予定

- 9 月 17 日 10:00 a.m- ゼミ室 6-238
- 前 30 分: 今までしてきたことの説明
- 中 30 分: スライド末尾に記した、研究の方向性についてのご相談
    - 勝手にこの方向性に進んできてしまった感があるが、この方向性でよいか?
- 後 30 分: Yes の場合はアドバイスをもらいたい点が山ほどあるのでその議論、No の場合はどのような方向性がよいかの議論
- 数理輪講は 11/29 このスライドは Yes の場合に一部使用予定

Improved Initial Placement for Fruchterman–Reingold Algorithm

Hiroki Hamaguchi
Supervisors: Prof. Akiko Takeda
Pierre-Louis Poirion
Andi Han
if ok, Prof. Naoki Marumo

2024/11/29

Introduction for Graph Drawing

Graph Drawing

Discrete Based

BFS layout
- for tree graph

Planar layout
- for planar graph
- [Tutte, 1963]
- [Chrobak and Payne, 1995]

Layered graph drawing
- for DAG
- [Sugiyama et al., 1981]

Spectral layout
- eigenvector of Laplacian

Continuous Based

Kamada–Kawai (KK)
[Kamada and Kawai, 1989]

$$\text{minimize} \quad \sum_{i<j} \frac{k_{i,j}}{2}(d_{i,j} - l_{i,j})^2$$

Fruchterman–Reingold (FR)
[Fruchterman and Reingold, 1991]

$$\text{minimize} \quad \sum_{i<j} \left(\frac{a_{i,j} d_{i,j}^3}{3k} - k^2 \log d_{i,j} \right)$$

$d_{i,j}$: distance between nodes v_i, v_j / $l_{i,j}$: optimal distance / $k, a_{i,j}$: constant

NetworkX and Graphviz support KK and FR

Definition of FR Layout

- Historically, FR layout seeks an **equilibrium of two forces**:

$$F^a_{i,j}(d) := \frac{a_{i,j} d^2}{k} \quad \text{(attraction)}, \quad F^r(d) := -\frac{k^2}{d} \quad \text{(repulsion)}$$

- This problem can be redefine as an optimization problem:

$$\text{minimize} \sum_{i<j} E_{i,j}(d) \quad \text{where} \quad E_{i,j}(d) := \int_0^d F^a_{i,j}(r)\,\mathrm{d}r + \int_\infty^d F^r(r)\,\mathrm{d}r = \frac{a_{i,j} d^3}{3k} - k^2 \log d$$

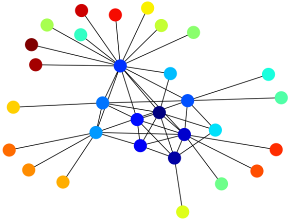
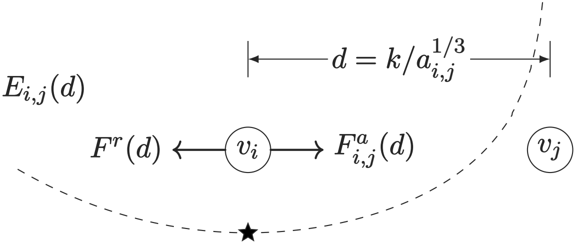


Figure: Example of FR layout



Algorithm for FR Layout 1

“spring_layout” in NetworkX and “fdp” in Graphviz

- Adaptive cooling scheme
- **gradient descent method** with constant step size per each vertex
- strong **theoretical background**
- [Tunkelang, 1999]

Algorithm 2: Fruchterman-Reingold

```
area ← W ∗ L ;                                   /* frame: width W and length L */
initialize G = (V, E) ;                          /* place vertices at random */
k ← √(area/|V|) ;                                /* compute optimal pairwise distance */
function f_r(x) = k^2/x ;                        /* compute repulsive force */
for i = 1 to iterations do
    foreach v ∈ V do
        v.disp := 0; ;                           /* initialize displacement vector */
        for u ∈ V do
            if (u ≠ v) then
                Δ ← v.pos − u.pos ;              /* distance between u and v */
                v.disp ← v.disp + (Δ/|Δ|) ∗ f_r(|Δ|) ;   /* displacement */
    function f_a(x) = x^2/k ;                    /* compute attractive force */
    foreach e ∈ E do
        Δ ← e.v.pos − e.u.pos ;                  /* e is ordered vertex pair .v and .u */
        e.v.disp ← e.v.disp − (Δ/|Δ|) ∗ f_a(|Δ|);
        e.u.disp ← e.u.disp + (Δ/|Δ|) ∗ f_a(|Δ|);
    foreach v ∈ V do
                    /* limit max displacement to frame; use temp. t to scale */
        v.pos ← v.pos + (v.disp/|v.disp|) ∗ min(v.disp, t);
        v.pos.x ← min(W/2, max(−W/2, v.pos.x));
        v.pos.y ← min(L/2, max(−L/2, v.pos.y));
    t ← cool(t) ;                                /* reduce temperature for next iteration */
```

Figure: [Kobourov, 2012]

Algotithm for FR Layout 2

"sfdp" in Graphviz

- Scalable Force-Directed Placement
- **Multilevel** approach
- Barnes–Hut algorithm (Q-tree)[Hu, 2006] [Barnes and Hut, 1986]
- These methods are **out of scope**, but our work is still **applicable**.

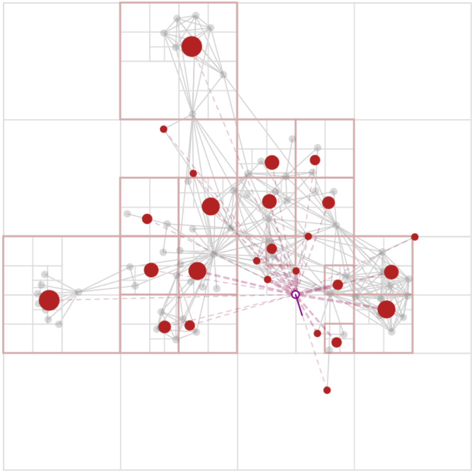


Figure: `https://jheer.github.io/barnes-hut/`

Hardness of graph drawing

- FR algorithm is powerful for network-like graphs, but **not for other types**.
- Escaping from a plateau is **very difficult**.
- Drawing ◯ ≅ untangling tangled earphones

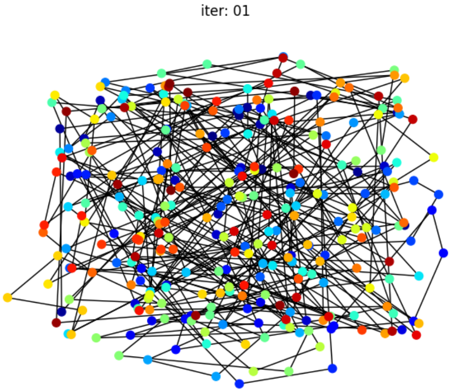


Figure: FR layout by NetworkX

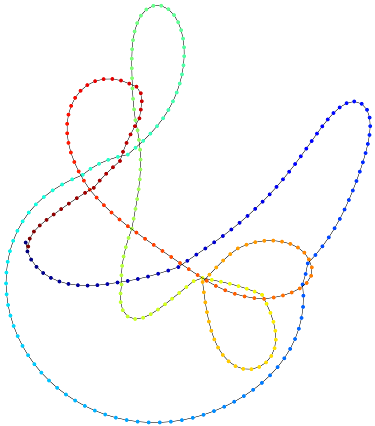


Figure: fdp layout by Graphviz

Graph Drawing as Continuous Optimization Problem

- Research on graph drawing by **continuous optimization** is a hot topic.
  - stress majorization [Gansner et al., 2005]
  - **L-BFGS** method [Hosobe, 2012] (8 cites)
  - GPU based [Gajdoš et al., 2016]
  - **SGD** (stochastic gradient descent) [Zheng et al., 2019] (66 cites)
  - and so on [Ahmed et al., 2020, Ahmed et al., 2022, Khan et al., 2024]

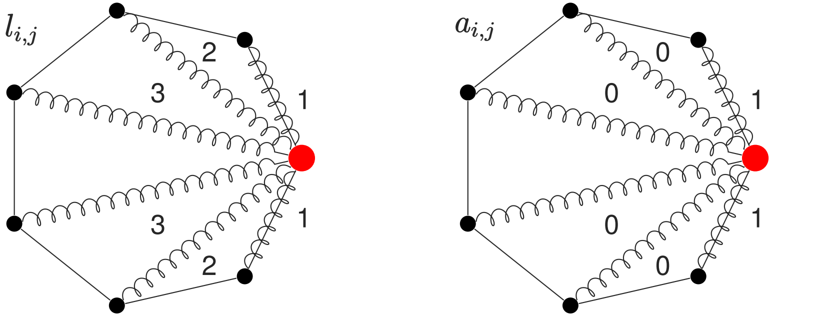
todo: explanation of L-BFGS and SGD
L-BFGs: quasi-Newton method with limited memory
SGD: stochastic gradient descent (mini-batch)

→ Our research aims to take over this flow of the research.

Problem: Inapplicability of SGD

- **SGD** is powerful for KK layout, but **not for FR layout**.
- KK layout first computes the optimal distance $l_{i,j}$ by Dijkstra's algorithm.
- FR layout has $a_{i,j} = 0$ if v_i, v_j are not connected.

$$\text{(KK)} \quad f_{i,j}(d) = \frac{k}{2}(d_{i,j} - l_{i,j})^2 \qquad\qquad \text{(FR)} \quad f_{i,j}(d) = \frac{a_{i,j} d^3}{3k} - k^2 \log d$$



余談: SGD の問題点

Graph Drawing by Stochastic Gradient Descent Jonathan X. Zheng, Samraat Pawar, Dan F. M. Goodman からの抜粋:

このアイデアにはいくつかの問題があります。ひとつは、すべての長距離の力を同等に扱うことが、グラフ理論上の距離に忠実でないという点です。もうひとつは、これらの力の相対的な強さが、グラフの最終的な配置に大きな影響を与える可能性のある追加パラメータに依存しているという点です【23】。これをグラフ理論上の距離に依存させ続ければ、これらの問題を回避することができますが、最短経路の計算と保存の問題が再び発生します。この依存関係を維持するためのアプローチのひとつとして、Khoury らによるものがあります【21】。彼らは、特異値分解に基づく距離行列の低ランク近似を用いています。これにより非常に高い効果が期待できますが、それでも APSP（全対最短経路問題）を必要とします。

The Purpose of this research

Existing works	[Hosobe, 2012] [Zheng et al., 2019]	[Ghassemi Toosi et al., 2016]	[Gower et al., 2019]
abstract	L-BFGS / SGD for graph drawing	pre-processing as initial placement	random subspace algorithm
limitation	L-BFGS is slow SGD is un-apllicable	just using a circle layout	no existing work for FR layout
What we do	practical speed up	refine the strategy	apply to FR layout

余談: 試した手法群

- シンプルな random subspace algorithm
  - この pair-wise separable function に対してなら、絶対に random subspace 的アルゴリズムが上手く機能するはずという直感が、全ての出発点
  - **ダイレクトな適応だと上手くいかないが、適切な工夫でどうにか出来ないか?** というのがこの研究の核心 想定以上に random subspace は有効そうという結論です
- (丸茂先生から頂いた非常に本質的なアドバイスである) 頂点毎に L-BFGS を適用
  - シンプルな random subspace algorithm と同じ理由で遅い
  - (cubic regularized Newton 以前の根本的な問題)
- CG 法
  - 元々の研究課題の一環として、SciPy が実装しているリスタート機能付きの β_{+}^{PR} と強 Wolfe 条件による CG 法を調べていました
  - Zoutendijk 条件による証明について、簡単な部分は理解したつもりですが、subspace 化した時にどうすればいいのかは全く分からず
- stochastic CG 法 “Stochastic Conjugate Gradient Algorithm with Variance Reduction”
  - ただし、収束性証明は強凸の場合のみ
- L-BFGS 法自体の改良の検討 → 自作だと数値爆発で諦め

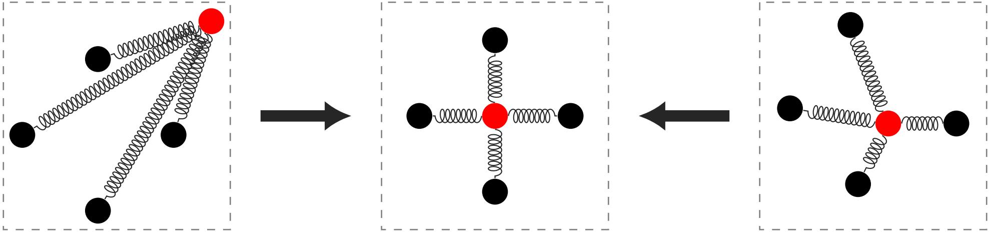
Key Observation: stochastic strategy

For far-quadratic objective functions

- ✓ **SGD** excels at rough optimization
- ✗ **L-BFGS** is ineffective for this case
  $\mathcal{O}\left(|V|^2\right)$ per iteration is too expensive

For near-quadratic objective functions

- ✓ **L-BFGS** excels at precise optimization
- ✗ **SGD** is ineffective for this case
  it is too easy to find a locally optimal solution



(本当は SGD でなく、Random subspace と読み替えてもらう方が正しい)

(FR アルゴリズムは L-BFGS に近いが、前者が苦手なことは「ねじれ」の問題として知られている)

Key Technique 1: Separation of the objective function

$$f(X) = \sum_{i<j} f_{i,j}(d_{i,j}) = \underbrace{\overbrace{\sum_{(i,j)\in E} f^a_{i,j}(d_{i,j})}^{\text{attraction}}}_{\text{sparse}} + \underbrace{\overbrace{\sum_{i<j} f^r_{i,j}(d_{i,j})}^{\text{repulsion}}}_{\text{dense}}$$

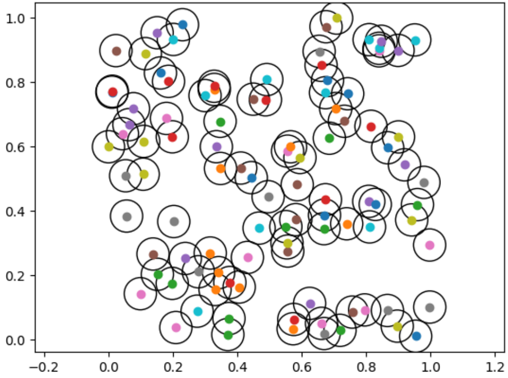
$$\rightarrow \text{minimize} \quad \sum_{(i,j)\in E} f^a_{i,j}(d_{i,j}) \quad \text{subject to} \quad f^r_{i,j}(d_{i,j}) \leq \epsilon, \quad i < j$$

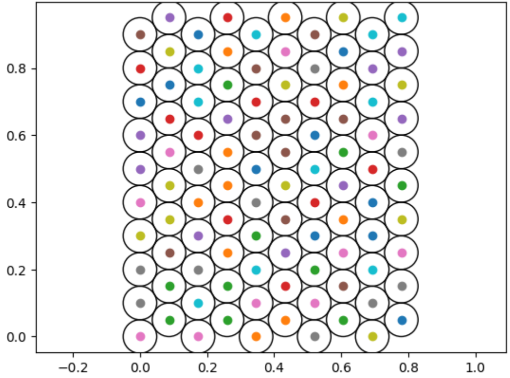
$$\rightarrow \text{minimize} \quad \sum_{(i,j)\in E} f^a_{i,j}(d_{i,j}) \quad \text{subject to} \quad \text{each } x_i \text{ has an exclusive } \epsilon'\text{-ball}$$

- Without constraints, $X = 0$ is the optimal solution $\rightarrow$ **closest packing**
- With **hexagonal close-packed** structure, f is the sum of $|V|^2 \rightarrow |E|$ terms
- partially overlapped with [Ghassemi Toosi et al., 2016, Li et al., 2022]
- (これはあくまで前座で、Key Technique 2 が上手く動作する為の準備)

proposed algorithm 1

- exact projection: min cost flow problem ($\mathcal{O}(|V|^3)$)
- heuristic projection: sort by coordinates ($\mathcal{O}(|V| \log |V|)$)





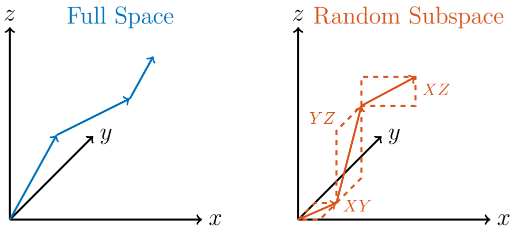
余談: TSP との関連性

- 固定された頂点配置に対して、サイクルグラフの頂点の割り当てを考える
- 最小化すべき関数 $f^a(d)$ を、FR レイアウトに限らず一般化して考える
- 特に、$f^a(d) = d$ とした時、これは TSP に他ならない
- TSP は当然 NP 困難 ヒューリスティックによって解かれる
  - 代表的な TSP の解法: (Held-Karp), 2-opt, 3-opt, Lin-Kernighan, Christofides, etc.
  - 参考になる部分とならない部分がそれぞれある。
- 面白い関連性として挙げてみました

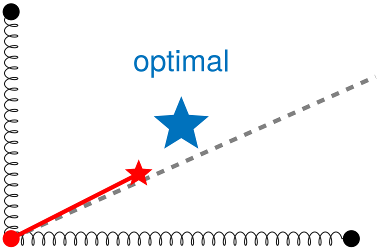
Key Technique 2: Randomized subspace algorithm

- Researches on Randomized Subspace Newton (**RSN**) are conducted.
  - Proposed in [Gower et al., 2019]
  - Extended in [Cartis et al., 2022],[Fuji et al., 2022],[Higuchi et al., 2024]
  - Analysis is based on [Karimireddy et al., 2018]
- (疑問:random subspace を名乗るのに JL-lemma は必須か? 提案論文は不使用)
- (単なる勘ですが、この状況まで落とし込めば部分空間法が強いはず)

- Apply some second-order method to subspaces
- graph drawing: optimize over $\prod_{i=1}^{n} \mathbb{R}^2$ $\to$ suitable for **RSN**!



Advantages of Randomized Subspace Newton



$g = \nabla f$

Descent direction

$d = (\nabla^2 f)^{-1} \nabla f$

Newton's direction

- random subspace problem (subproblem for vertex i):

$$\min_{x_i \in \mathbb{R}^2} \sum_{j \in \mathrm{Adj}(i)} \frac{a_{i,j} \|x_i - x_j\|_2^3}{3k} \quad \leftarrow \textbf{CONVEX optimization}$$

- Original problem, subspace problem, and restricted problem are all **NON-CONVEX**
- All distance $d_{i,j} = \|x_i - x_j\|_2 \geq \epsilon' \rightarrow$ numerically stable

proposed algorithm 2

Algorithm 1: Randomized Newton Direction Update for Vector v_i

Input: Set of indices S, initial vectors v_i

Output: Updated vector v_i

```
Initialize X randomly
while not converged do
    Randomly select an index i ∈ S except for the taboo list
    Compute Newton's direction d_i = (∇²f_i)^{-1} ∇f_i
    Update v_i subject to the constraints
    if not updated then
        Add i to the taboo list
    end
end
return X
```

Possible Application 1

- Originally, this kind of problems are solved by **stochastic coordinate descent**
- Random subspace algorithms might be applicable to these problems

6.2 Objective functions arising from graphs

Many optimization can be written as a linear combination of functions that only involve pairs of variables coupled due to some graph structure. For example, problems in image segmentation might couple adjacent pixels. In topic modeling, terms that appear in the same document may be coupled.

Consider an undirected graph $G = (V, E)$ where the edges $(j, l) \in E$ connect two vertices j and l from $V = \{1, 2, \ldots, n\}$. Suppose our objective has the following form, where each component x_i of the variable $x \in \mathbb{R}^n$ is associated with vertex i:

$$f(x) = \sum_{(j,l) \in E} f_{j,l}(x_j, x_l) + \lambda \sum_{j=1}^{n} \Omega_j(x_j),$$

where f_{jl} (for all $(j, l) \in E$) and the regularization functions Ω_j (for $j = 1, 2, \ldots, n$) are all differentiable. Evaluation of the functiion f and the full gradient ∇f would be an $O(|E|)$ operation (if we assume that evaluation of each f_{jl} and ∇f_{jl} is $O(1)$). To implement a CD method efficiently, we could store the values of f_{jl} and ∇f_{jl} at the current x, for all $(j, l) \in E$. To compute the ith gradient component $\nabla_i f(x)$, we need to sum components from the terms $\nabla f_{jl}(x)$ for which $j = i$ or $l = i$ (at a total cost proportional to the number of edges incident on vertex i) and evaluate the term $\nabla \Omega_i(x_i)$. In updating the values of f_{jl} and ∇f_{jl} after the step in x_i, we need again only change those components for which $j = i$ or $l = i$. The "expected" cost of one CD iteration is thus $O(|E|/n)$. We see once again the desired $1/n$ relationship between the cost per iteration of CD and the cost per iteration of a gradient method.

Figure: hyperlink

-Friendly Structures | Convergence of Randomized Coordinate Descent | Faster Coordinate Optimization | Stochastic Gra

Pairwise-Separable Functions

- A more interesting example is pairwise-separable functions,

$$f(w) = \sum_{i=1}^{d} \sum_{j=1}^{d} f_{ij}(w_i, w_j),$$

which depend on a function of each pair of variables.

  - This includes quadratic functions.

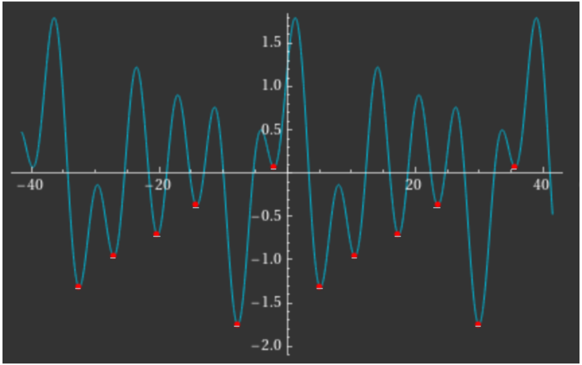
- An example is label propagation for semi-supervised learning.
  - In this application, each f_{ij} measures how similar labels are between neighbours.

- Cost of gradient descent vs. coordinate descent:
  - Double-sum has $O(d^2)$ terms.
    - Gradient descent needs to compute gradient of all these terms.
  - Each w_j only appears in $O(d)$ terms.
    - Coordinate optimization only needs to use these terms.

Figure: hyperlink

Possible Application 2

- Optimize $f(x)$ where $x \in \mathbb{R}^n$, not $\prod_{i=1}^{n} \mathbb{R}^2$.
- Assume that if two of the points are too close, $f(x)$ drastically increases.
- Similar algorithms might be applicable to this problem.
- Is there any concrete instance?
- Is there any algorithm already well known?



Possible Application 3

- (Continuous relaxation of) graph isomorphism problem
  - displaying symmetry is at least as difficult as graph isomorphism [Eades, 1984]
  - When we draw $G \coloneqq G_1 \cup G_2$, G displays symmetry if $G_1 \cong G_2$.
- Graph isomorphism problem with Frank-Wolfe algorithm [Klus and Gelß, 2023]

$$\begin{aligned} \underset{Q}{\text{minimize}} \quad & \|QA - BQ\|_F^2 \\ \text{subject to} \quad & Q \in \left\{Q \in \mathbb{R}^{n \times n} \mid Q^\top Q = I_n\right\} \end{aligned}$$

- Many-to-Many graph isomorphism [Zaslavskiy et al., 2010]

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & \|G - PHP^\top\|_F^2 \\ \text{subject to} \quad & P \in \left\{P \in \{0,1\}^{n \times n} \mid P1_n = 1_n, P^\top 1_n = 1_n\right\} \end{aligned}$$

- (随分な遠回りでしたが、Random subspace+Riemannian optimization という当初の目標への回帰?)

Summary

Future Work

- More numerical experiments
- Theoretical analysis
- (少なくとも自分が知る限り and 思う限り) かなり面白い話だと思いますし、多少はやる価値がある内容だと自負しています

Reference I

[Ahmed et al., 2020] Ahmed, R., De Luca, F., Devkota, S., Kobourov, S., and Li, M. (2020).
Graph Drawing via Gradient Descent, $(gd)^2$.
In Auber, D. and Valtr, P., editors, *Graph Drawing and Network Visualization*, pages 3–17, Cham. Springer International Publishing.

[Ahmed et al., 2022] Ahmed, R., De Luca, F., Devkota, S., Kobourov, S., and Li, M. (2022).
Multicriteria Scalable Graph Drawing via Stochastic Gradient Descent, $(sgd)^2$.
IEEE Transactions on Visualization and Computer Graphics, 28(6):2388–2399.

[Barnes and Hut, 1986] Barnes, J. and Hut, P. (1986).
A hierarchical O(N log N) force-calculation algorithm.
Nature, 324(6096):446–449.

[Cartis et al., 2022] Cartis, C., Fowkes, J., and Shao, Z. (2022).
Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares.
2211.09873.

[Chrobak and Payne, 1995] Chrobak, M. and Payne, T. H. (1995).
A linear-time algorithm for drawing a planar graph on a grid.
Information Processing Letters, 54(4):241–246.

[Eades, 1984] Eades, P. (1984).
A heuristic for graph drawing.
Congressus numerantium, 42(11):149–160.

Reference II

[Fruchterman and Reingold, 1991] Fruchterman, T. M. J. and Reingold, E. M. (1991).
Graph drawing by force-directed placement.
Software: Practice and Experience, 21(11):1129–1164.

[Fuji et al., 2022] Fuji, T., Poirion, P.-L., and Takeda, A. (2022).
Randomized subspace regularized Newton method for unconstrained non-convex optimization.
2209.04170.

[Gajdoš et al., 2016] Gajdoš, P., Ježowicz, T., Uher, V., and Dohnálek, P. (2016).
A parallel Fruchterman–Reingold algorithm optimized for fast visualization of large graphs and swarms of data.
Swarm and Evolutionary Computation, 26:56–63.

[Gansner et al., 2005] Gansner, E. R., Koren, Y., and North, S. (2005).
Graph Drawing by Stress Majorization.
In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., and Pach, J., editors, *Graph Drawing*, volume 3383, pages 239–250. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Ghassemi Toosi et al., 2016] Ghassemi Toosi, F., Nikolov, N. S., and Eaton, M. (2016).
Simulated Annealing as a Pre-Processing Step for Force-Directed Graph Drawing.
In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 997–1000, New York, NY, USA. Association for Computing Machinery.

Reference III

[Gower et al., 2019] Gower, R., Kovalev, D., Lieder, F., and Richtarik, P. (2019).
RSN: Randomized subspace newton.
In Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Higuchi et al., 2024] Higuchi, R., Poirion, P.-L., and Takeda, A. (2024).
Fast Convergence to Second-Order Stationary Point through Random Subspace Optimization.
2406.14337.

[Hosobe, 2012] Hosobe, H. (2012).
Numerical optimization-based graph drawing revisited.
In *2012 IEEE Pacific Visualization Symposium*, pages 81–88.

[Hu, 2006] Hu, Y. (2006).
Efficient, high-quality force-directed graph drawing.
The Mathematica journal, 10:37–71.

[Kamada and Kawai, 1989] Kamada, T. and Kawai, S. (1989).
An algorithm for drawing general undirected graphs.
Information Processing Letters, 31(1):7–15.

[Karimireddy et al., 2018] Karimireddy, S. P., Stich, S. U., and Jaggi, M. (2018).
Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients.
1806.00413.

Reference IV

[Khan et al., 2024] Khan, B., Johnstone, M., and Creighton, D. (2024).
A Many-objective Evolutionary Algorithm Approach for Graph Visualization.
In *2024 IEEE International Systems Conference (SysCon)*, pages 1–7.

[Klus and Gelß, 2023] Klus, S. and Gelß, P. (2023).
Continuous optimization methods for the graph isomorphism problem.
2311.16912.

[Kobourov, 2012] Kobourov, S. G. (2012).
Spring embedders and force directed graph drawing algorithms.
arXiv preprint arXiv:1201.3011, 1201.3011.

[Li et al., 2022] Li, J., Tao, Y., Yuan, K., Tang, R., Hu, Z., Yan, W., and Liu, S. (2022).
Fruchterman–reingold hexagon empowered node deployment in wireless sensor network application.
Sensors, 22(5179).

[Sugiyama et al., 1981] Sugiyama, K., Tagawa, S., and Toda, M. (1981).
Methods for Visual Understanding of Hierarchical System Structures.
IEEE Transactions on Systems, Man, and Cybernetics, 11(2):109–125.

[Tunkelang, 1999] Tunkelang, D. (1999).
A Numerical Optimization Approach to General Graph Drawing.
PhD thesis, Carnegie Mellon University.

Reference V

[Tutte, 1963] Tutte, W. T. (1963).
How to Draw a Graph.
Proceedings of the London Mathematical Society, s3-13(1):743–767.

[Zaslavskiy et al., 2010] Zaslavskiy, M., Bach, F., and Vert, J.-P. (2010).
Many-to-Many Graph Matching: A Continuous Relaxation Approach.
In Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 515–530, Berlin, Heidelberg. Springer.

[Zheng et al., 2019] Zheng, J. X., Pawar, S., and Goodman, D. F. M. (2019).
Graph drawing by stochastic gradient descent.
IEEE Transactions on Visualization and Computer Graphics, 25(9):2738–2748.

どう研究したいか

- 以下は本当に仰る通りで、非常に申し訳なく思います。すいません。
  - 「PL や Andi がいるのに，彼らからの助言を仰がずに，多様体や random subspace アルゴリズムから離れて，あまり専門家のいない方向へ進んでいくのは，なんだか勿体無い気がしたもので.」「正直，我々の得意とするところから離れるほど，困った時にアドバイスできなくなるし，新規性があるのかどうかも判断つかなくなります．もちろん，やりたいことをやったらいい気もしますが，私のところにきて 1 つめの研究（学生の持ち込んだ研究）で論文に採択レベルの新規性のある研究ができた人は今のところ，いないので.」
- 前提として、5 研の皆さんと研究対象は出来る限り近くにしたいです。楽なので。
- 言い訳として、研究のやり方にかなり大きなズレを感じています。
  - 「まず問題を見つける (spring layout が L-BFGS などでも遅い) → 解決策を考える (座標降下など、特定の手法に拘らず全ての選択肢を考える) → 収束性など理論保証をする (→ 他手法へも適用可な一般的アルゴリズムへ拡張する)」というボトムアップはかなり得意な自信があります。(e.g. 7 研、RA、趣味)
  - 一方で、「まず一般的なアルゴリズムを考える (random subspace) → 理論保証をする → 上手くいく問題を見つける (機械学習など)」というトップダウンの研究は滅茶苦茶に苦手なのだなと痛感しています。
  - 理論解析自体は、かなり好きです (e.g. 丸茂先生の夏季学校の問題)

どう研究するべきか

- 自分の中での最大の疑問: トップダウン型研究のゴールはどこにあるのか?
- Pierre さんらの「この研究は数値実験を真面目にやっていない」という発言は自分の中でかなり衝撃的 (以前からかなり迷子気味でしたが、これで完全に路頭に迷いました)
  - 想定される「ゴール」というのは中程度の優位性を持ったアルゴリズムの開発でしょうか?
  - しかし、私は研究の動機 (次スライド) をかなり数値実験に置いた人間です
  - 一般論として、汎用解法で汎用的な問題に対して STOA を与えること (トップダウン) は至難の業だが、特定の問題に特定の解法を与えること (ボトムダウン) は比較的容易
  - トップダウンだと、上記 2 つの間でどうしても矛盾が生じてしまう
- 一般論として、トップダウン型研究はどういうモチベーションでやるべきですか?
  - 他分野の研究ならともかく、最適化の研究でトップダウン (手法ありき) はかなり難しい印象を受けています。
  - トップダウン型研究では存在しないゴールテープを追いかける羽目になりませんか? どうそれを回避するべきですか?
  - 「上手く行く問題を見つける」というゴールポストずらしで、良い成果が出る未来がシンプルに想像できず、かなり困ってしまいました。
  - 「相補的 (by 品川先生)」というのが大きなヒントかなとは感じています。

何を研究したいか

- 4 月あたりに研究したい分野は特に無いと答えてしまいましたが、数か月経って少しは自分のしたいことが定まってきた気がします
  - 「離散と連続の橋渡し」 離散に対する思い入れが思ってた以上に大きかったです
  - 離散と連続の両面を持つ問題に対して、それぞれの特性を活かしたアルゴリズムを考えるのが最高に楽しい
- 自分が研究職を志す動機は梅谷先生に 2 年前メールをしたのが恐らく原点です。
  - 「しっかり学ぶ数理最適化」のヒューリスティックの章に対し、詳細な実装例を与えた
  - 「メタヒューリスティックの研究者はとても少ないので目指して下さい」とお世辞を頂けた
  - (最近分かったこととして、ヒューリスティックとトップダウン研究はかなり相性が悪い)
  - 数理 5 研で直接その研究をする気は皆無ですが (専門家の助言を受けられる内にそういうことはしたい為、特に丸茂先生の話は許される限り関わりたいです)、問題固有の構造を生かすということは常に意識していたいです
- 梅谷先生も武田先生も共に産業側との接点があるという認識でいますが、そこは割と研究室志望の動機でもあります。
  - 「世の中の「困った」を解決する」一助になりたい
  - 殆どお金に結びつかない (機械学習/GPT 等ではないということ) が需要は存在する問題への、公共財としての実用的なプログラム開発が、人生レベルでの目標の一つ
    - networkx への contribution はこの研究が論外と否定されたとしてもやっておきたい
  - 自分が理論解析に対して消極的 (に見える?) のは、この辺りの動機が要因です

メモ

- リスクヘッジをご配慮して下さっている印象を受けましたが、学生身分であることへの楽観 (新規性皆無 or 先行研究丸被りでも死に至らない) に基づき、(意味のある) リスクを取ること自体は厭わないつもりです
  - 幸運にも勝ち得た実績でギャンブルをしたい (工学部長賞・JSIAM・QS・山下記念・AQIS)
  - 上振れを引いたら更にリスクをとって上振れを狙いたい性格 破滅したらその時に考えたい楽観主義者です
  - (我儘を言えば、ハイリスク・ローリターンなことをしていたら私を止めて頂きたいです)
- 自分が大学院において真に最小化したいリスクは、「卒業できない (流石に卒業くらいはしたいですが……) or 奨学金取れない」ではなく、「将来研究者として自走できない」こと
  - 奨学金自体はその実績で生活できる分くらいは恐らく確保できそう
  - そういう「上振れ」を引いているのなら、今の自分が取るべきリスクは、研究者としての自走力を養うこと
  - 成果がほぼ確約されているが自分が将来やらない類の研究は、可能なら回避したいです

「この研究を続けて良いか?」No の場合

- PL さんや Andi さんやの助言を仰げる範囲だと、私の能力不足の為、まず数値実験結果でいい結果が出る未来があんまり見えていません。路頭に迷っている状態なので、かなり大きく助言を頂きたいです。
- CG 法に関する話の方は、以前お伝えした通り、実質 SciPy の下位互換なことが分かってしまいました。
- いずれにせよあまり明確なビジョンを持っていない状態なので、色々と助言を頂ければ幸いです。

「この研究を続けて良いか?」Yes の場合

- 論文の形にまとめること / 数値実験をきちんと行うこと など、やるべきことは比較的明確
- 11 月中までに論文の原型が書くことが今のところの目標
- まず Key Technique 1 の方に対して、理論保証は絶対にしたいと考えています。
  - 当初は「より良い大域最適解へ収束することを言えばよい」→「武田先生がこれまでにした研究の中だとガウシアンホモトピー法の研究が動機としては一番近くて参考になる???」と考えていました
  - ただ、この研究を私が理解した範囲内だと、まず手法が収束すると言えることが前提?
  - 射影勾配法の収束性証明の議論を援用して何か言えるか?
  - また、最適化の専門家から見て自明な改善点は多いと思いますが、それの議論
- random subspace(?) の方については、そもそも何が言えれば嬉しいか?
- SGD+taboo はもっとあるかと思っていたら全然ありませんでした。そもそも「研究」の対象ではない?