Faster Fruchterman–Reingold Algorithm by Random Subspace Method

Hiroki Hamaguchi ⬤

Abstract—The abstract goes here.

Index Terms—Graph drawing, Fruchterman–Reingold algorithm, Random Subspace method.

I. Introduction

GRAPH is a mathematical structure representing pairwise relationships between objects, and graph drawing is one of the most fundamental tasks in data science. Numerous general algorithms have been proposed for graph drawing. For example, Sugiyama et al. proposed a layered graph drawing [1], while Davidson and Harel introduced an algorithm using simulated annealing [2]. Among these, one of the most popular strategies is force-directed algorithms.

In force-directed algorithms, the graph is modeled as a physical system of particles. These include methods such as the Tutte embedding [3] for planar graphs, the Kamada–Kawai algorithm [4] using vertex distance as a cost function, and the Fruchterman–Reingold (FR) algorithm [5], which is the primary focus of this paper.

The FR algorithm is one of the most widely used force-directed algorithms and is implemented in many modern graph drawing libraries such as NetworkX [6] and igraph [7]. The algorithm is based on a physical model of a system of particles and springs, positioning vertices so that edges are neither too long nor too short.

However, the FR algorithm is tend to be captured in local minima and has a high computational cost of $\mathcal{O}(n^2)$, where n is the number of vertices in the graph. To address this kind of computational burden, several methods have been proposed. One strategy is to approximate the n-body simulation using hierarchical methods such as the fast multipole method [8], the Barnes–Hut
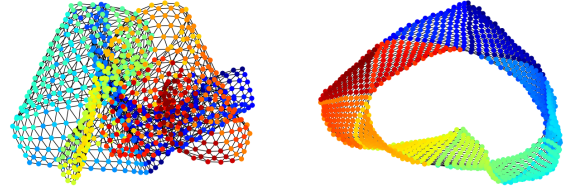

Fig. 2: Comparison of the FR algorithm (left, NetworkX) and L-BFGS-B (right, my implementation) on the `jagmesh1` dataset.

approximation [9], multilevel approaches [10], or stress majorization [11].

Another approach is to accelerate the optimization algorithm directly, which aligns with the spirit of our work. Recent advances have accelerated the FR algorithm through various methods, including GPU parallel architectures [12], numerical optimization techniques such as L-BFGS-B [13], and Stochastic Gradient Descent (SGD) [14].

In this paper, we propose a new algorithm to calculate the FR model more efficiently, drawing heavily on the Random Subspace method [15], [16], [17], [18].

The rest of the paper is organized as follows. In Section II, we define the optimization problem for the FR algorithm. In Section III, we introduce the Random Subspace method. In Section IV, we present the experimental results. Finally, we conclude the paper in Section V.

II. Preliminary

In this section, we define an optimization problem for FR algorithm. Let $\mathbb{R}_+$ be a set of non-negative real numbers, $\mathbb{R}_{++}$ be a set of positive real numbers, and $A = (a_{i,j}) \in \mathbb{R}_+^{n \times n}$ be an adjacency matrix of a undirected graph $G = (V, E)$. Each vertex $v_i \in V$ is assigned a position $x_i \in \mathbb{R}^d$ and we define $x = (x_1, \ldots, x_n) \in \mathbb{R}^{d \times n}$ as a matrix of positions. For an optimal distance k, and a distance d between two vertices v_i and v_j, Fruchterman and Reingold defined the power
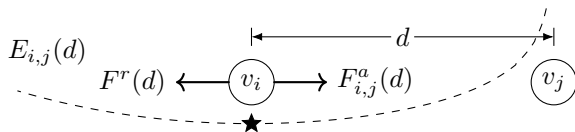

Fig. 1: Fruchterman–Reingold model. The equilibrium of the attractive force $F_{i,j}^a(d)$ and the repulsive force $F^r(d)$ is the optimal distance $d = k/\sqrt[3]{a_{i,j}}$.

of attraction $F_{i,j}^a : \mathbb{R}_+ \to \mathbb{R}$ and the power of repulsion $F^r : \mathbb{R}_{++} \to \mathbb{R}$ as

$$F_{i,j}^a(d) := \frac{a_{i,j}d^2}{k}, \quad F^r(d) := -\frac{k^2}{d}.$$

The total energy for these powers $E_{i,j} : \mathbb{R}_{++} \to \mathbb{R}$, stress of the graph, is defined as

$$E_{i,j}(d) := \int_0^d F_{i,j}^a(r)\,\mathrm{d}r + \int_\infty^d F^r(r)\,\mathrm{d}r$$
$$= \frac{a_{i,j}d^3}{3k} - k^2 \log d.$$

The energy function $E_{i,j}$ is convex for $a_{i,j} \in \mathbb{R}_{++}$ and minimized when $d = k/\sqrt[3]{a_{i,j}}$, but is not Lipschitz continuous. Based on these energies, the optimization problem for FR algorithm is defined as

$$\begin{array}{c} \text{minimize} \\ x \in \mathbb{R}^{d \times n} \end{array} \quad f(x) := \sum_{i<j} E_{i,j}(d_{i,j}) \qquad (1)$$

where $d_{i,j} = \|x_i - x_j\|$ is the Euclidean distance.

In order to introduce the Random Subspace method, we define a function $f_i : \mathbb{R}^d \to \mathbb{R}$ for $1 \le i \le n$ as

$$f_i(x_i) := \sum_{j \neq i} E_{i,j}(d_{i,j}).$$

The gradient and the Hessian of f_i are

$$\nabla f_i(x_i) = \sum_{j \neq i} \left(\frac{a_{i,j}d_{i,j}}{k} - \frac{k^2}{d_{i,j}^2} \right)(x_i - x_j),$$

$$\nabla^2 f_i(x_i) = \sum_{j \neq i} \left(\frac{a_{i,j}d_{i,j}}{k} - \frac{k^2}{d_{i,j}^2} \right) I_d +$$
$$\sum_{j \neq i} \left(\frac{a_{i,j}}{kd_{i,j}} + \frac{2k^2}{d_{i,j}^4} \right)(x_i - x_j)(x_i - x_j)^\top.$$

As pointed out in [19], the FR algorithm is a gradient descent method for the energy function f.

III. Algorithm

For $t \in [0, 1)$, let define as

$$E_{i,j}^t(d) = \frac{a_{i,j}d^3}{3k} - \frac{k^2 d^{1-t} - 1}{1 - t}.$$

Note that

$$\lim_{t \to 1} -\frac{k^2 d^{1-t} - 1}{1 - t} = -k^2 \log d.$$

by L'Hôptial's rule.

$$f_i^t(x_i) := \sum_{j \neq i} E_{i,j}^t(d_{i,j}).$$

The gradient and the Hessian of f_i are

$$\nabla f_i^t(x_i) = \sum_{j \neq i} \left(\frac{a_{i,j}d_{i,j}}{k} - \frac{k^2}{d_{i,j}^{1+t}} \right)(x_i - x_j),$$

$$\nabla^2 f_i^t(x_i) = \sum_{j \neq i} \left(\frac{a_{i,j}d_{i,j}}{k} - \frac{k^2}{d_{i,j}^{1+t}} \right) I_d +$$
$$\sum_{j \neq i} \left(\frac{a_{i,j}}{kd_{i,j}} + \frac{k^2(1 + t)}{d_{i,j}^{3+t}} \right)(x_i - x_j)(x_i - x_j)^\top.$$

IV. Experiment

We used dataset from [20] and MatrixMarket [21]. https://reference.wolfram.com/language/tutorial/GraphDrawingIntroduction.html

V. Conclusion

todo

References

[1] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical System Structures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 2, pp. 109–125, Feb. 1981.

[2] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," *ACM Transactions on Graphics (TOG)*, vol. 15, no. 4, pp. 301–331, 1996.

[3] W. T. Tutte, "How to Draw a Graph," *Proceedings of the London Mathematical Society*, vol. s3-13, no. 1, pp. 743–767, 1963.

[4] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, Apr. 1989.

[5] 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.

[6] A. Hagberg, P. J. Swart, and D. A. Schult, "Exploring network structure, dynamics, and function using NetworkX," Jan. 2008.

[7] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006.

[8] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, Dec. 1987.

[9] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, Dec. 1986.

[10] Y. Hu, "Efficient, high-quality force-directed graph drawing," *The Mathematica journal*, vol. 10, pp. 37–71, 2006.

[11] E. R. Gansner, Y. Koren, and S. North, "Graph Drawing by Stress Majorization," in *Graph Drawing*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and J. Pach, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, vol. 3383, pp. 239–250.

[12] P. Gajdoš, T. Ježowicz, V. Uher, and P. Dohnálek, "A parallel Fruchterman–Reingold algorithm optimized for fast visualization of large graphs and swarms of data," *Swarm and Evolutionary Computation*, vol. 26, pp. 56–63, Feb. 2016.

[13] H. Hosobe, "Numerical optimization-based graph drawing revisited," in *2012 IEEE Pacific Visualization Symposium*, 2012, pp. 81–88.

[14] J. X. Zheng, S. Pawar, and D. F. M. Goodman, "Graph drawing by stochastic gradient descent," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 9, pp. 2738–2748, 2019.

[15] R. Gower, D. Kovalev, F. Lieder, and P. Richtarik, "RSN: Randomized subspace newton," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[16] T. Fuji, P.-L. Poirion, and A. Takeda, "Randomized subspace regularized Newton method for unconstrained non-convex optimization," Sep. 2022.

[17] C. Cartis, J. Fowkes, and Z. Shao, "Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares," Nov. 2022.

[18] R. Higuchi, P.-L. Poirion, and A. Takeda, "Fast Convergence to Second-Order Stationary Point through Random Subspace Optimization," Jun. 2024.

[19] D. Tunkelang, "A numerical optimization approach to general graph drawing," Ph.D. dissertation, Carnegie Mellon University, 1999.

[20] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Transactions on Mathematical Software (TOMS)*, vol. 38, no. 1, pp. 1–25, 2011.

[21] R. F. Boisvert, R. Pozo, K. Remington, R. F. Barrett, and J. J. Dongarra, "Matrix Market: A web resource for test matrix collections," in *Quality of Numerical Software: Assessment and Enhancement*, R. F. Boisvert, Ed. Boston, MA: Springer US, 1997, pp. 125–137.