効率的な最適化手法 "L-BFGS法"の、NetworkX での活用と SciPy の実装について

SciPy Data 2026
浜口広樹

自己紹介

玻璃という名前でTwitterなどをやっています。

「浜口広樹」で検索すると、個人のHPがあり、本日のスライドがあります。

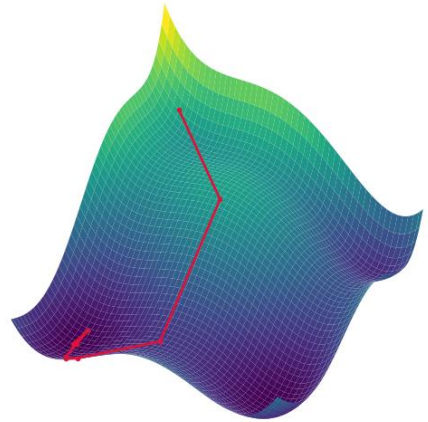
- 浜口広樹 (はまぐち　ひろき)
- 所属
  - 東京大学 大学院 情報理工学系研究科 数理情報学専攻 (修士2年)
  - 来年度より博士課程進学予定
- 専門分野
  - 連続最適化、離散最適化、量子情報など
- 研究テーマ
  - 不正確な関数値下における実用的な正則化準ニュートン法
  - 最適化手法を用いたグラフ描画とその初期配置
  - 点配置同定問題における解の一意性条件
  - 魔法状態評価の大規模化に向けた効率的アルゴリズム

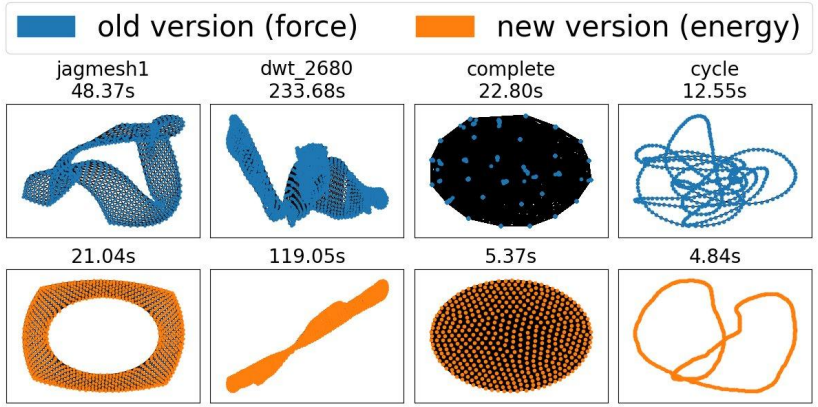
本日の概要

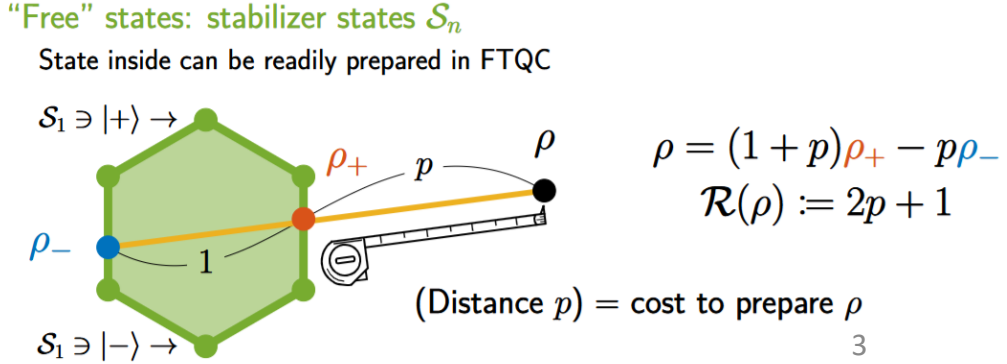
1. 連続最適化とL-BFGS法の概説
(数学、基礎)

2. NetworkXへのL-BFGS法を
用いた貢献について
(情報科学、応用)

3. その他の学術的な応用例
(基礎と応用の橋渡し)







Part1.連続最適化とは?

- 数理最適化

$$\underset{x \in C}{\text{minimize}} \quad f(x)$$

- 連続最適化
  変数が連続的に変化する

$$x \in \mathbb{R},\ \mathbb{R}^{n \times m},\ \mathbb{C},\ \mathrm{St}(n,m), \ldots$$

- 離散最適化(組合せ最適化)
  変数が離散的に変化する

$$x \in \mathbb{N},\ \mathbb{Z},\ \{0,1\}^n,\ \mathbb{N}^n \times \mathbb{R}^m, \ldots$$

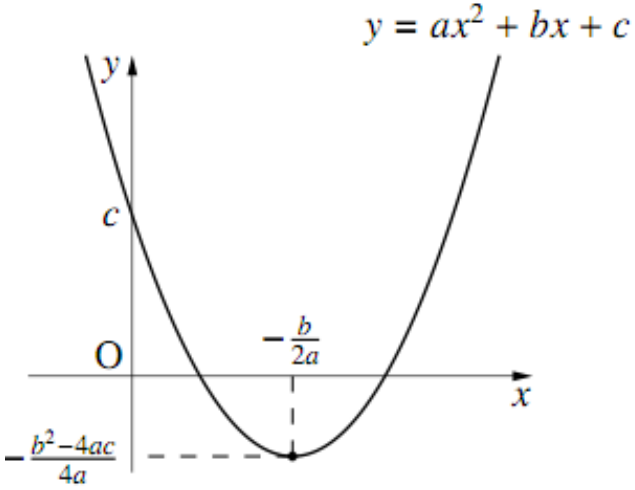
引用: 日本オペレーションズ・リサーチ学会 (CC BY-ND 4.0)

例:

- 累計売上金額や社会の動向から、将来の需要を予測
- 在庫状況や商品の納品個数から、生産計画を立案

無制約最適化問題 $\underset{x \in \mathbb{R}^n}{\text{minimize}} \ f(x)$ の解法

簡単な問題 (2次形式)

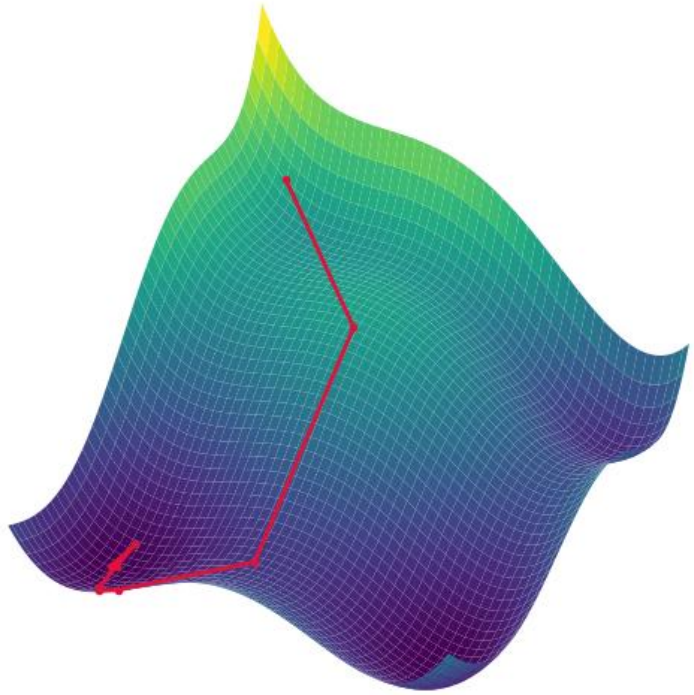


$$f(x) = x^\top A x + b^\top x + c \quad (A \succ 0)$$

$$\nabla f(x) = 2Ax + b$$

$$x^* = \mathrm{argmin}_{x \in \mathbb{R}} f(x) = -\frac{1}{2} A^{-1} b$$

複雑な問題



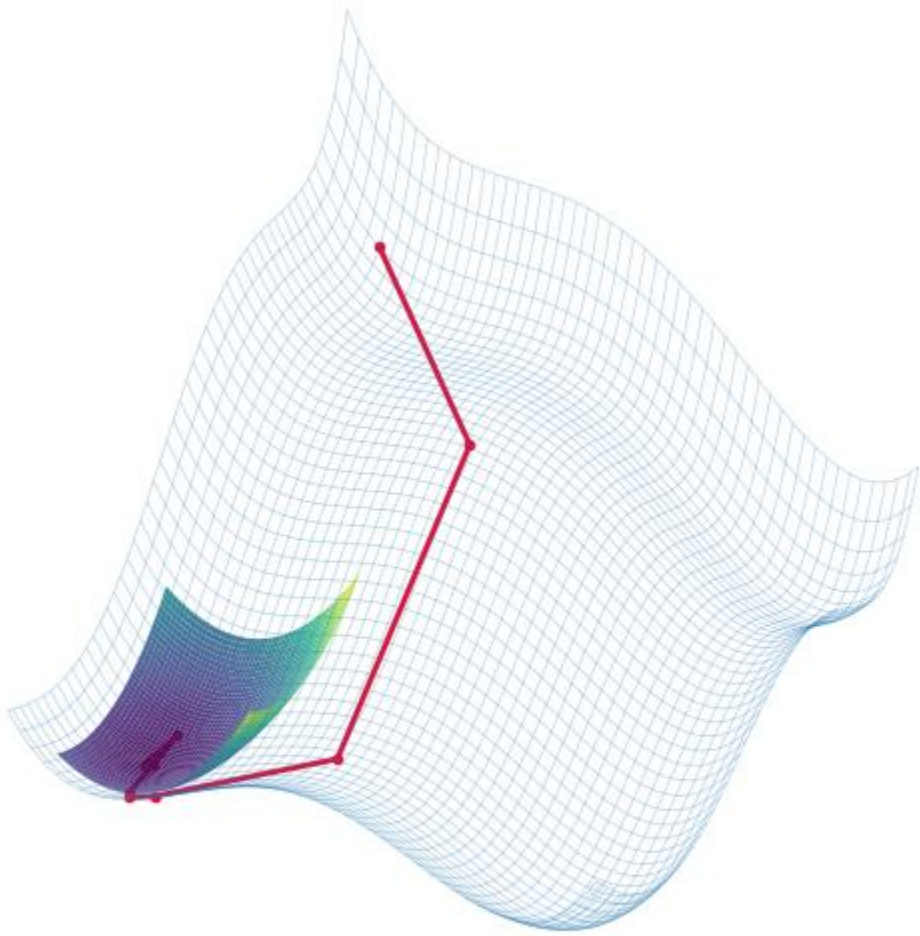
$$f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right) x^2 + xy + \left(-4 + 4y^2\right) y^2$$

反復法: 逐次的に簡単な問題(2次形式)に帰着する
代表例がニュートン法や準ニュートン法(SciPyのL-BFGS-B)

準ニュートン法とは?

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x)$$

準ニュートン法はこれを解く代表的手法で実用上最速。



点列を $x_0, x_1, \ldots, x_k, \ldots$ と逐次的に更新して、最適解へ。
関数値 $f(x_k)$ とその勾配 $\nabla f(x_k)$のみから
近似ヘッセ行列($\mathrm{B_k} \approx \nabla^2 f(x_k)$)および二次近似を作る。
B_kの更新方法の代表例がBFGS法。
計算量を抑えた亜種(Limited-memory)がL-BFGS法。

Folker Hoffmann氏によるscipy.optimizeの
解説動画も理解におすすめです。

Part.1のまとめ

- 連続最適化問題

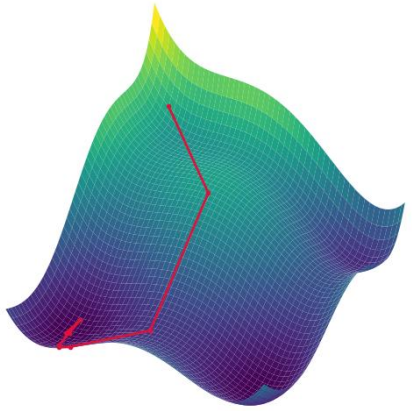
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x)$$

- 準ニュートン法が解法の一つ
- scipy.optimizeで実装されている

- あくまで個人の感想ですが、数理最適化は基礎技術寄り
- モデリング次第で実はこの技術が使える、みたいな場面が存在
- 数学的に解きにくい問題もあり、どう定式化するかが一番重要ともよく言われる

モデリング・定式化

効率的な解法の提供



Part2. NetworkXへの応用

- 2025年頃に、
  NetworkX(グラフライブラリ)へ
  SciPyおよびL-BFGS法を用いた
  Contributionを行いました
- PR-#7889 PR-#8041 PR-#8042 PR-#8145
- Issue-#8113
- 右のGalleryページは
  私が作りました
- ここではそれについてお話しします

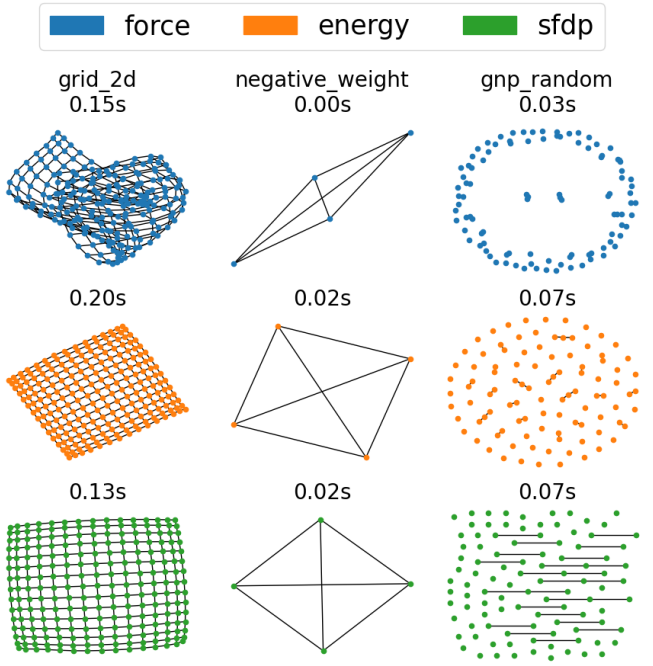
Spring Layout

Draw graphs using the three different spring layout algorithms.

The spring layout is typically generated by the Fruchterman–Reingold force-directed algorithm. This algorithm treats edges as springs holding nodes close while treating nodes as repelling objects and simulates the system until it reaches equilibrium. The algorithm also terminates when it reaches a maximum number of iterations.

NetworkX offers mainly three different kinds of methods based on the same theoretical foundation:

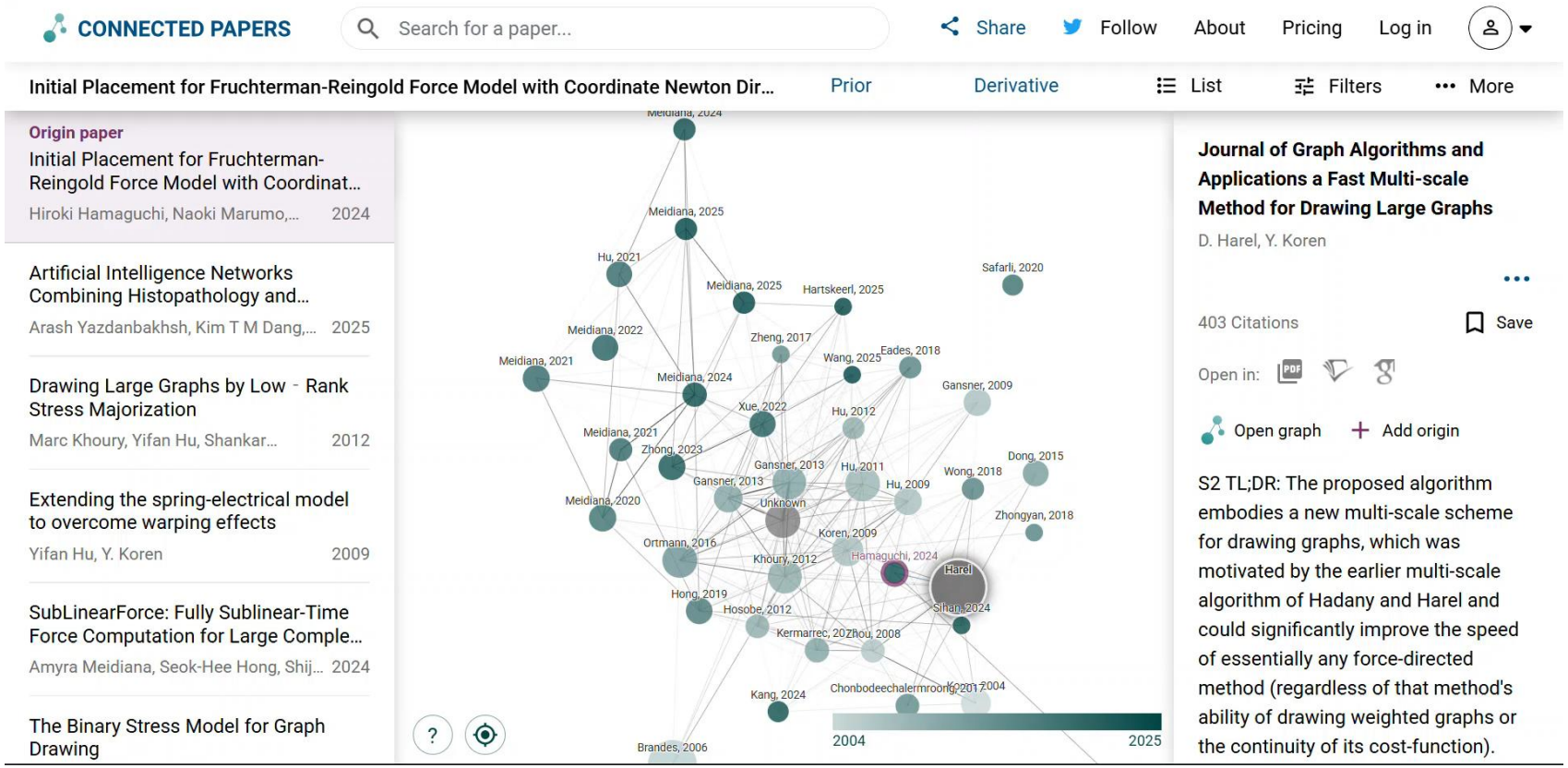
- `nx.spring_layout(G, method="force")`
  - The default for graphs with fewer than 500 nodes in `nx.spring_layout`.
  - Direct implementation of the Fruchterman–Reingold force-directed algorithm.
  - Can handle negative edge weights as they are.
- `nx.spring_layout(G, method="energy")`
  - The default for graphs with more than or equal to 500 nodes in `nx.spring_layout`.
  - It solves an energy-based optimization problem, taking the absolute value of negative edge weights.
  - Uses gravitational forces acting on each connected component to prevent divergence.
- `nx.nx_agraph.graphviz_layout(G, prog="sfdp")`
  - Uses `sfdp` from GraphViz to compute the layout.
  - Employs a multilevel approach for faster force-directed graph drawing.
  - Requires separate installation of GraphViz. For details, see `networkx.drawing.nx_agraph`



グラフ描画とは?

グラフ G = (V,E)
頂点集合　V
辺集合　E

2次元上に
辺や頂点の
重なりなく描画
できると
分かりやすい



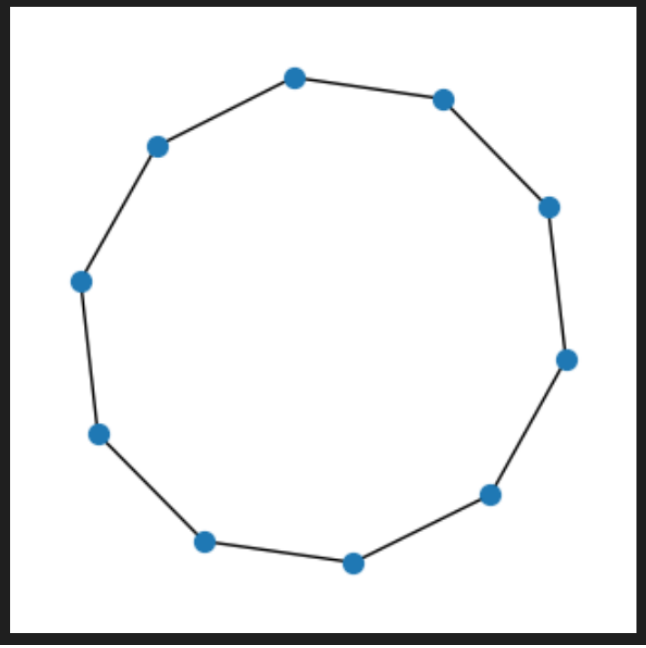
特に有名なのがFruchterman–Reingoldモデル (頂点間に引力と斥力が働いている)

従来のNetworkXが抱えていた問題

従来実装(Fruchterman–Reingoldが提案したシミュレーションベースの手法)はかなり遅い
効率的な手法(Scalable Force-Directed Placement)も存在するが、実装が非常に膨大
→　NetworkXに適した描画アルゴリズムの改善をしたいと考えていた

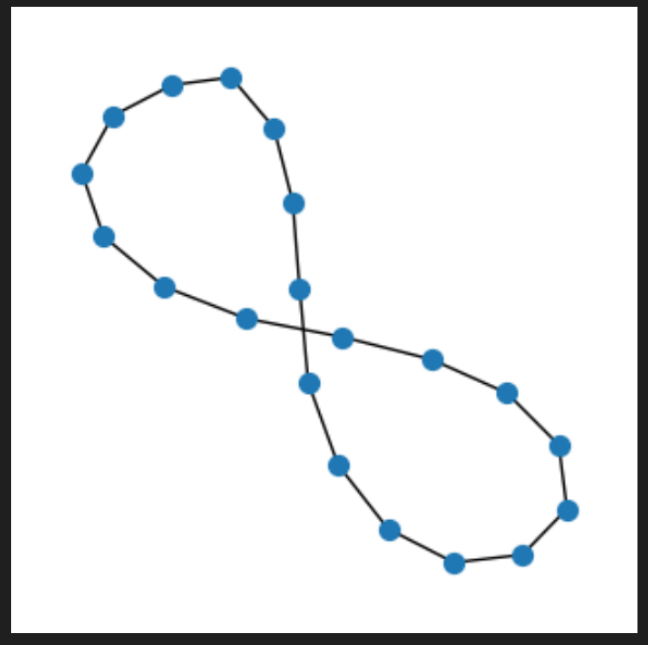
```
import networkx as nx

G = nx.cycle_graph(10)
nx.draw(G, node_size=50)
```

✓ 0.2s



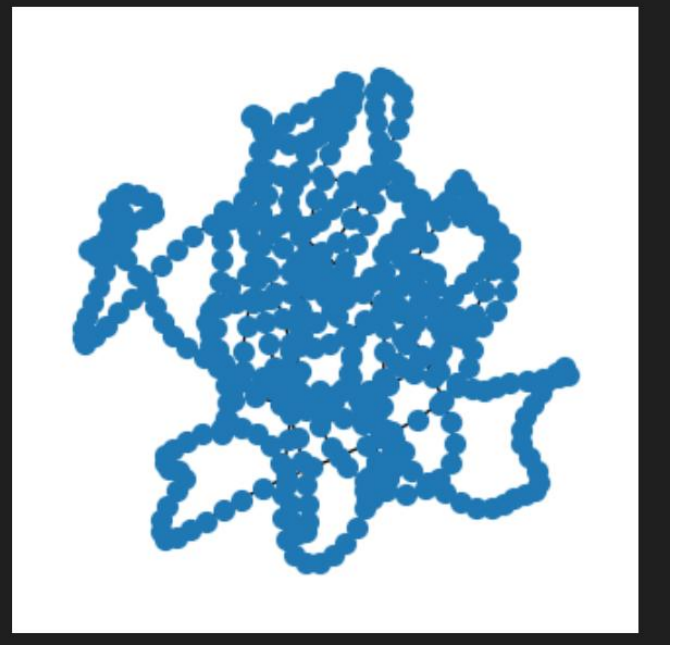
```
import networkx as nx

G = nx.cycle_graph(20)
nx.draw(G, node_size=50)
```

✓ 0.2s

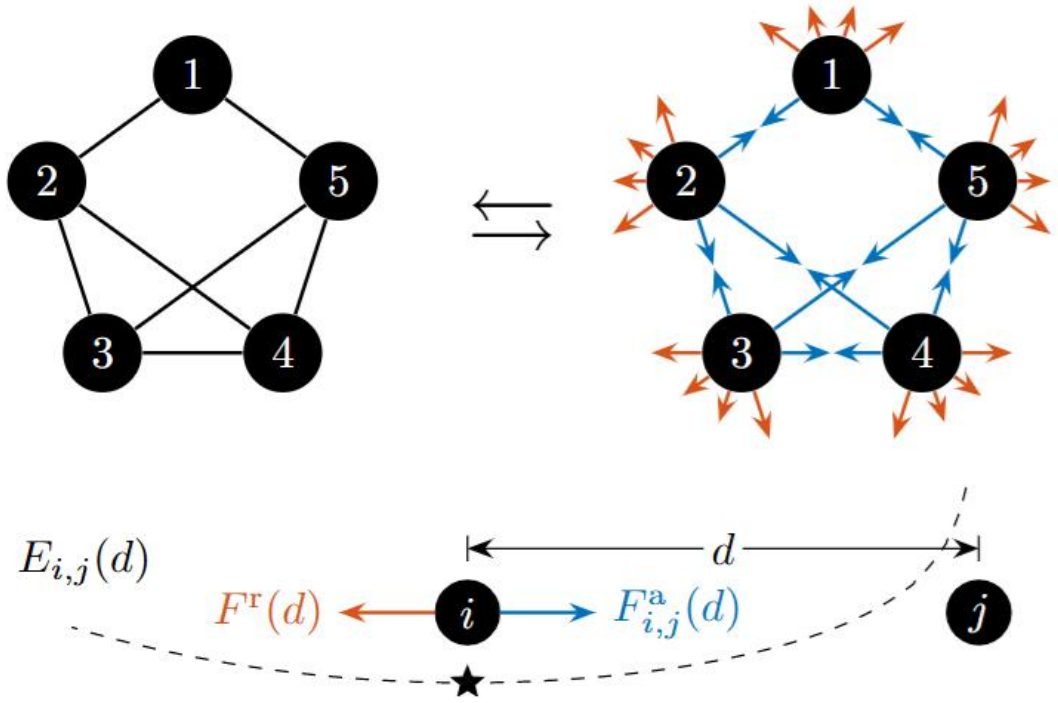


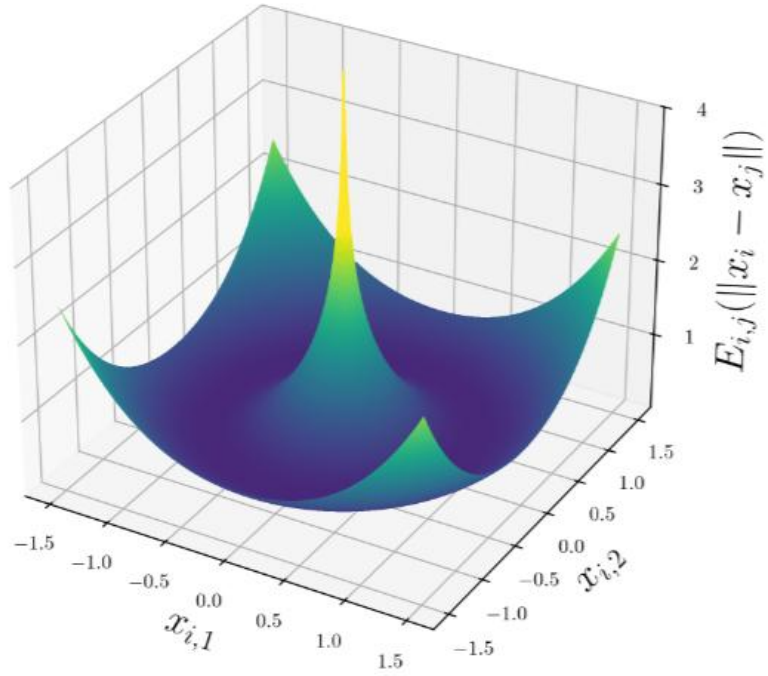
```
import networkx as nx

G = nx.cycle_graph(500)
nx.draw(G, node_size=50)
```

✓ 11.5s



連続最適化への帰着





- グラフ描画の問題は、連続最適化問題に帰着できる(先行研究など)
- それに関する研究をしており、NetworkXへは、この先行研究+αを実装

最適化問題の導出

The two forces in the FR force model:

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

Its scalar potential, energy, is defined as

$$E^{\mathrm{a}}_{i,j}(d) := \int_0^d F^{\mathrm{a}}_{i,j}(r)\,\mathrm{d}r = \frac{w_{i,j} d^3}{3k}, \quad E^{\mathrm{r}}(d) := \int_\infty^d F^{\mathrm{r}}(r)\,\mathrm{d}r = -k^2 \log d,$$

$$E_{i,j}(d) := E^{\mathrm{a}}_{i,j}(d) + E^{\mathrm{r}}(d).$$

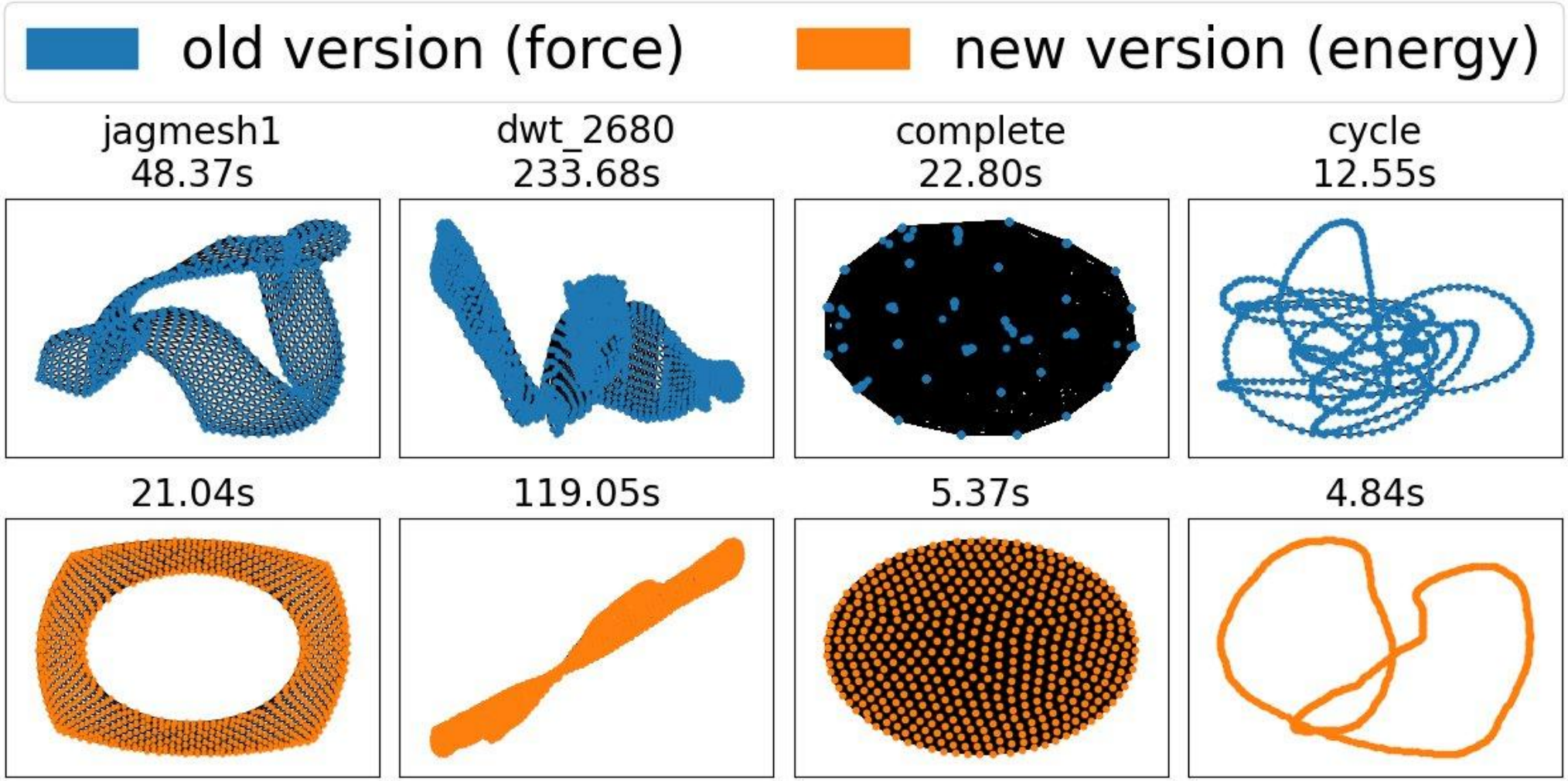
Seek equilibrium $\Leftrightarrow$ **find local minimum of** $f(X)$ (**non**-convex):

$$\underset{X \in \mathbb{R}^{2\times n}}{\text{minimize}} \quad f(X) := \sum_{i<j} E_{i,j}(\|x_i - x_j\|).$$

簡単にいうと、
1. 頂点間に力が働くモデル
2. 力を積分してポテンシャルに
3. これが最適化の対象になる

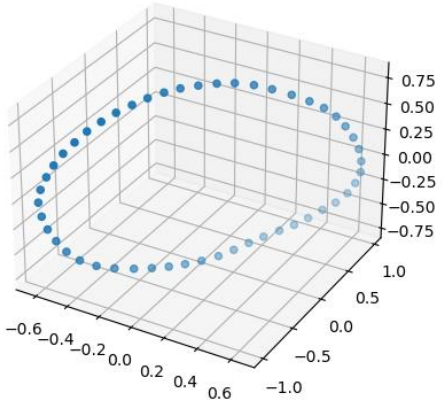
優れたソルバーを使う方が速い!

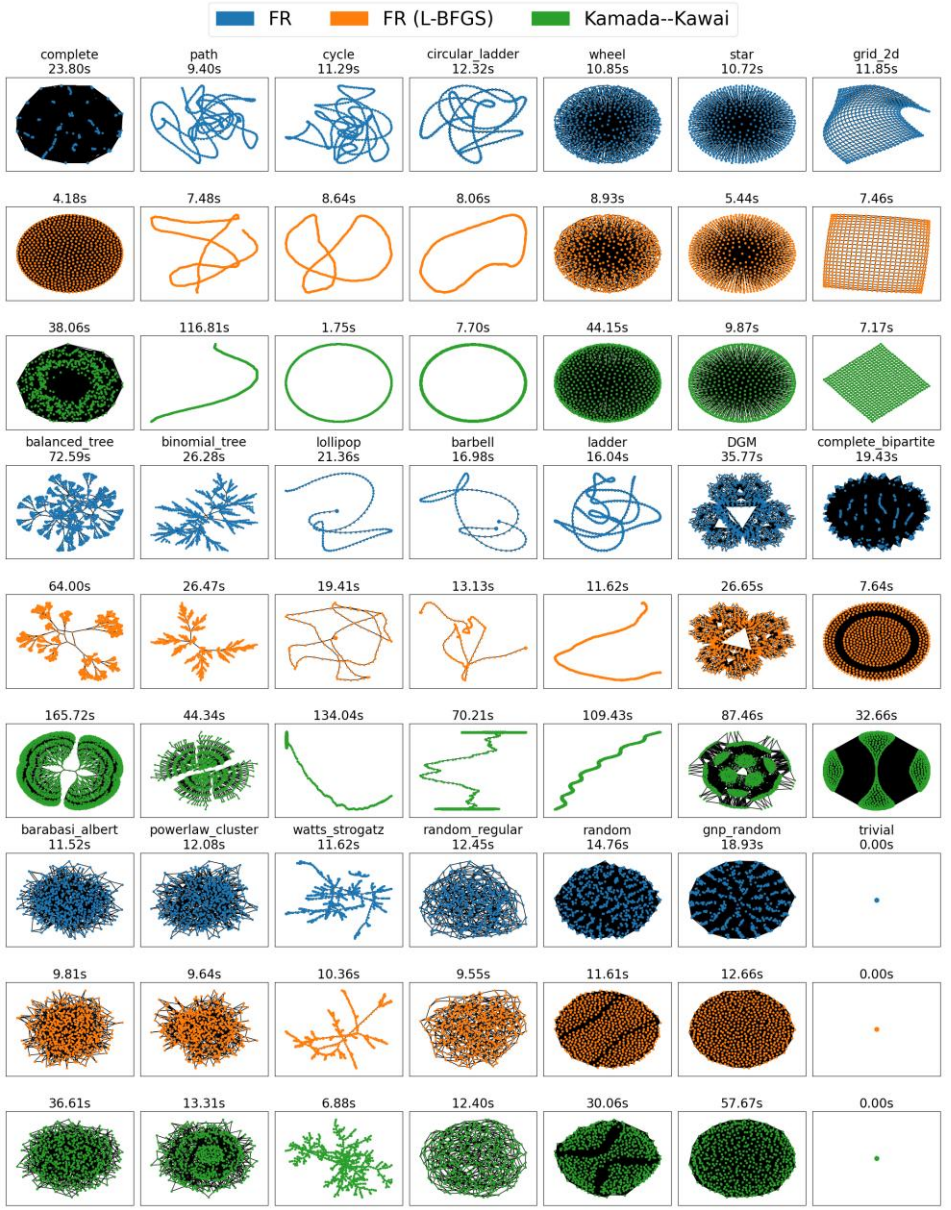
改善したアルゴリズムの出力結果



Pull Requestについて

- 網羅的な比較は結構喜ばれた
- 細かい仕様も念を入れてtest
  (一部頂点を固定した場合、
  3次元空間上で描画する場合、
  辺重みに負のものがある場合)
- 後方互換性や、今後さらに
  methodを追加する場合に
  可読性が高いか? などを
  中心にレビューして頂いた



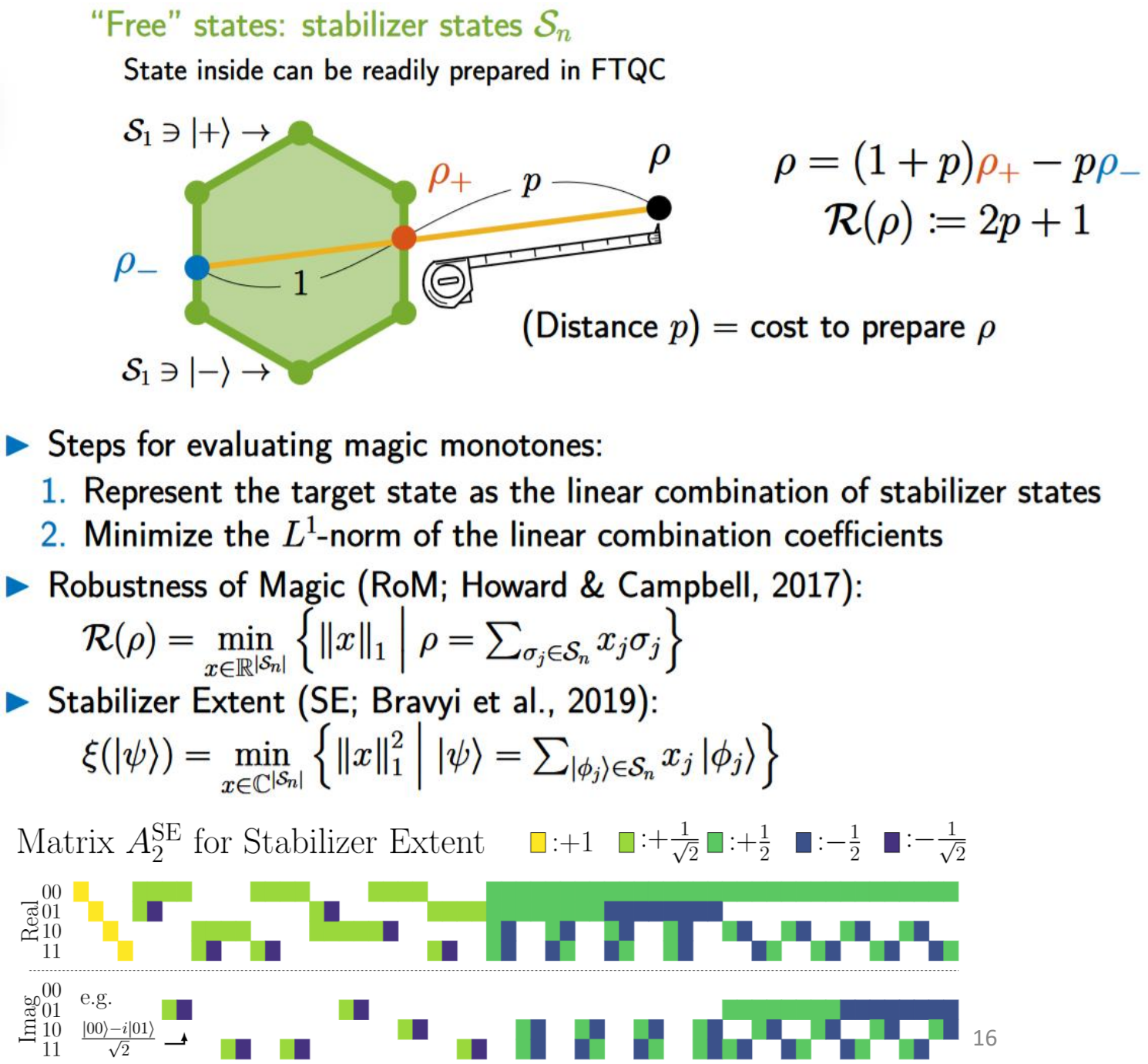


Part3. 学問寄りの応用例

- お越しくださっている皆様の興味関心は非常に多様かと思います
- そこで、浅く広く、具体的な応用例について述べようと思います
  - 参考文献のリンクなども張ったので、興味のあるものがあれば是非
- 何か琴線に触れる例があれば幸いです

量子情報

- 量子情報の研究における例
- (大雑把には)ある状態の良さが、
  空間のある集合からの距離で測れる
- 最短距離をちゃんと計算したい
  →連続最適化が使える
- (OSSではないですが)**gurobi**という
  最適化ソルバーなどがよく用いられる
- 我々の研究成果もOSSとして公開中



[Fast computation of magic monotones]
Hiroki Hamaguchi, Kou Hamada, Yoshioka Nobuyuki
Presented at 24th Asian Quantum Information Science Conference, 2024

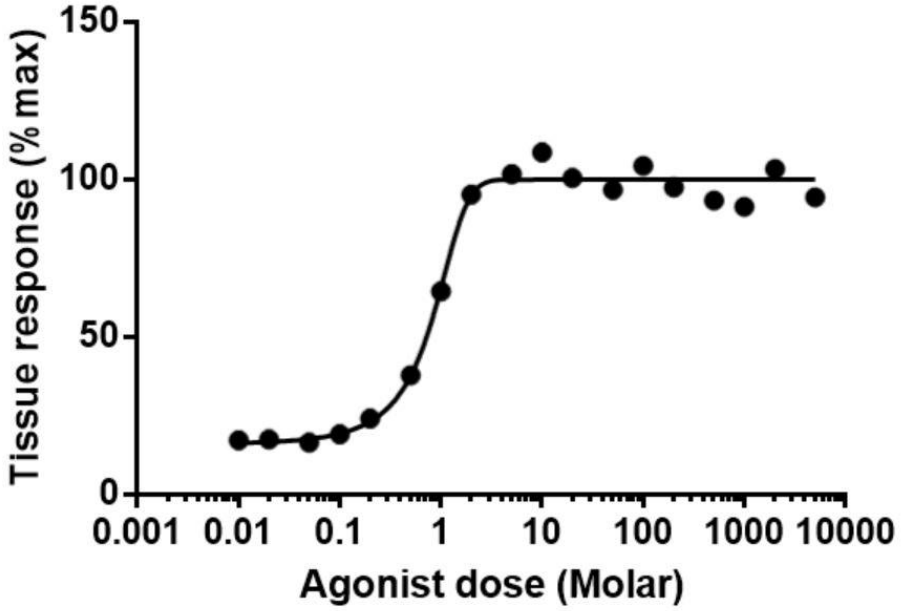
単調回帰(PAVA)

$$\min_{x} \quad \sum_{i=1}^{n} w_i (y_i - x_i)^2$$

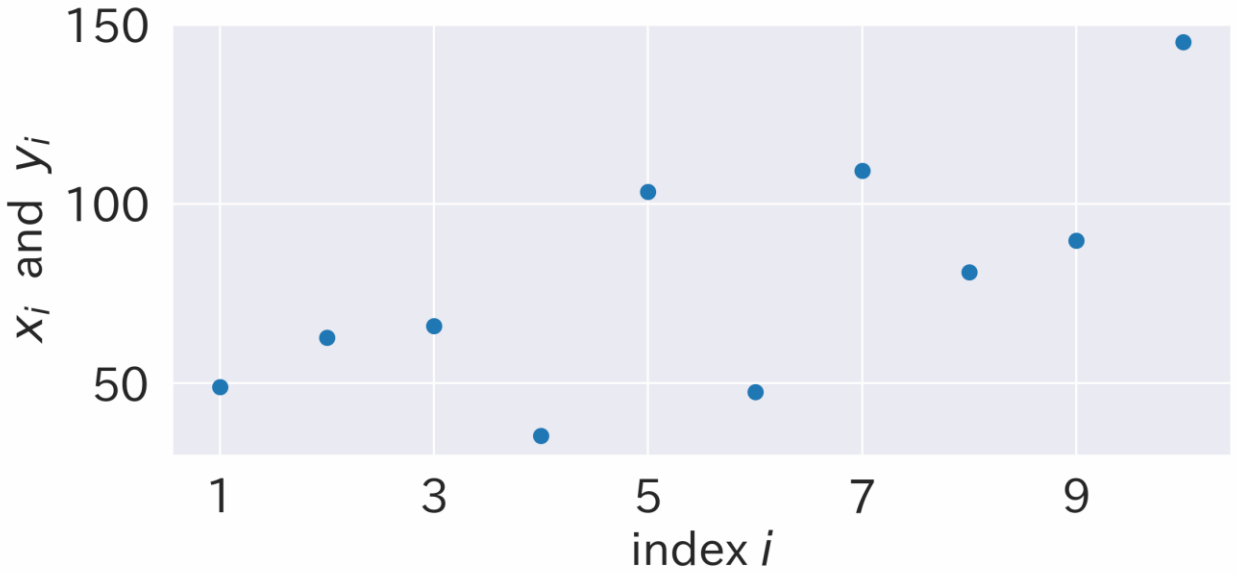
$$\text{s.t.} \quad x_1 \leq x_2 \leq \cdots \leq x_n$$

```
from sklearn.isotonic import IsotonicRegression
from scipy.optimize import isotonic_regression
```

- 錐最適化の一種 (やや特殊な例ですが)
- 確率的分類や麻酔学や毒性学に応用
- 実行可能領域はIsotonic Coneと呼ばれる錐
- PAVAと呼ばれる、割と面白いO(n)解法
- 1955年が初出だが、**R package** (OSS)は2022年にも更新されている。SciPyにも。



用量 – 反応曲線 By Jamgoodman — Own work, CC BY – SA 4.0

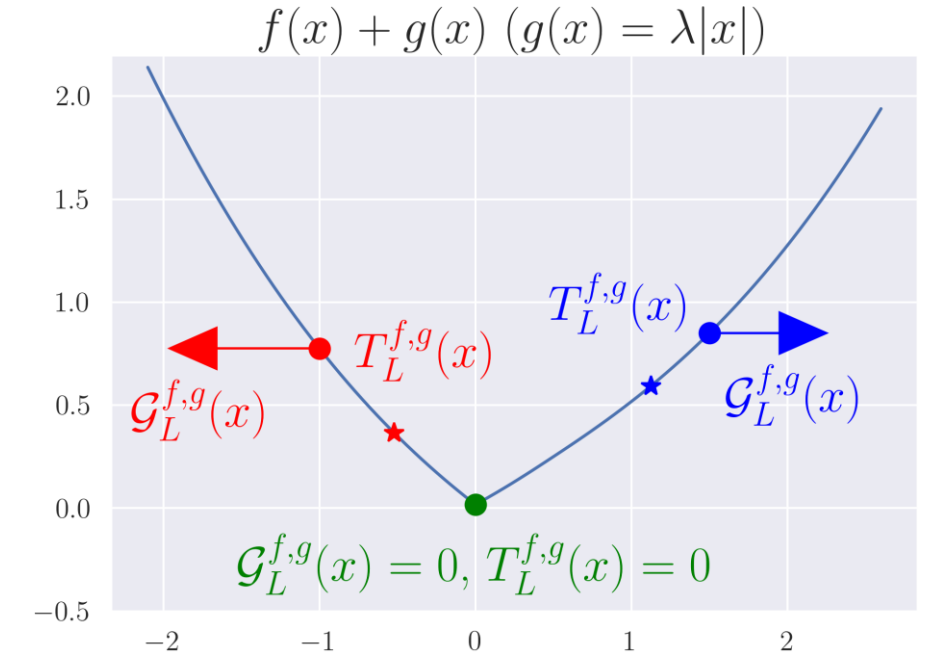


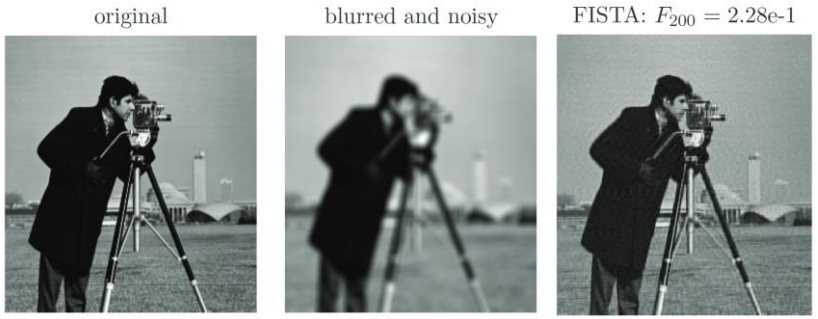
Isotonic Regression (単調回帰)と Pool Adjacent Violators Algorithm (PAVA)について

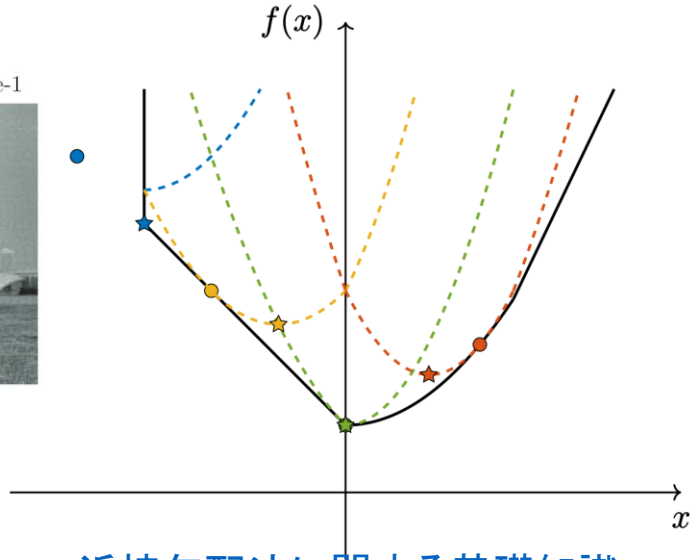
非平滑最適化問題

$$\mathrm{prox}_f(v) := \arg\min_{x \in \mathrm{R}^n} \left\{ f(x) + \frac{1}{2}\|x - v\|_2^2 \right\}$$

- かなり雑な説明ですが、関数がカクカクしていると一般に辛い
- 絶対値やそのベクトル版であるL1ノルムがその代表例
- 近接勾配法をはじめとする非平滑最適化手法
- 画像処理や信号処理に応用例
- **PyProximal**などのOSSが知られている



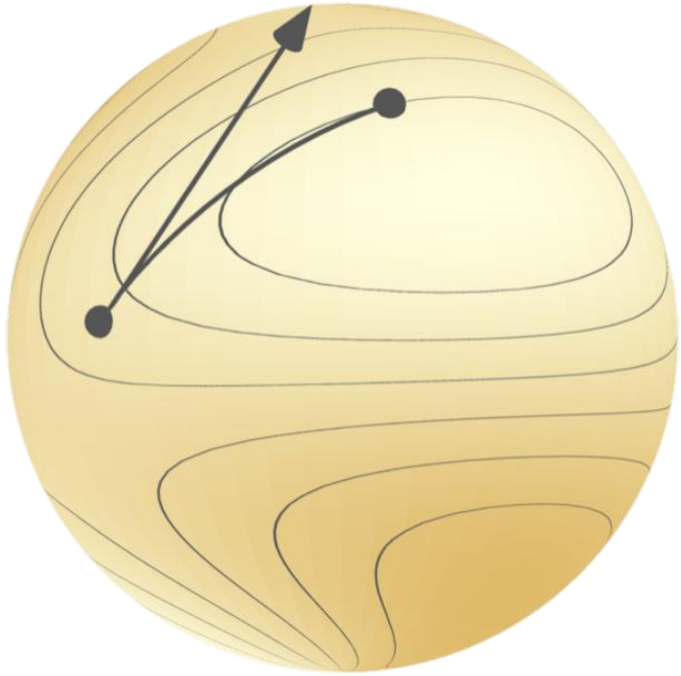




A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems
AMIR BECK AND MARC TEBOULLE

近接勾配法に関する基礎知識
浜口広樹

リーマン多様体上の最適化



- 一定のクラスの制約付き最適化問題を、リーマン多様体上の無制約最適化問題に
- かなり雑な説明ですが、球体の表面に張り付きながら最適化を行うイメージ
- 双曲空間上での機械学習などに応用
(私はよく存じ上げません……)
- **Manopt**というOSSが非常に有名
(Matlab, Python, Juliaで整備されている)

- Stiefel manifold
  - $\{X \in \mathbb{R}^{n\times p} : X^\top X = I_p\}^k$
- Grassmann manifold
  - $\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^\top X = I_p\}^k$
- Bounded-rank matrices
  - $\{X \in \mathbb{R}^{m\times n} : \mathrm{rank}(X) \leq r\}$
- Symmetric, positive definite matrices
  - $\{X \in \mathbb{R}^{n\times n} : X = X^\top \text{ and } X \succ 0\}$

参考文献: Riemann多様体上の連続最適化 Yoshinobu Ogura

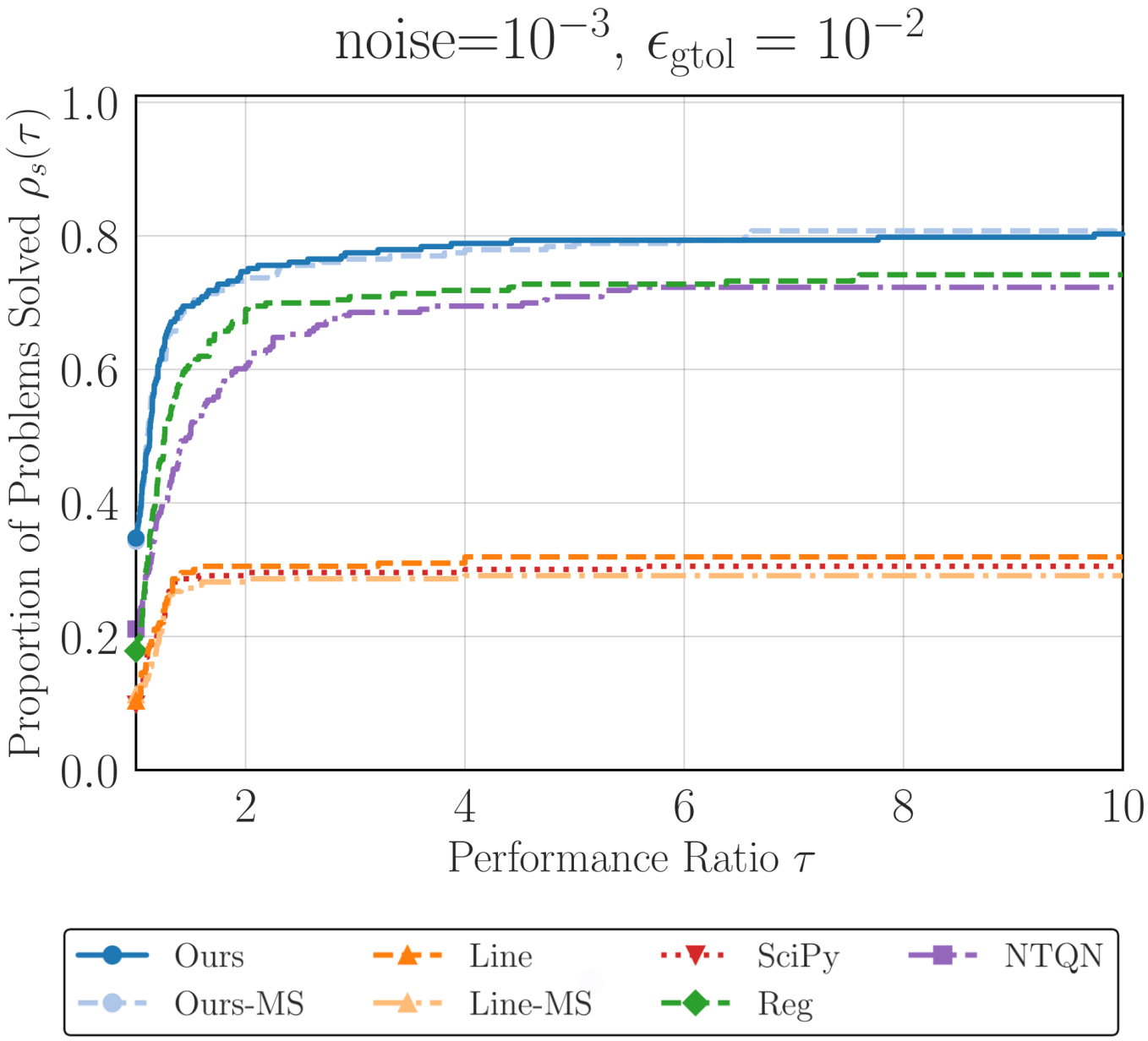
リーマン多様体上の最適化―特異値分解の例を通して― 冷めたコーヒー (はてなブログ)

余談: 直近の研究

詳細は省略しますが、
直近の研究はL-BFGS法自体の
改良を目指す。

特に数値誤差が大きいときに
大きく改善できる。

SciPyへの貢献を目指しています。
(いつか採用されると嬉しい)



まとめ

1. 連続最適化は逐次法がメイン
   L-BFGS法はSciPyも提供する高速な手法

2. NetworkXのdraw関数はFRモデルを使う
   L-BFGS法によって効率的に最適化可能

3. 多様な問題に対して多様な解法が存在

OSSは抽象的な数学や情報学と、具象的な現実の実装の橋渡しであると思います。
今後もその開発に携われたら嬉しいです。