory, an edge crossing occurs when either two edges cross at a single point or when two edges overlap, fewer crossing number of a graph makes it easier for people to understand the drawing. We counted the number of edge crossings in different graph to measure this metric.

3.2. Main idea

The layout algorithm is split into two major phases. In the initialization phase, we select a node as root node and place it at the origin of coordinates. In the iteration phase, we extract a spanning tree from the network using breadth-first search (BFS) approach and place all descendant nodes of the root according to the depth of the nodes in the tree. Firstly, we place the n_0 child nodes of the root node uniformly at random on a circle with radius r and centered at the root node, which we call the first level. Then, the descendant nodes of n_0 are placed uniformly at random on annulus wedges with radius r_d and centered at the parent of each node respectively, according to the proposed angle assignment rules for child nodes.

3.3. Angle assignment rules for child nodes

In PLANET, each child node is placed in the polar system centered at the parent node rather than in a single coordinate system centered on the root node. In this way, the edges between parent and child nodes will never cross or overlap, and all sibling nodes have the same edge length. In this way, the hierarchical properties of the network are more clearly demonstrated. The algorithm developed by Pavlo et al. [5] is based on similar ideas, but the remote nodes and edges are not clearly displayed in these drawings. Using the angle assignment rules for child nodes proposed in this paper, the available angular range of each node will not exceed the angular range of its parent node, and the length of an edge between nodes will not be influenced by the angular range of the parent node (see Fig. 2). The angle assignment rules for the child nodes are as follows.

Given a tree $T = (V, E)$, with vertices V and edges E, the polar coordinate of a node can be represented by (r, θ), the rectangular coordinate of a node can be represented by (x, y) and these can be calculated by

$$\begin{cases} x = r \cdot \cos\theta, \\ y = r \cdot \sin\theta. \end{cases} \tag{3}$$

We place the child nodes at coordinates radially as follows:

For each node v of tree T, let the coordinates of the root node (layer 0) be $(0, 0)$, and for each non-leaf node v, let $(v_1, \dots, v_n)$ be v's child nodes. The parameter r in polar coordinates of each layer is some user-defined value, and the parameter θ varies with the number of layers.

Layer 1: Let the polar angle of the ith $(i = 1, 2, \dots, n_1)$ child node of v in the parent-centric model be

$$\theta_1 = \frac{2(i-1)\pi}{n_1}, n_1 \neq 0, \tag{4}$$

where the angle assigned to each child node is $\frac{2\pi}{n_1}$.

Layer 2: Let f_j be the number of child nodes of the jth $(j = 0, 1, \dots, m)$ layer parent node of v, and let the polar angle of the ith $(i = 1, 2, \dots, n_2)$ child node of v be

$$\theta_2 = \begin{cases} \theta_1 & \text{if } n_2 = 1, \\ \theta_1 - \frac{\pi}{f_0} + \frac{2(i-1)\pi}{(n_2-1)f_0} & \text{otherwise,} \end{cases} \tag{5}$$

where the angle assigned to each child node is $\frac{2\pi}{n_2 f_0}$.

Layer 3 and above: Let d be the depth of node v in the tree T, and let the polar angle of the ith $(i = 1, 2, \dots, n_d)$ child node of v be

$$\theta_d = \begin{cases} \theta_{d-1} & \text{if } n_d = 1, \\ \theta_{d-1} + \frac{2(i-1)\pi}{(n_d-1)\prod_{k=0}^{m} f_k} & \text{if } \theta_{d-1} < \theta_{d-2} \text{ and } n_d \neq 1, \\ \theta_{d-1} - \frac{2(i-1)\pi}{(n_d-1)\prod_{k=0}^{m} f_k} & \text{if } \theta_{d-1} > \theta_{d-2} \text{ and } n_d \neq 1, \\ \theta_{d-1} - \frac{\pi}{\prod_{k=0}^{m} f_k} + \frac{2(i-1)\pi}{(n_d-1)\prod_{k=0}^{m} f_k} & \text{otherwise,} \end{cases} \tag{6}$$

where $m = d - 2$, and the angle assigned to each child node is $\frac{2\pi}{n_d \prod_{k=0}^{m} f_k}$.

3.4. Layout algorithm

Given a network G, we first need to select a node as the root of the tree. Although by default the node with highest degree is selected as the root, other criteria are also applicable. We then use the BFS method to iteratively calculate and record the polar coordinates of each node relative to the parent node, which is then converted into absolute Cartesian coordinates.

In the following algorithm $PLANET(G)$, G is an adjacent matrix of a network, s is the root node, d is the depth of each node, r_0 is the user-defined edge length of the first layer and ξ is the user-defined constant for different layers. $adj[v]$ is the set of nodes adjacent to v after removing the visited node from the network, and $(XY(v)_{px}, XY(v)_{py})$ are the Cartesian coordinates of node v's parent node. $Enquene(Q, v)$ means add the set of nodes v to the queue Q, while $Dequene(Q)$ means remove the head element from the queue Q. The result of the algorithm XY are the absolute Cartesian coordinates of all nodes in the graph.

As each node in the queue Q is traversed and their coordinates calculated, the overall time complexity of this method is $O(N)$, where N is the number of nodes in the network. Hence, the computation time increases linearly with the size of the network.

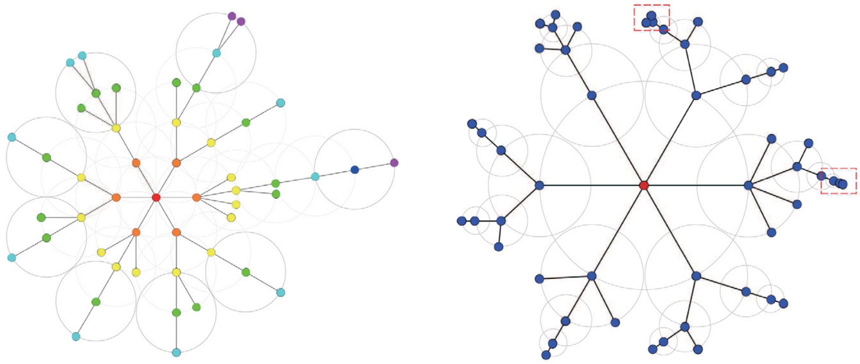
Algorithm 1 PLANET(G)

Input: G-an adjacent matrix of a network
Output: XY-the Cartesian coordinates of all nodes

//Initialization
$XY(s) \leftarrow (0, 0)$
$d \leftarrow 0$
$r_0 \leftarrow 1$
$Q \leftarrow adj(s)$
//Iteration: calculate the coordinate of nodes on dth levels
while $Q \neq \emptyset$ **do**
  $d \leftarrow d + 1$
  $r_d \leftarrow r_0 + (\xi \cdot d)$
  if d=1 **then**
    for each $v \in Q$ **do**
      //Calculate v's polar coordinates relative to parent node
      $\theta \leftarrow$using formula (4)
      //Convert v's polar coordinates to absolute Cartesian coordinates
      $XY(v) \leftarrow (x, y) \leftarrow (r_d \cdot cos\theta, r_d \cdot sin\theta)$
      $m \leftarrow adj[v]$
      $Enqueue(Q, m)$
    end for
  else if d=2 **then**
    for each $v \in Q$ **do**
      $\theta \leftarrow$using formula (5)
      $XY(v) \leftarrow (XY(v)_{px} + r_d \cdot cos\theta, XY(v)_{py} + r_d \cdot sin\theta)$
      $m \leftarrow adj[v]$
      $Enqueue(Q, m)$
    end for
  else
    for each $v \in Q$ **do**
      $\theta \leftarrow$using formula (6)
      $XY(v) \leftarrow (XY(v)_{px} + r_d \cdot cos\theta, XY(v)_{py} + r_d \cdot sin\theta)$
      $m \leftarrow adj[v]$
      $Enqueue(Q, m)$
    end for
  end if
  $Dequeue(Q)$
end while

4. Experimental results

To demonstrate the effectiveness of our algorithm, we apply it to the visualization of a spanning tree and three empirical network datasets, and compare the performance of this layout to the results of Pavlo et al.'s algorithm. In



(a) (b)

Fig. 3. Two drawings of the same tree, using (a) PLANET and (b) Pavol et al.'s layout algorithm.

all comparison cases, PLANET produces attractive visualization results. The layout helps to give an easier and clearer understanding of the hierarchical structures of the network and reduces the number of edge crossings.

4.1. Spanning tree of random graph

We first describe an experiment which tests and compares both PLANET and Pavlo et al.'s algorithm [5], using spanning trees extracted from randomly generated graphs and with a randomly selected root node.

Fig. 3 shows the results of two drawings generated from the same tree. We used a spanning tree with 50 nodes from a random graph with a height of six. Pavlo et al.'s algorithm produces a symmetrical drawing. As the height of the tree increases, the containment circle of remote descendants cannot be reduced any further, and the descendant nodes cannot be displayed. As shown in the dashed rectangles in Fig. 3(b), the parent and child nodes are squeezed together. PLANET (Fig. 3(a)) produces a better graph in which the remote descendants are displayed clearly. All of the edges in the figure are equal in this example, but the edge length can be set according to the user's needs. The addition of more layers of sub-nodes will not affect the readability of the graph.

4.2. Empirical social networks

Visualization techniques have been used for decades to analyze social networks. In this field, a radial layout works as a "target sociogram" and is used to explore social networks from a single individual's perspective. The placement of the nodes in the radial layout makes it easy to discern how far and deep information can travel from the root node. We tested the proposed algorithm using three datasets and compared it with Pavlo et al.'s algorithm in the first two examples.

In the first example, we used a subset of a retweet network dataset from Rossiand and Ahmed [19], which represents retweet relationships between Twitter users and was collected from various social and political hashtags. We compare the results of the spanning tree and the tree-and-link drawing of the retweet network with those of Pavlo et al.'s algorithm. In Pavlo et al.'s algorithm, the node's angular space at the first level needs to be set manually, and the greater the assignable angle, the more easily remote edges can cross. For comparison, we set the assignable angle of the first level in Pavlo et al.'s algorithm the same as that in our algorithm.

Fig. 4(a) and (b) show the comparative results for the spanning trees. All nodes are visible and can be clearly in the results from PLANET, but in the drawing produced using Pavlo et al.'s algorithm, some child nodes that are further away from the root nodes overlap with the parent nodes, and some remote edges are crossed. Without auxiliary rings, our algorithm looks clearer and more aesthetically pleasing.

By adding other edges to the network, a spanning tree drawing can be transformed into a tree-and-link layout. The latter can show not only the hierarchical relationship in the tree, but also the non-inheritance relationship between nodes in the network. Fig. 4(c) and (d) present comparative results for the drawing of the tree-and-link scheme. We adjust the edge lengths and node sizes in the tree-and-link layout in our algorithm, since suitable adjustments of edge lengths and node sizes can make the topological features of the network clearer. The node size represents the degree of the node, and the edge length increases as the height of the tree increases. As shown in Fig. 4(c) and (d), PLANET produces a satisfactory drawing that reflects the overall structure of the network and the interaction between nodes of different levels, while Pavlo et al.'s algorithm produces an untidier result due to the reduction in the length of the remote edges.

The Zachary Karate Club network consists of 34 nodes and 78 edges, in which nodes represent members of the club, and edges represents friendships between members. Fig. 5 presents comparative results for PLANET and Pavlo et al.'s

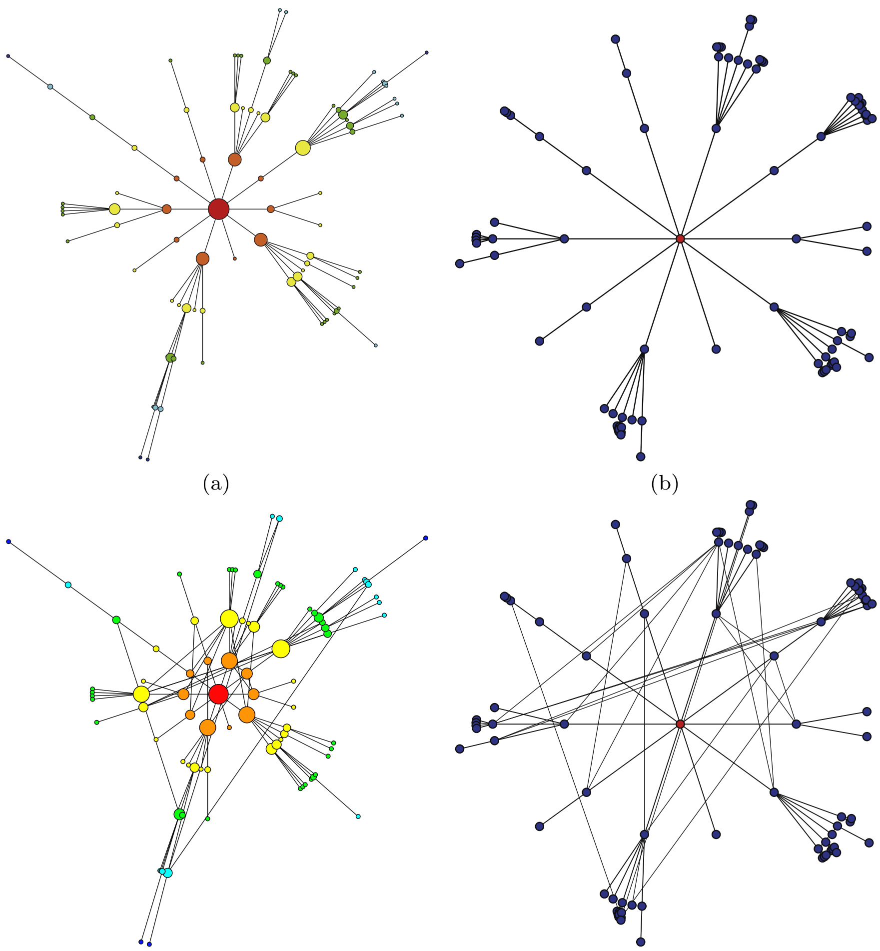

(a) (b) (c) (d)

Fig. 4. An example graph of a retweet network, generated using two algorithms: a spanning tree drawing by (a) PLANET and (b) Pavlo et al.'s algorithm; and a tree-and-link drawing using (c) PLANET and (d) Pavlo et al.'s algorithm.

algorithm for this network. After the rest of the edges are added to the spanning tree, the interaction between nodes at different levels can be observed, and this is clearer in PLANET. As Fig. 5(a) shows, the leaf nodes that are further from root node have weaker connections with non-leaf nodes. Moreover, users can identify other important nodes from the graph produced by PLANET.

To test the performance of PLANET on weighted networks, we conducted an experiment on a sawmill employee communication network and an ant colony interaction network. The first dataset is collected from a small sawmill, the nodes represent the employees in sawmill, including planer, mill, manager etc., and the edges represent the communication relationship between employees [20]. If the communication frequency between two employees exceeds a certain evaluation level, there will be an edge between them in the communication network. The network contains 36 nodes and 62 edges, and we assign a random weight to each edge. Fig. 6 is the result of communication network generated by PLANET, where the black edge represents the edge of the spanning tree and the gray edge represents the rest edges. The thickness of the edge indicates the level of communication frequency. The root node in Fig. 6(a) is a planer named Juan, and the root node in Fig. 6(b) is the manager. Comparing the two figures, it can be found that Juan is more likely to be the central person of the network than the manager. The second dataset is one of the real-world ant interaction networks collected from Israel [21]. The nodes represent ants in the ant colony and edges represent interaction between ants in the colony. A pair of ants was considered to interact when the front end of one ant was located within the trapezoidal shape representing the other ant. The network is an undirected weighted network which contains 37 nodes and 68 edges. Fig. 7 is the result of ant colony interaction network visualization generated by PLANET, where the black edge represents the edge of the spanning tree and the gray edge represents the rest edges. The thickness of the edge indicates the interaction frequency between ants. The results of the two datasets suggest that the quality of the layouts of weighted network generated by PLANET are also satisfactory.

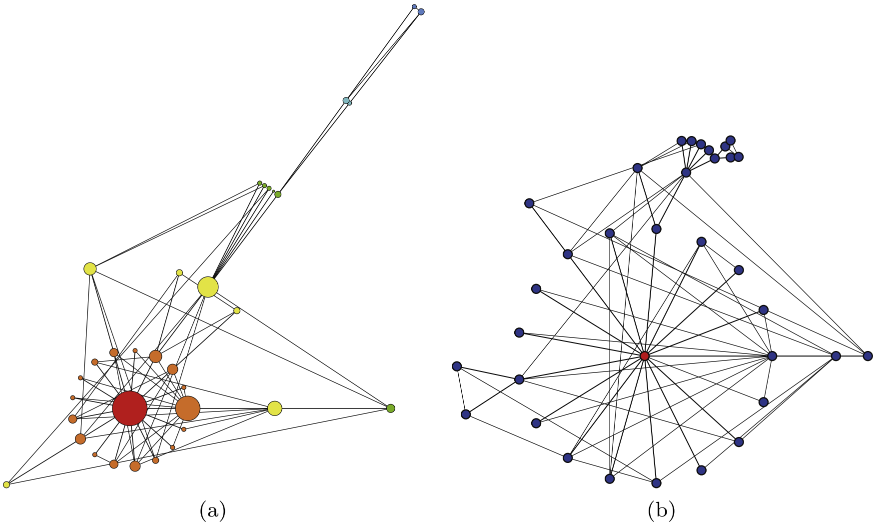


Fig. 5. Graphs of the karate club network generated by (a) PLANET and (b) Pavlo et al.'s algorithm.

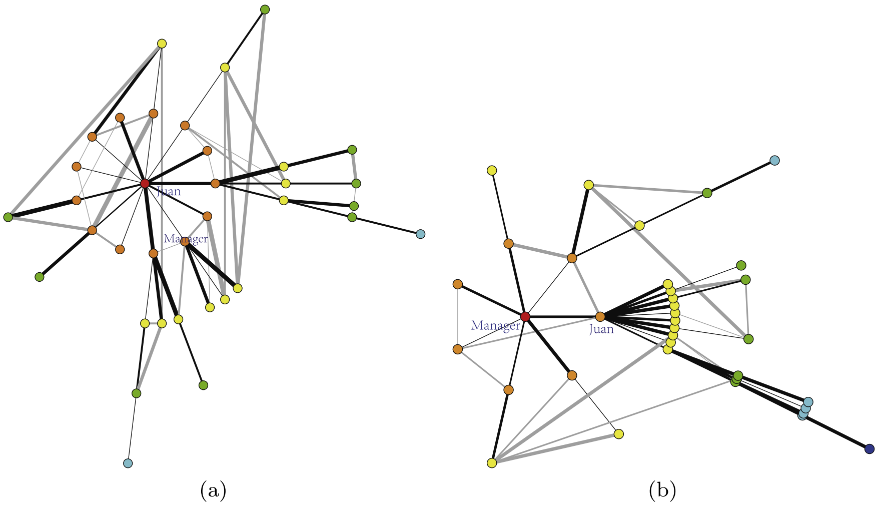


Fig. 6. Visualizations of the sawmill employee communication network by PLANET with the central node rooted at (a) Juan and (b) The manager.

4.3. Comparison of measurements

The natural criteria to measure a layout algorithm in practice are the execution times and the quality of the drawings.

We conducted 2000 experiments on random networks to validate the theoretical analysis of the execution time given above. We generated random networks with a number of edges fixed at 5000, and let the number of nodes increase from 500 to 5000 (a reconnection technique was used to guarantee the connectivity of the generated networks). Fig. 8(a) demonstrates that the execution time of PLANET, which increases linearly with respect to the number of nodes, is comparable to Pavlo et al.'s algorithms.

For the purpose of aesthetics evaluation, we calculated three metrics from experiments with networks of varying sizes: standard deviation of node distribution, coefficient of variance of the edge length and number of edge crossing. Each experiment comprised 400 trials per algorithm, in which 10 random graphs of order 50–500 (inclusive) nodes and order 100–1000(inclusive) edges were generated. Fig. 8(b) presents standard deviations of node distribution of graphs generated from PLANET and Pavlo et al.'s algorithm. The size of sub-region is 0.1*0.1. PLANET achieves encouraging results compared to the Pavlo et al.'s approach, which clearly demonstrates the effectiveness of the proposed angle assignment rules. As shown in Fig. 8(c), coefficients of variance of edge length of graphs generated by PLANET are much lower than Pavlo et al.'s approach. Because the edge length of each layer can be set by user freely in PLANET (all edge lengths are set to 1

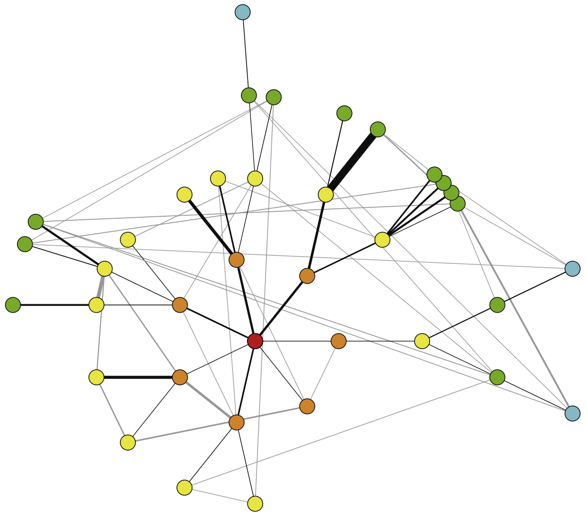


Fig. 7. Visualization of the ant colony interaction network produced by PLANET.

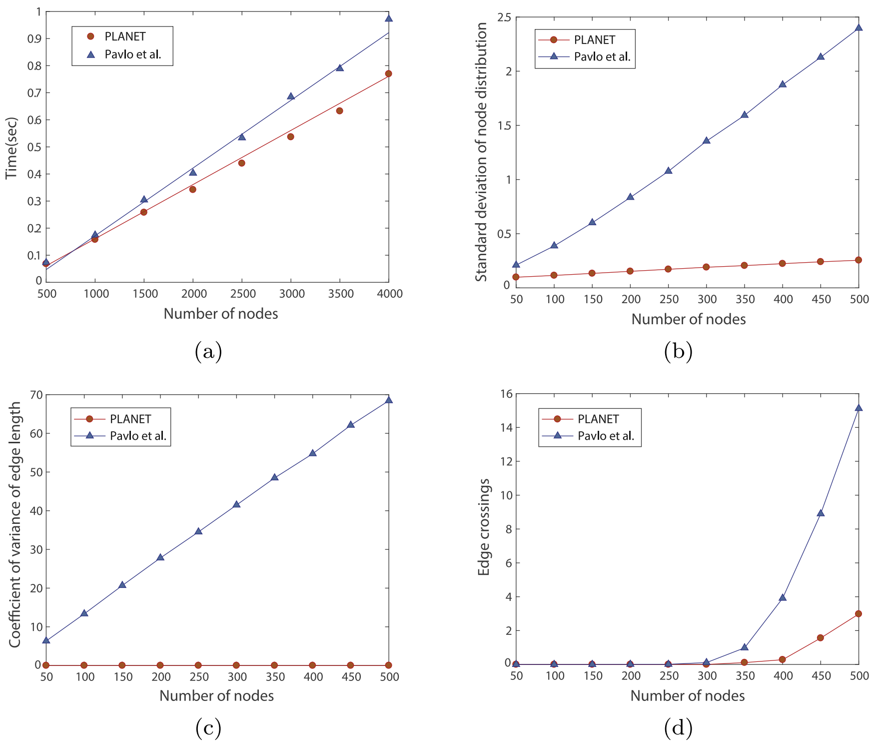


Fig. 8. Measurements of PALNET and Pavlo et al.'s algorithm on (a) execution time, (b) standard deviation of node distribution, (c) coefficient of variance of edge length and (d) number of edge crossing.

in this example) and will have little influence on the final layout, but in Pavlo et al.'s algorithm (the edge length of the first layer is set to 1 in this example), it is inevitable for edge lengths decreased quickly (even decreased to zero) with the number of layers increased. Fig. 8(d) shows numbers of edge crossing of graphs produced by Planet and Pavlo et al.'s visualization scheme. The nodes' angular space at the first layer has been given a low value to avoid edge cross for Pavlo et al.'s approach. As clearly shown in Fig. 8(d), our algorithm produce less crossings than Pavlo et al.'s approach.

5. Discussion

The experimental results show that PLANET is a visualization algorithm with high execution efficiency, which can achieve uniform node distribution, minimum edge crossings and which is suitable for visualizing and analyzing trees and the hierarchical structures of networks. In contrast with traditional radial layout algorithms, sibling nodes (child nodes of a given parent node) are placed close together within a limited angular range. The edge lengths for sibling nodes are equal in PLANET, since this is more consistent with the visual preferences of human beings and the rules of symmetry and aesthetics. Unlike Pavlo et al.'s algorithm, our method does not produce edge crossings or overlap between parent and child nodes in the visualization of a spanning tree. In the visualization of networks, PLANET can show the interaction between different levels of nodes more clearly. Thus, PLANET combines the advantages of a traditional radial layout with those of a parent-centric radial layout algorithm, which reflects the hierarchical relationship between nodes below the root node and retains the overall structure of the network. Although hyperbolic layout algorithms can achieve these goals and take up less space by placing nodes using a non-Euclidean space, PLANET is more effective and is simpler in terms of implementation.

However, it is worth noting that the space utilization of a plane can still be improved in PLANET. Moreover, as the number of nodes increases, sibling nodes may overlap due to the limited angular space of the parent node. Research into modified methods for angular space allocation could lead to a better result that can avoid overlap between sibling nodes and edge crossing between edges, and can improve the utilization of the drawing area. It is also natural to extend our drawing methods to graph animation. The use of focus and context technology and interactive interfaces would allow users to explore and understand the network much more easily.

6. Conclusions

In this paper, we present a parent-centric radial layout algorithm for networks. The algorithm is useful for visualizing networks with many links or large diameters and can be used as an interactive tool for highlighting influential nodes in networks. We introduce angle assignment rules for child nodes, which help to produce a drawing with uniform node distribution, a minimal number of edge crossings and a clear hierarchical structure. We conducted experiments to compare PLANET with the algorithm developed by Pavlo et al. and the results suggest that the quality of the layouts and the computational efficiency of PLANET are satisfactory.

Acknowledgments

GH, YL and XT are supported by the National Natural Science Foundation of China (71771213, 71522014, 91846301), XL and YT are supported by the National Natural Science Foundation of China (71790615, 71690233, 71774168) and Hunan Science and Technology Plan Project (2017RS3040, 2018JJ1034).

References

[1] P. Eades, N.C. Wormald, Edge crossings in drawings of bipartite graphs, Algorithmica 11 (4) (1994) 379–403.
[2] H.C. Purchase, Metrics for graph drawing aesthetics, J. Visual Languages Comput. 13 (5) (2002) 501–516.
[3] M. Jia, S. Swaminathan, E.S. Wurtele, J. Dickerson, MetNetGE: VIsualizing biological networks in hierarchical views and 3D tiered layouts, in: Bioinformatics and Biomedicine Workshop, 2009. BIBMW 2009. IEEE International Conference on, IEEE, 2009, pp. 287–294.
[4] R. Melchiotti, O. Rötzschke, M. Poidinger, ArchiLD: hierarchical visualization of linkage disequilibrium in human populations, PLoS One 9 (1) (2014) e86761.
[5] A. Pavlo, C. Homan, J. Schull, A parent-centered radial layout algorithm for interactive graph visualization and animation, 2006, ArXiv preprint cs/0606007.
[6] P. Eades, A heuristic for graph drawing, Congress. Numer. 42 (1984) 149–160.
[7] E.M. Reingold, J.S. Tilford, Tidier drawings of trees, IEEE Trans. Softw. Eng. (2) (1981) 223–228.
[8] J.Q. Walker, A node-positioning algorithm for general trees, Softw. - Pract. Exp. 20 (7) (1990) 685–705.
[9] P. Eades, Drawing Free Trees, International Institute for Advanced Study of Social Information Science, 1991.
[10] K.-P. Yee, D. Fisher, R. Dhamija, M. Hearst, Animated exploration of dynamic graphs with radial layout, in: InfoVis, IEEE, 2001, p. 43.
[11] J. Rekimoto, M. Green, The information cube: Using transparency in 3d information visualization, in: Proceedings of the Third Annual Workshop on Information Technologies & Systems, WITS'93, 1993, pp. 125–132.
[12] J. Lamping, R. Rao, P. Pirolli, A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., 1995, pp. 401–408.
[13] P. Doemel, Webmap: a graphical hypertext navigation tool, Comput. Netw. ISDN Syst. 28 (1–2) (1995) 85–97.
[14] D. Holten, Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data, IEEE Trans. Visual. Comput. Graph. 12 (5) (2006) 741–748.
[15] I. Herman, M.S. Marshall, G. Melançon, D.J. Duke, M. Delest, J.-P. Domenger, Skeletal images as visual cues in graph visualization, in: Data Visualization'99, Springer, 1999, pp. 13–22.
[16] G.D. Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice Hall PTR, 1998.
[17] S. Hachul, M. Jünger, Large-graph layout algorithms at work: an experimental study, J. Graph Alg. Appl. 11 (2) (2007) 345–369.
[18] O.-H. Kwon, T. Crnovrsanin, K.-L. Ma, What would a graph look like in this layout? a machine learning approach to large graph visualization, IEEE Trans. Visual. Comput. Graph. 24 (1) (2018) 478–488.
[19] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI, vol. 15, 2015, pp. 4292–4293.
[20] W. De Nooy, A. Mrvar, V. Batagelj, Exploratory Social Network Analysis with Pajek: Revised and Expanded Edition for Updated Software, vol. 46, Cambridge University Press, 2018.
[21] D.P. Mersch, A. Crespi, L. Keller, Tracking individuals shows spatial fidelity is a key regulator of ant social organization, Science 340 (6136) (2013) 1090–1093.