Fruchterman–Reingold Layout with Subspace Method as Initial Placement

Hiroki Hamaguchi Naoki Marumo Akiko Takeda

***Abstract*—日本語で書かれている場合、執筆中を表す。オレンジの文章は正確性が疑問視される記述や TODO な記述。また、orcidlink が壊れているので author のところで暫定措置を取っているが、これは日本語を全削除して dvipdfmx を取り除けば ok。Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.**

***Index Terms*—Graph Drawing, Optimization, Fruchterman–Reingold Layout, Random Subspace Method.**

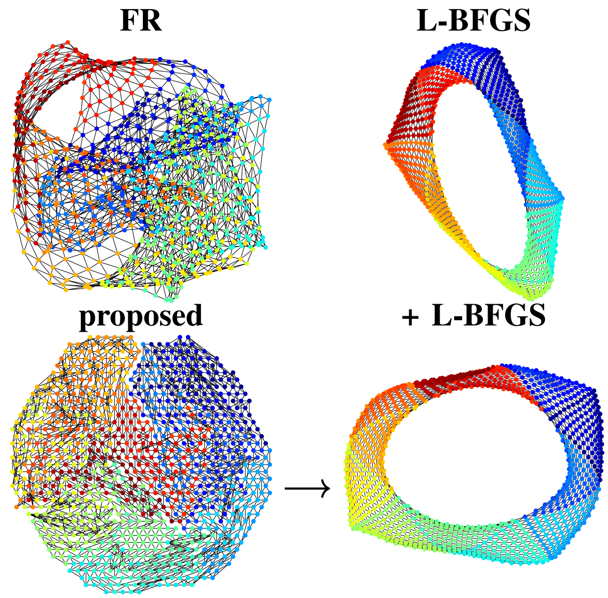

Fig. 1: Comparison of the FR algorithm, L-BFGS, and the proposed method for the `jagmesh1` dataset.

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. Indeed, numerous kinds of algorithms have been proposed for graph drawing [1], [2], [3], [4]. Among these, one of the most popular strategies is force-directed algorithms.

In force-directed algorithms, we model a graph as a system of particles with forces acting between them. This class of algorithms includes the Kamada–Kawai (KK) layout [5], Eades' spring embedder [**?**], and the Fruchterman–Reingold (FR) layout [6], [7], which serves as the central focus of this study.

The FR algorithm is one of the most widely used force-directed algorithms. It is also implemented in many modern graph drawing libraries such as NetworkX [8], Graphviz [9], and igraph [10].

However, both the KK layout and the FR layout suffer from high computational complexity, specifically $\mathcal{O}\left(|V|^2\right)$ per iteration as it is, where $|V|$ denotes the number of vertices. This computational burden makes it difficult to apply these algorithms to large-scale graphs.

To address this kind of burden, several methods have been developed. These include approximating the n-body simulation using multipole expansions [11] or the Barnes–Hut approximation [12], gradually refining the layouts using a multilevel approach [13] and employing stress majorization [14].

Another approach is to directly accelerate the optimization process, which aligns with the aim of our work. Recent researches have accelerated the algorithms for FR layout or KK layout through various methods, including adaptation to GPU parallel architectures [15], utilizing numerical optimization techniques such as L-BFGS [16], and Stochastic Gradient Descent (SGD) [17].

Based on such advances, in this paper, we investigate the ability of another algorithm: subspace methods. Subspace methods are a class of optimization algorithms that restrict the optimization to a lower-dimensional subspace of the solution space, which can significantly reduce the computational cost per iteration. We propose a new algorithm for the FR layout that leverages the subspace method, and we demonstrate its effectiveness through experiments.

The rest of the paper is organized as follows. In Section II, we define the optimization problem for the FR algorithm. In Section **??**, we present our research question based on previous works. In Section III, we propose a new algorithm that utilizes the subspace method for the FR layout. In Section IV, we present our experimental results. Finally, we conclude and discuss future work in Section VI.

II. Preliminary

In this section, we formulate the FR layout as a continuous optimization problem, and introduce the conventional

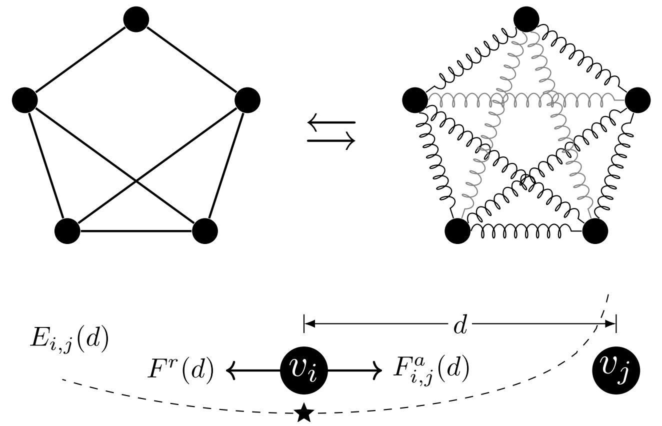

Fig. 2: (Top) Fruchterman–Reingold layout. It models $\mathcal{O}(n^2)$ springs between all pairs of vertices. (Bottom) The equilibrium between attractive force $F^a_{i,j}(d)$ and repulsive force $F^r(d)$ is achieved at $d = k/\sqrt[3]{w_{i,j}}$, which equals k when $w_{i,j} = 1$.

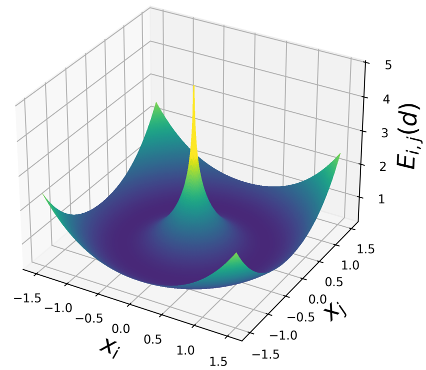

Fig. 3: Energy function $E_{i,j}(d)$ for $x_j = (0, 0)$, $w_{i,j} = 1$ and $k = 1$. Although $E_{i,j}$ is convex as a function of d, it is not convex as a function of x_i. As x_i approaches x_j, the energy function diverges.

approaches to this problem, namely the FR algorithm and the L-BFGS method.

A. Fruchterman–Reingold layout

Let $\mathbb{R}_{>0} \coloneqq \{x \in \mathbb{R} \mid x > 0\}$, $\mathbb{R}_{\geq 0} \coloneqq \{x \in \mathbb{R} \mid x \geq 0\}$, and let $W = (w_{i,j}) \in \mathbb{R}^{n\times n}_{\geq 0}$ be an adjacency matrix of a graph $G_W = (V, E)$, where $V = \{v_i \mid 1 \leq i \leq n\}$ is a set of vertices and $E = \{(v_i, v_j) \mid w_{i,j} > 0\}$ is a set of edges. We call $w_{i,j}$ as a weight of the edge (v_i, v_j).

We will only consider undirected connected graphs with non-negative weights. Although the FR algorithm in NetworkX, for example, can handle directed unconnected graphs with negative weights, this paper does not focus on such cases. For directed graphs, slight modifications of algorithms or converting them to undirected graphs may be effective. For unconnected graphs, algorithms can be applied to each connected component independently. When negative weights are present, the optimization problem may become unbounded, but with non-negative weights and ensuring the graph's connectivity, the problem is always bounded and solvable. In summary, the conditions for W is formulated as follows:

$$W \in \mathbb{R}^{n\times n}_{\geq 0}, \quad W = W^\top, \quad G_W \text{ is connected.} \tag{1}$$

Fruchterman and Reingold [6] proposed a force-directed layout called the Fruchterman–Reingold (FR) layout, as known as a spring layout [8]. Let $x_i \in \mathbb{R}^2$ be the position of the vertex $v_i \in V$, and $X = (x_1, \ldots, x_n) \in \mathbb{R}^{2\times n}$ be the configuration of the graph. For a parameter k and a distance $d_{i,j} \coloneqq \|x_i - x_j\|_2$ between two vertices v_i and v_j, the attraction force $F^a_{i,j} : \mathbb{R}_{>0} \to \mathbb{R}$ and the repulsion force $F^r : \mathbb{R}_{>0} \to \mathbb{R}$ is defined as

$$F^a_{i,j}(d) \coloneqq \frac{w_{i,j} d^2}{k}, \quad F^r(d) \coloneqq -\frac{k^2}{d}.$$

FR layout seeks the equilibrium of the forces between all pairs of vertices, as shown in Figure 2.

We can also interrupt forces by its scalar potential [16], in other words, energy $E_{i,j} : \mathbb{R}_{>0} \to \mathbb{R}$, which is defined by

$$\begin{aligned} E_{i,j}(d) &\coloneqq \int_0^d F^a_{i,j}(r)\,\mathrm{d}r + \int_\infty^d F^r(r)\,\mathrm{d}r \\ &= \frac{w_{i,j} d^3}{3k} - k^2 \log d. \end{aligned} \tag{2}$$

As a remark, this energy function $E_{i,j}$ is convex as a function of d and minimized when $d^* = k/\sqrt[3]{w_{i,j}}$, but it is not Lipschitz continuous and is not convex as a function of x_i. Refer to Figure 3.

Now, the optimization problem for FR layout can be formulated as the minimization of the energy function $f : \mathbb{R}^{2\times n} \to \mathbb{R}$, as known as a stress of the graph:

$$\underset{X \in \mathbb{R}^{2\times n}}{\text{minimize}} \quad f(X) \coloneqq \sum_{i<j} E_{i,j}(d_{i,j}) \tag{3}$$

This is because seeking an equilibrium of the forces is equivalent to minimizing the energy function $E_{i,j}$ for all pairs of vertices. In the following, we will discuss the optimization of this problem.

III. Proposed Algorithm

Based on the research question above, we propose a new algorithm for the FR layout that leverages the subspace method. 我々の提案手法は三段階に分けて説明される。Firstly,III-A 節で、最適化問題 (3) を変形し離散最適化問題として hexagonal lattice を用いた簡略化された定式化を行う。Secondly,III-B 節で、Random に選ばれた頂点についての Newton's direction を用いて、その離散最適化問題を連続緩和によって解く方法を示す。Thirdly,III-C 節で、上記の解を初期解として、最適化問題 (3) を FR algorithm や L-BFGS method で解くという、提案手法の全体像を示す。

A. reduction to the discrete optimization problem

まず、最適化問題 (3) を制約付き連続最適化問題へと変形する。式 (2) に示した $E_{i,j}$ は、$w_{i,j} = 0$ の時に $-k^2 \log d_{i,j}$

となり、実用上多くのグラフが満たす疎な性質 ($|E| \ll |V|^2$) も考慮すると、以下のように式変形できる:

$$\underset{X \in \mathbb{R}^{2\times n}}{\text{minimize}} \quad f(X) = \sum_{(i,j)\in E} \frac{w_{i,j} d_{i,j}^3}{3k} - \sum_{i<j} k^2 \log d_{i,j} \tag{4}$$

この最適化問題 (4) は、次のように近似すれば、$|V|^2$ から $|E|$ の計算量で目的関数の値を求めることができる:

$$\begin{aligned} \underset{X \in \mathbb{R}^{2\times n}}{\text{minimize}} \quad & \sum_{(i,j)\in E} \frac{w_{i,j} d_{i,j}^3}{3k} \\ \text{subject to} \quad & d_{i,j} \geq \epsilon, \quad \forall (i,j)\,(i<j) \end{aligned} \tag{5}$$

ただし、ϵ は適当に定めた正の定数である。この近似は、$-k^2 \log d_{i,j}$ が $d_{i,j}$ に対して単調減少な凸関数であることから、$d_{i,j}$ が一定以上の時には、$-k^2 \log d_{i,j}$ の値が極端に大きくなることは無く、ある程度の妥当性を持つ。

しかし、依然問題 (5) は $\mathcal{O}\left(|V|^2\right)$ の制約を持ってしまい、目的関数が $\mathcal{O}(|E|)$ の計算量で求められる利点が生かせていない為、更なる簡略化を行う。ここで、FR layout に対する固定された初期配置の考え方 [4] を援用する。この研究では、円環状の初期配置と Simulated Annealing(SA) を用いることで、最適化問題の解を得るための初期解を高速に得ることができると報告されている。同様に、初期配置を求めるという前提で問題 (5) を簡略化すると、以下のような離散最適化問題を得ることができる:

$$\begin{aligned} \underset{\pi : V \to Q}{\text{minimize}} \quad & \sum_{(i,j)\in E} \frac{w_{i,j} d_{i,j}^3}{3k} \\ \text{subject to} \quad & x_i = \pi(v_i), \quad \forall v_i \in V, \\ & \|q_i - q_j\|_2 \geq \epsilon, \quad \forall q_i, q_j \in Q\,(q_i \neq q_j), \\ & \pi(v_i) \neq \pi(v_j), \quad \forall v_i, v_j \in V\,(v_i \neq v_j) \end{aligned} \tag{6}$$

つまり、頂点 V から、離散的な点配置 Q への単射 π を最適化する問題として定式化される。点配置を予め固定しておけば、$\mathcal{O}\left(|V|^2\right)$ 個の制約を一々確認する必要がなくなるため、計算量が $\mathcal{O}(|E|)$ に削減され、高速化が期待される。離散最適化問題を持ち出すのには、ここに理由がある。

この離散的な点配置 Q としては、例えば n 個の半径 ϵ の円を $\mathbb{R}^2$ 上に充填させたものなどが考えられるが、本研究では六方格子を用いることにした。六方格子は、空間における最密充填構造の一つであり、また計算が容易であるという利点がある。なお、先行研究 [27] でも、FR layout と六方格子の関連性が指摘されているが、本研究とはあまり関係はない。

B. Newton's direction for discrete optimization

次に、Random に選ばれた頂点についての Newton's direction を用いて、その離散最適化問題を連続緩和によって解く。この Newton's direction を求める部分は、本質的には節 ?? で述べた RSN method と同一である。

Random に選ばれた頂点 v_i に対する目的関数 $f_i(x_i)$ の Hessian matrix は、ならし時間計算量 $\mathcal{O}(|E|/n)$ で計算でき、またその逆行列は 2×2 行列の為、計算量は $\mathcal{O}(1)$ である。

そして、そのようにして求めた Newton's direction を用いて、離散最適化問題を連続緩和によって解くことができる。具体的には、Newton's direction による update 後の頂点の位置を求め、それを離散的に配置されている六方格子上の最近傍点に投影し、その点を新たな頂点の位置とする。また、元の位置から新たな頂点まで順に頂点を動かすことにより、離散最適化問題の制約を満たしながら解くことが出来る。図 ??にその概要を示したので、参照されたい。

以上により、最適化問題 (3) を解くための良質な初期解を得ることができる。

C. pseudo code

The implementation about hexagonal grid is based on [28].

Algorithm 1: Random Subspace Newton for Fruchterman–Reingold layout

```
Input: Graph G_W = (V, E), subspace dimension s
Output: Point configuration X = (x_1, ..., x_n)
define parameters k, t, dt, iterations;
x_i ← random for all v_i ∈ V;
for j ← 0 to iterations do
    compute gradient ∇f_i(x_i) for all v_i ∈ V;
    x_i ← x_i − (P^⊤ ∇^2 f_i(x_i) P)^{-1} P^⊤ ∇f_i(x_i) for all
      v_i ∈ V;
    t ← t − dt;
    if convergence condition then
        break;
return pos
```

IV. Numerical Experiment

We used dataset from [29] and MatrixMarket [30].
https://reference.wolfram.com/language/tutorial/GraphDrawingIntroduction.html

A. 網羅的な実験結果

B. 詳細な実験結果

V. Combination with Other Techniques

頂点の縮約。sfdp. それによって、更に多変数の問題を解くことができる可能性がある。

VI. Discussion

In this paper, we investigated the effectiveness of the Random Subspace Newton method for the Fruchterman–Reingold algorithm.

A. Application to Other Problems

graph isomorphic problem, graph matching problem, graph embedding problem, etc.

VII. Acknowledgment

The author would like to express our sincere gratitude to PL Poirion and Andi Han for their insightful discussions, which have greatly inspired and influenced this research.

References

[1] W. T. Tutte, "How to Draw a Graph," *Proceedings of the London Mathematical Society*, vol. s3-13, no. 1, pp. 743–767, 1963. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1112/plms/s3-13.1.743

[2] M. Chrobak and T. H. Payne, "A linear-time algorithm for drawing a planar graph on a grid," *Information Processing Letters*, vol. 54, no. 4, pp. 241–246, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002001909500020D

[3] 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, 1981. [Online]. Available: https://ieeexplore-ieee-org.utokyo.idm.oclc.org/document/4308636

[4] F. Ghassemi Toosi, N. S. Nikolov, and M. Eaton, "Simulated Annealing as a Pre-Processing Step for Force-Directed Graph Drawing," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '16 Companion. Association for Computing Machinery, 2016, pp. 997–1000. [Online]. Available: https://dl.acm.org/doi/10.1145/2908961.2931660

[5] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019089901026

[6] 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. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102

[7] S. G. Kobourov, "Spring Embedders and Force Directed Graph Drawing Algorithms," 2012. [Online]. Available: http://arxiv.org/abs/1201.3011

[8] A. Hagberg, P. J. Swart, and D. A. Schult, "Exploring network structure, dynamics, and function using NetworkX," 2008. [Online]. Available: https://www.osti.gov/biblio/960616

[9] J. Ellson, E. Gansner, L. Koutsofios *et al.*, "Graphviz— Open Source Graph Drawing Tools," in *Graph Drawing*, P. Mutzel, M. Jünger, and S. Leipert, Eds. Springer, 2002, pp. 483–484.

[10] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006. [Online]. Available: https://igraph.org

[11] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999187901409

[12] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation algorithm," *Nature*, vol. 324, no. 6096, pp. 446–449, 1986. [Online]. Available: https://www.nature.com/articles/324446a0

[13] Y. Hu, "Efficient, high-quality force-directed graph drawing," *The Mathematica journal*, vol. 10, pp. 37–71, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:14599587

[14] E. R. Gansner, Y. Koren, and S. North, "Graph Drawing by Stress Majorization," in *Graph Drawing*, D. Hutchison, T. Kanade, J. Kittler *et al.*, Eds. Springer Berlin Heidelberg, 2005, vol. 3383, pp. 239–250. [Online]. Available: http://link.springer.com/10.1007/978-3-540-31843-9_25

[15] 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, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650215000644

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

[17] 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.

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

[19] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, 1989. [Online]. Available: https://doi.org/10.1007/BF01589116

[20] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[21] Y. Qiu, "Yixuan/LBFGSpp," 2024. [Online]. Available: https://github.com/yixuan/LBFGSpp

[22] N. Okazaki, "Chokkan/liblbfgs," 2024. [Online]. Available: https://github.com/chokkan/liblbfgs

[23] 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 *et al.*, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bc6dc48b743dc5d013b1abaebd2faed2-Paper.pdf

[24] T. Fuji, P.-L. Poirion, and A. Takeda, "Randomized subspace regularized Newton method for unconstrained non-convex optimization," 2022. [Online]. Available: http://arxiv.org/abs/2209.04170

[25] C. Cartis, J. Fowkes, and Z. Shao, "Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares," 2022. [Online]. Available: http://arxiv.org/abs/2211.09873

[26] R. Higuchi, P.-L. Poirion, and A. Takeda, "Fast Convergence to Second-Order Stationary Point through Random Subspace Optimization," 2024. [Online]. Available: http://arxiv.org/abs/2406.14337

[27] J. Li, Y. Tao, K. Yuan *et al.*, "Fruchterman–reingold hexagon empowered node deployment in wireless sensor network application," *Sensors*, vol. 22, no. 5179, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5179

[28] A. J. Patel, "Hexagonal Grids," Red Blob Games, Tech. Rep., 2013. [Online]. Available: https://www.redblobgames.com/grids/hexagons/

[29] 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.

[30] R. F. Boisvert, R. Pozo, K. Remington *et al.*, "Matrix Market: A web resource for test matrix collections," in *Quality of Numerical Software: Assessment and Enhancement*, R. F. Boisvert, Ed. Springer US, 1997, pp. 125–137. [Online]. Available: https://doi.org/10.1007/978-1-5041-2940-4_9

Appendix A
Optimal Scaling

When we optimize a placement for FR-layout with an initial placement obtained, for instance through KK-layout, scaling the initial placement at first can often yield better results than directly using the unmodified initial placement. In this section, we address the problem of finding the optimal scaling factor that minimizes the energy function for a given configuration.

A. Optimal Scaling Algorithm

Formulating the optimization through scaling, the task reduces to selecting an appropriate scaling factor $x \in \mathbb{R}_{>0}$ that minimizes the following energy function:

$$\begin{aligned}
\phi(x) &:= \left(\sum_{i<j} \frac{w_{ij}(x d_{ij})^3}{3k} \right) - k^2 \sum_{i<j} \log(x d_{ij}) \\
&= x^3 \left(\sum_{i<j} \frac{w_{ij} d_{ij}^3}{3k} \right) - \log(x)(k^2 n(n-1)) \\
&\quad - k^2 \sum_{i<j} \log(d_{ij})
\end{aligned}$$

$$\phi'(x) = 3x^2 \left(\sum_{i<j} \frac{w_{ij} d_{ij}^3}{3k} \right) - \frac{k^2 n(n-1)}{x}$$

$$\phi''(x) = 6x \left(\sum_{i<j} \frac{w_{ij} d_{ij}^3}{3k} \right) + \frac{k^2 n(n-1)}{x^2}$$

The function $\phi(x)$ is convex, and we can find the optimal scaling factor x by using Newton's method. This algorithm achieves sufficient convergence within a few iterations, and when we pre-compute the coefficients of $\phi(x)$ with $w_{i,j} > 0$, the time complexity is just $\mathcal{O}(|E|)$.

Appendix B
Challenges of the RSN for the FR Algorithm

In this study, we proposed utilizing the subspace method as an initial placement. Then, a natural question can be arose: Can the subspace method alone achieve "fast" optimization throughout the entire process? Specifically, we want to investigate whether the subspace method can optimize the positions of all vertices efficiently without the constraint of the hexagonal lattice.

To address this, we consider the following algorithm as a natural extension and application of the random subspace methods. Namely, we randomly select a vertex v_i, apply Newton's method to f_i using Eq. **??** and its Hessian:

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

Then, we update the position of vertex v_i, and repeat this process until convergence. However, this approach fails to work effectively in practice.

We have reached a tentative conclusion that achieving "fast" optimization using the subspace algorithm alone is unlikely. This section outlines some of the reasons behind this negative outcome.

It is important to note that these challenges do not necessarily imply fundamental limitations of the subspace method. On the contrary, improvements based on these identified issues could potentially enhance the effectiveness of the subspace method.

A. *Ignorance of other vertex movements*

L-BFGS を始めとする、問題全体のヘッシアンを考慮する手法は、ある頂点の位置を更新する際に、他の頂点の動きも考慮すると言える。ここでは、それがどのような意味を持つか論ずる。

B. *Inaccuracy of quadratic approximation*

まず、前提として、非凸であり、cubic regularized Newton method をはじめとする正則化の追加が求められる。しかし、そのような正則項

二次近似が著しく悪くなる場合がある。しかし、subspace に限定すると、特に問題が生じやすくなる。その一例が Fig. 4 で示すような状況である。

However, unfortunately, the RSN method is not necessarily effective for the FR layout. As shown in Figure 4, although the energy function f_i with respect to the position x_i of each vertex is convex in the FR layout, the overall energy function f with respect to x_i is not convex.

これに対する解決策として、line search の実施などである。しかし、そもそもの Newton's direction が最適解から大きく外れるという問題は、必ずしも解決できない可能性があることには注意されたい。

Appendix C
Another approach based on the Proposed Method

本論文では六方格子に基づいた Subspace Method を提案したが、基本的に同一のアイデアに基づいた他のアプローチも考えられる。

A. *Non-randomized approach*

一つは、各頂点毎に最適化する際に、ランダムに頂点を選んで最適化していくのではなく、$|V|$ 点全てを同時に最適化する方法である。

こうすることによって、六方格子という制約をよりラフに扱うことが出来る。

六方格子への射影は、MinCostFlow などで $\mathcal{O}\left(|V|^3\right)$ で厳密解が出せる。ソートによる擬似的な射影で、$\mathcal{O}(|V| \log |V|)$ で近似解が出せる。

B. *Non-Newton approach*

もう一つは、Newton 法を使わずに、勾配法を使う方法である。

以上のいずれも、基本的に性能は落ちると考えているが、実装が簡略化するなどの利点や、イテレーション毎の計算コストが多少減るという利点があり、検討の余地が残されている。

Hiroki Hamaguchi Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan.

Naoki Marumo Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan.

Akiko Takeda Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan. Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan.

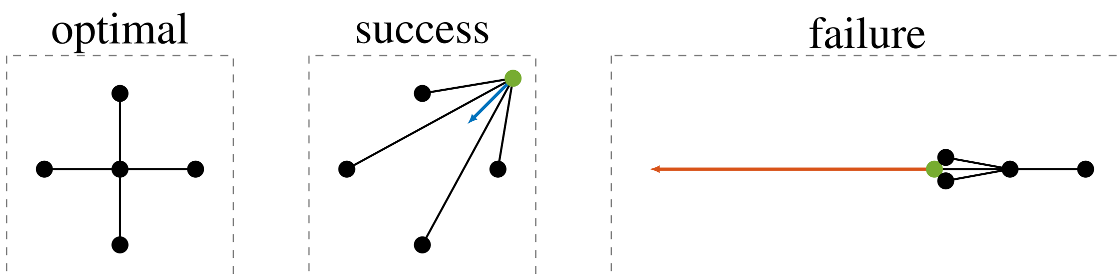

Fig. 4: Visualization of the problem of the subspace method. Let a graph as shown in the left (optimal), where k and all positive edge weights $w_{i,j}$ are set to 1. For the situation in the middle (success), the RSN works effectively. However, in the situation depicted on the right (failure), where the points are set as $x_0 = (0, 0), x_1 = (-1, 0), x_2 = (-0.85, 0.155), x_3 = (-0.85, -0.155), x_4 = (1, 0)$, the Hessian for the subspace of x_1 is approximately $\begin{pmatrix} 1.841 & 0 \\ 0 & 1.159 \end{pmatrix}$. This Hessian, while positive definite and not ill-conditioned, leads to a Newton direction that clearly deviates significantly from the global optimal solution.