Big Data Research ●●● (●●●●) ●●●–●●●

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

# Towards Visualizing Big Data with Large-Scale Edge Constraint Graph Drawing

Ariyawat Chonbodeechalermroong, Rattikorn Hewett

*Department of Computer Science Texas Tech University, Lubbock, TX, USA*

ARTICLE INFO

*Article history:*
Received 22 August 2016
Received in revised form 1 August 2017
Accepted 7 October 2017
Available online xxxx

*Keywords:*
Large graphs
Force-directed
Constraint enforcement methods

ABSTRACT

Visualization plays an important role in enabling understanding of big data. Graphs are crucial tools for visual analytics of big data networks such as social, biological, traffic and security networks. Graph drawing has been intensively researched to enhance aesthetic features (i.e., layouts, symmetry, cross-free edges). Early physic-inspired techniques have focused on synthetic abstract graphs whose weights/distances of the edges are often ignored or assumed equal. Although recent approaches have been extended to sophisticated realistic networks, most are not designed to address very large-scale weighted graphs, which are important for visual analyses. The difficulty lies in the fact that the drawing process, governed by these physical properties, oscillates in large graphs and conflicts with specified distances leading to poor visual results. Our research attempts to alleviate these obstacles. This paper presents a simple graph visualization technique that aims to efficiently draw aesthetically pleasing large-scale straight-line weighted edge graphs. Our approach uses relevant physic-inspired techniques to promote a graphs and proposes a weak constraint-based approach to handle large-scale computing and competing goals to satisfy both weight requirements and aesthetic properties. The paper describes the approach along with experiments on both synthetic and real large-scale weighted graphs including that of over 10,000 nodes and comparisons with state-of-the-art approaches. The results obtained show enhanced and promising outcomes toward a general-purpose graph drawing technique for both big synthetic and real network data analytics.

## 1. Introduction

Visual analytics enable big data understanding and extraction of useful insights. Many real world data, such as paper citations, social, biological and traffic networks, predominantly use graphs as basic structures for visualization. Graph drawing is essential and has been studied intensively [1–10]. Traditionally, graph drawing algorithms have focused on aesthetic properties of synthetic and abstract graphs, where the aesthetically pleasing criteria are based on the graph symmetry and layout to maximize readability and the number of cross-free edges [9].

Early physic-inspired graph drawing techniques are force-directed algorithms [1,7] that model straight-line undirected edge graphs after a spring or electrical system where edges mimic springs. The spring layout method relies on spring forces, behaving similar to those in Hooke's law, that attract or repulse adjacent nodes till they are stable. Unconnected nodes repel each other as electric charged particles do and this nature can evolve to reduce narrow layout and edge crossing. The force-directed method iteratively minimizes the system energy by moving the nodes along the direction of the force. The minimal energy state corresponds to a well-configured layout. Unfortunately, these traditional techniques only perform well for small graphs of no more than a few hundred nodes [9]. One of the major issues for scaling these algorithms is due to many local minima (thus, oscillation and non-converged drawing) in large graphs that lead to unreadable drawings.

To cope with the scalability issue, the force-directed approaches have been extended to many multi-scale (or multi-level) layout approaches [4–7,10] that draw a graph from coarse-scale simple structures to fine-scale complex structures. The techniques involve graph coarsening [5–7,9] including vertex filtrations [4]. The multi-scale approach produces large graphs of over 10,000 nodes in a reasonable time [9]. One of the state-of-the-art well-performing multi-level algorithms is the algorithm by Hu [7]. Hu's algorithm combines the multi-level approach with a method similar to a well-studied $n$-body problem in physics to reduce repulsive force calculations [7,9]. The algorithm is widely used and has been im-

---

*E-mail addresses:* ariyawat.chonbodeechalermroong@ttu.edu (A. Chonbodeechalermroong), Rattikorn.hewett@ttu.edu (R. Hewett).

https://doi.org/10.1016/j.bdr.2017.10.001
2214-5796/

Please cite this article in press as: A. Chonbodeechalermroong, R. Hewett, Towards Visualizing Big Data with Large-Scale Edge Constraint Graph Drawing, Big Data Res. (2017), https://doi.org/10.1016/j.bdr.2017.10.001

plemented on graph drawing engines such as GraphViz [11] and Gephi [12].

Typical force-directed graph drawing performs well on synthesized abstract graphs due to their symmetric nature, however, this is not always the case for graphs representing real data. Moreover, most algorithms either do not deal with the weights/distances of the graph edges or assume them to be equal. However, in practice, weights/distances representing strengths of the relationships between nodes can be crucial for extracting useful insights. For example, edge length in communication network graphs can have an important implication on cost and resource requirements. Recent force-directed algorithms designed to address these issues include ForceAtlas2 [13] that handles weighted edges for real relational networks such as web graphs and social networks. ForceAtlas2 applies a power-law distribution of degree to a modified computation of repulsive forces to rely on the degree of the nodes. Using a strong electrical force, ForceAtlas2 allows visualization of important clusters. ForceAtlas2 has been widely used and implemented in an interactive visualization tool, Gephi. Although recent approaches have been extended to sophisticated real networks, most are not designed to address very large-scale weighted graphs, which are crucial for visual analyses. The difficulty lies in the fact that the drawing process governed by these physical properties oscillates in large graphs or conflicts with specified distances leading to poor visual results. Our research attempts to alleviate these obstacles. This paper presents a simple graph visualization technique that aims to efficiently draw aesthetically pleasing large-scale straight-line weighted edge graphs. The proposed approach is based on a constraint-based approach to cloth simulation [14], commonly used in computer graphic and game development to provide efficient realistic simulation of particles with given distance constraints [14,15]. We introduce a *weak constraint-based approach* that relaxes rigid distance preference requirements (within acceptable bounds) in order to improve graph readability. The weak constraints allow flexibility in trading the distance precision with the layout quality. Using the weak constraint enforcement with appropriate physic-inspired techniques, our approach yields graphs that balance the weight precision with aesthetic quality. Note that the distance is typically assumed to be inversely proportional to the weight. This is because an edge with a high weight represents a spring with a strong attracting force that pulls the edge distance short.

The rest of the paper is organized as follows. Section 2 describes related work followed by our proposed algorithm with some illustrations in Section 3. Section 4 presents experimental results to compare our approach with two state-of-the-art approaches, namely Hu's multi-level algorithm and ForceAtlas2. Section 5 gives a summary and concludes the paper.

## 2. Related work

*Force-directed methods* are among the most well-studied and flexible techniques for drawing graphs with straight-line edges [9]. Classic physic-inspired algorithms mimic spring and electrical systems [1,3,9]. The basic idea is to replace edges with springs, start with a random initial graph layout and then let go the system so that the spring forces, governing by Hookes Law rules, will reconfigure the graph. The force-directed approach maps the graph layout to the energy of the physical system in such a way that low energy corresponds to an aesthetic layout (e.g., equal distance edge, symmetry and well-spaced for non-adjacent nodes) [9]. The goal is to find the graph layout whose energy is minimized. Ideally, the graph drawing is evolving until it converges to a stable state of minimum energy.

Eades' algorithm [1] makes two modifications. One is to use the logarithm strength forces for far away nodes because Hookes Law (linear) springs are too strong. The other is to make non-adjacent nodes repel each other where the repulsive force is inversely proportional to the square root of the edge distance. Fruchterman and Reingold [3] introduced their algorithm using attractive spring forces between adjacent nodes but an electrical repulsive force between all pairs of nodes. The algorithm re-defines new attractive and repulsive forces in such a way that a *high* graph density gives a *low* attractive force but a very *high* repulsive force resulting in "even vertex distribution" so that the layout is well-spaced. The algorithm also adds a notion of "temperature" to control the displacement of the nodes in the same fashion as that of simulated annealing [16] in order to avoid local minima issue in finding the optimal graph layout. This local minima issue can cause oscillation and prevents the algorithm from converging. The problem is more severe when the graphs are large and complex. In fact, this is the main issue with the force-directed algorithms. They perform well for small graphs of a few hundred nodes [9].

Recent ForceAtlas2 [13] is a widely used force-directed algorithm that is designed to efficiently draw graphs with weighted edges on real networks. ForceAtlas2 takes the degree of nodes into account while repulsive forces are being computed. This helps improve node distribution and space layouts. It respects physical laws and provides a mechanism to balance tradeoffs between speed and precision. Since the high speed of node movement leads to high oscillation and inaccurate layouts, ForceAtlas2 estimates appropriate speeds to balance with acceptable accuracies. Although our approach uses physical properties (e.g., Newton's law of movement), the algorithm is driven by constraints (on distances) as opposed to forces. However, it is similar to Fruchterman and Reingold's algorithm in that both approaches use the notion of temperature to alleviate the local minima issue. Our approach is similar to ForceAtlas2 in aiming to balance speeds to reduce oscillation. While ForceAtlas2 approximates appropriate speeds by adapting between local and global speeds, our approach employs Verlet Integration [17] to integrate Newton's law of motion into estimating the trajectories of the node movement.

A *graph theoretic distance approach* introduced by Kamada and Kawai [8] defines a good layout to be when the geometric (Euclidean) distances between nodes are the same as their corresponding graph (theoretic) distances that are computed from the "All-Pairs-Shortest-Path". The approach sets up the spring system whose goal is to minimize the energy, which is equivalent to minimizing the difference between the geometric and graph distances. The approach has been used in several graph drawing approaches. The major issue of this approach is it's computational cost. The "All-Pairs-Shortest-Path" computation is expensive requiring $O(|V|^3)$ time using the Floud–Warshall algorithm [10]. The approach improves graph drawing of large graphs but, it still does not scale well for graphs of thousand nodes.

A large number of force-directed multi-level algorithms [4–7,10] have been studied to cope with scalability issue. The idea is to draw a graph from coarse-scale simple structures to fine-scale complex structures. The technique is referred to as *graph coarsening*. Nodes in a fine-scale level are relocated to capture local organized groups and then are relocated in a coarse-scale level that abstracts essential but simpler layouts. The fine-scale relocations are then performed to verify and correct, local disorders. In [5], the coarse-grain relocation applies energy function of [8] while the fine-grain relocation uses traditional force-directed formulae of [1,3]. Harel and Koren [6] introduced an algorithm using a $k$-center approximation in the coarsening process to find a subset of a set of nodes of size $k$ where the maximum distance between nodes of the two sets is minimized. Gajer's approach simplifies graph coarsening by using *vertex filtration* [4]. By restricting the number of nodes considered in relocating any node in the filtration of $O(\log|V|)$ levels, the approach applies formulae in [8] to reduce

Please cite this article in press as: A. Chonbodeechalermroong, R. Hewett, Towards Visualizing Big Data with Large-Scale Edge Constraint Graph Drawing, Big Data Res. (2017), https://doi.org/10.1016/j.bdr.2017.10.001