

プログラミング応用 第7回: なんか「うまくいく」アルゴリズム

清水 伸高 (塩浦研 助教)

今日の内容

1. 理論と実用のギャップ
2. k-means法
 - 平滑化解析
3. 局所探索と焼きなまし法
 - 最適解への収束
 - ボルツマン分布

理論と実用のギャップ

問題

NP困難であっても実用上高速なアルゴリズムが多く知られている. なぜ?

- LPに対する単体法 (最悪の場合は指数時間かかるが多くの場合は高速)
- 充足可能性判定問題に対するSATソルバー
- 整数計画問題(IP)に対するIPソルバー
- 焼きなまし法
- k-means法
- ニューラルネットによる学習 (汎化誤差が小さい)

アルゴリズムの理論保証

- これまで扱ってきた効率性の指標: **最悪時**時間計算量
 - 計算量 $\leq T(n)$ \iff **全て** のサイズ n の入力に対する計算ステップ数が $\leq T(n)$

「いじわるな」入力が存在すれば, 最悪時時間計算量はその入力に引っ張られる.

- 「いじわるな」入力は**実用ではまず現れない**こともある
 - このようなとき, 「実用的な計算量」 \ll 「最悪時時間計算量」

問題

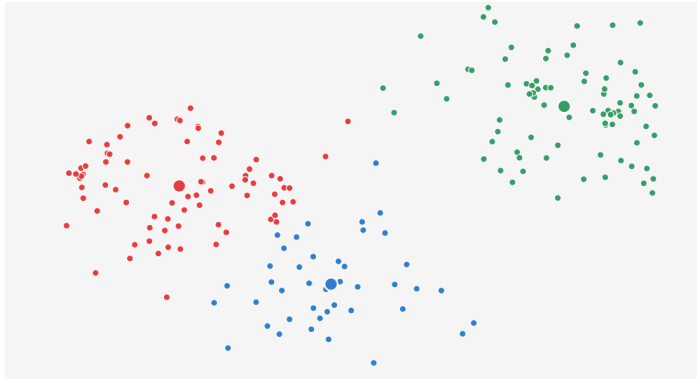
「実用的な計算量」はどのように定義すべきか?

- 専門的な内容ではあるが, 知っておく価値はとて高いので紹介したい

k-means法

- あらゆる分野で**高次元の数値データ**が得られる
 - 顧客行動ログ, 画像埋め込み, 文書ベクトル, etc
- これらのデータから学習して**将来予測**に活用したい
 - 購買の予想, 画像の生成(拡散モデル), 次の文の推測(Transformer), etc

- しかし多くの場合, **ラベル (正解)** が存在しない → **教師なし学習**
 - 画像データはあってもその画像が何なのか説明されることは少ない
- 似た性質を持つデータをまとめて潜在的なカテゴリや代表的な振る舞いや特徴を抽出する
 - この発想を **クラスタリング (clustering)** という



解きたい問題

- たくさんの点(データ点)を受け取り, 近い点同士でグループ化したい

問題

n 個の点 $x_1, \dots, x_n \in \mathbb{R}^d$ (データ点) を受け取り, k 個の点 $C = \{c_1, \dots, c_k\} \subseteq \mathbb{R}^d$ (中心点と呼ぶ) を出力せよ. ただし, 以下の目的関数を最小化したい:

$$\Phi(C) = \sum_{i=1}^n \min_{j \in [k]} \|x_i - c_j\|^2$$

- 各データ点を最も近い中心点に割り当てたときの二乗誤差の総和を最小化
- k も入力で与えられる (もしくは定数, 例えば $k = 3$ で固定)

目的関数の意味

$C = \{c_1, \dots, c_k\} \subseteq \mathbb{R}^d$ に対し,

$$\Phi(C) = \sum_{i=1}^n \min_{j \in [k]} \|x_i - c_j\|^2$$

Lloydのアルゴリズム

アルゴリズム

1. **初期化**: k 個の中心点 $C = \{c_1, \dots, c_k\}$ をランダムに初期化
 2. **割り当てステップ**: 各データ点 x_i を, 最も近い中心点に割り当てる (候補が複数ある場合は任意に一つ選ぶ)
 3. **更新ステップ**: 各 $c_j \in C$ について, c_j に割り当てられたデータ点の平均座標を計算し, c_j をその平均に更新する
 4. ステップ2と3を収束するまで (または所定の回数まで) 繰り返す
- 初期点の選び方にも工夫の余地がある → **k-means++**
 - データ点の次元が大きい場合, 前処理で工夫して高速化の余地がある
 - PCA (主成分分析): 大きい固有値に対応する固有空間への射影
 - Johnson-Lindenstraussの次元削減: ランダムな射影

Lloydのアルゴリズム

問題

- 最適解が得られる？
- アルゴリズムの反復回数は？

定理

- $\Phi(C)$ を最小化する C の計算は**NP困難**
- 反復回数が**指数時間かかる**データ点配置の例が存在



- 「いじわるな」データ点配置が存在
- 最悪時計算量の意味では困難

最悪時以外の計算量解析

最悪時計算量と実用性とのギャップをなんとかしたい.

- **平均時計算量** (Average-Case Complexity)
 - **入力は何らかの分布に従って生成される**と仮定 (ランダムな入力を考える)
 - 「平均的な」計算量を解析 (例えば期待値) → 「いじわるな」入力例には引っ張られない
 - 最適化の多くの論文はランダムに生成した入力上で計算機実験している

特定の分布でうまくいくことが保証されるに過ぎない. 多くの場合は入力分布が不明なので, アルゴリズムの実用性を捉えるかは疑問が残る.

最悪時以外の計算量解析

- **平滑化解析** (Smoothed Analysis)
 - いじわるな入力に **小さい「ノイズ」** を加えて得られた入力上での計算量を評価
 - 各入力の周辺だけで期待値をとる
- ノイズの分散が小さい → 最悪時計算量
- ノイズの分散が大きい → 平均時計算量

ノイズの分散: **0.0100**



平滑化解析 (Spielman & Teng, 2001)

- 最悪時だと指数時間であっても, 平滑化では多項式時間になる
 - 単体法 (LP)
 - k-means (クラスタリング)
 - 最大カットに対する局所探索
- 現状, 平滑化解析が成功している例はまだ少ない (研究の余地あり)
 - 平滑化解析の意味で多項式時間であることは示されていても, 多項式のオーダーはとて大きい ($O(n^{30})$ 時間など)

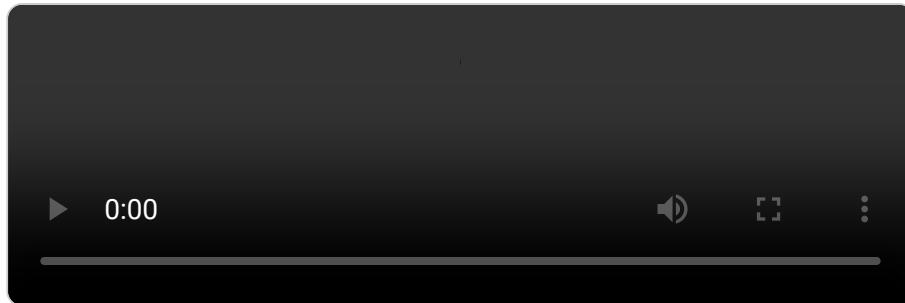
局所探索と焼きなまし法

局所探索

初期解からスタートし, 少しずつ解を更新して最適解に近づいていく方針に基づくアルゴリズムを総称して **局所探索** と呼ぶ.

- 勾配法
- ポテンシャルの更新に基づく最短経路問題の解法
- k-means法

k-meansのように, 必ずしも最適解にしくなくても「良い」解が得られることもある.



近似アルゴリズムとヒューリスティクス

実用的には、最適解でなくとも **目的関数値が十分良い解で事足りる** 場面が多い。

- 例えば、ナップザック問題や巡回セールスマン問題では、必ずしも最適解でなくとも、「十分良い」解があればよい場合が多い
- このような場合、**局所探索**は一つの有力なアプローチとなる
 - 好きなタイミングで停止できるので、十分反復を繰り返したあとの解を出力すればよい

注釈

最適解を求めることがNP困難であっても、「十分良い」解は効率的に求められるかもしれない

- 出力した解の質に理論的な保証があるアルゴリズム → **近似アルゴリズム**
- 理論的な保証はないが、実用上よく機能するアルゴリズム → **ヒューリスティクス (発見的解法)**

巡回セールスマン問題に対する局所探索(2-opt法)

問題

入力: 距離関数 $w: V \times V \rightarrow \mathbb{R}_{\geq 0}$ (ただし $w(u, u) = 0$)

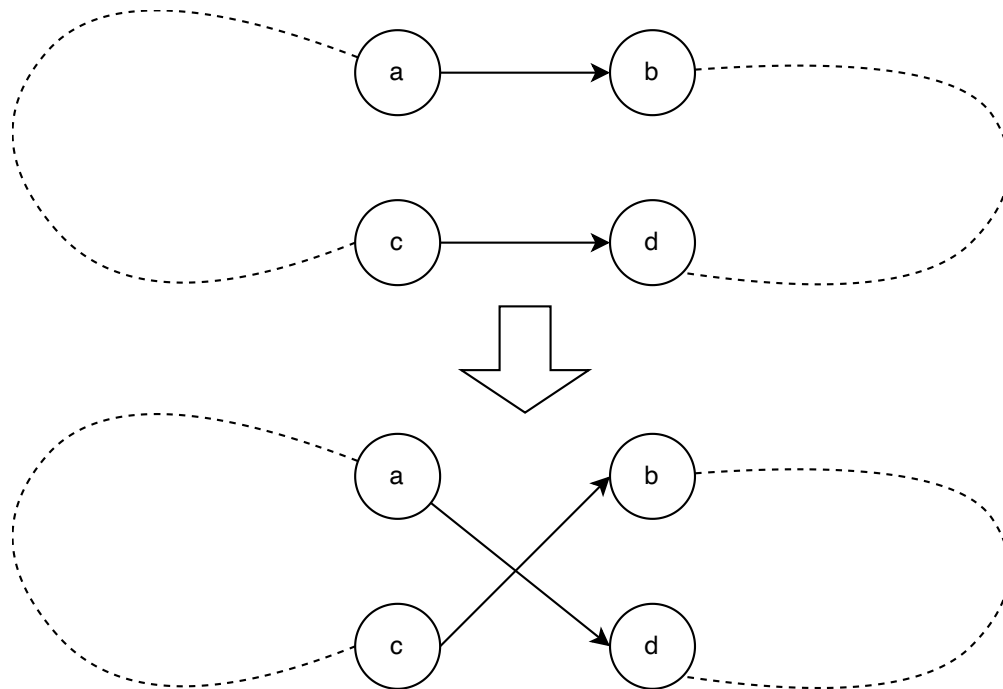
出力: 全ての頂点を一度ずつ訪れる巡回路 P であって, 総距離 $w(P) := \sum_{(u,v) \in P} w(u, v)$ を最小化

- 次のような局所探索アルゴリズム(2-opt法)が知られている:

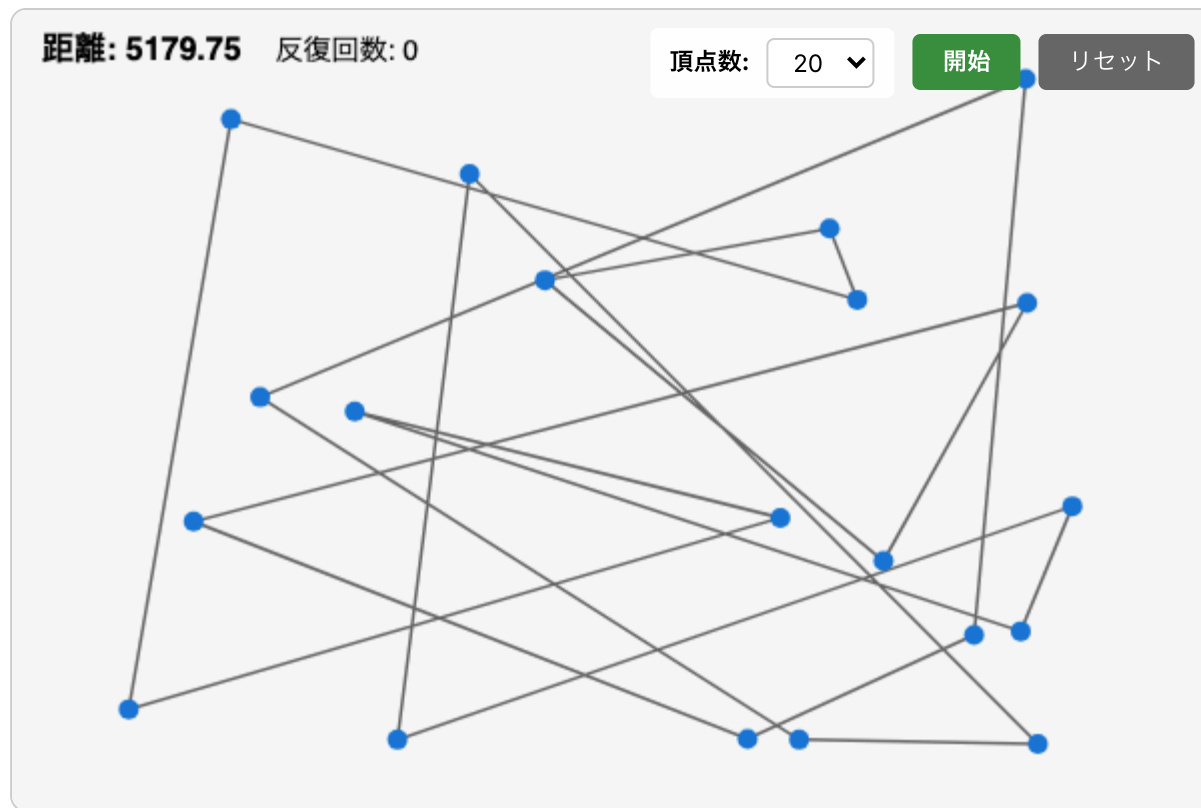
アルゴリズム

1. 初期巡回路 P を適当に構成
2. P の2つの辺を順序つきで選んでそれぞれ $(a, b), (c, d)$ とする. P から辺 $\{a, b\}, \{c, d\}$ を削除し, 新たに辺 $\{a, d\}, \{c, b\}$ を追加して得られる巡回路 P' を考える (ただし P' も巡回路になるようにする)
3. もし $w(P') < w(P)$ ならば, $P \leftarrow P'$ と更新し, ステップ2に戻る

巡回セールスマン問題に対する局所探索(2-opt法)

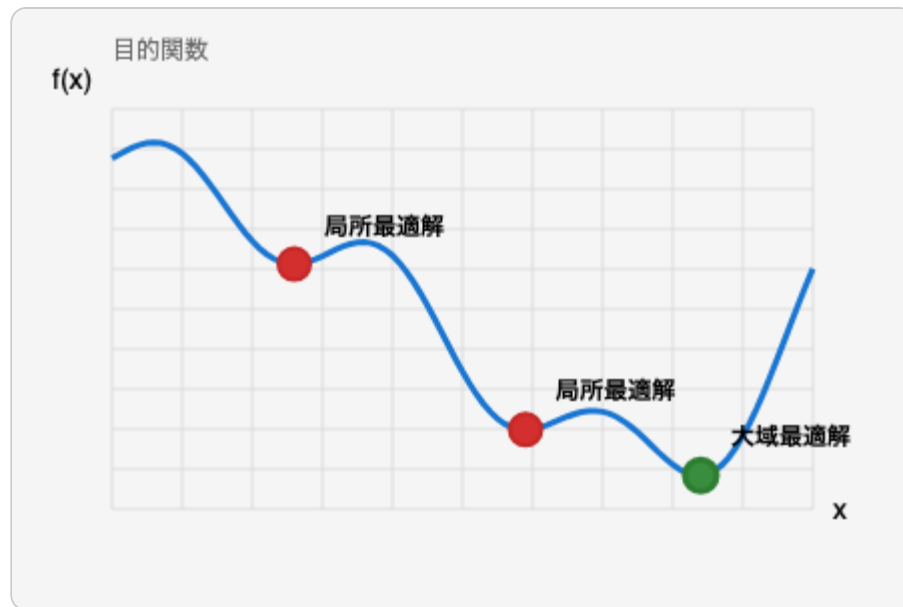


2-opt法の可視化



局所最適解と大域最適解

- 2-opt法の反復を繰り返すと, これ以上改善できない巡回路 P に到達する
 - これを**局所最適解**という
- 対して最適解を**大域最適解**という
- 局所最適解が**必ずしも最適解とは限らない**ことに注意



局所最適解から脱出するには一旦「山を登る」 (=目的関数値が悪化する操作を行う) 必要がある

焼きなまし法 (Simulated Annealing)

- 焼きなまし法: 目的関数値が悪化する操作も **確率的に** 受け入れる
 - **確率 θ** で $P \leftarrow P'$ と更新
 - ただし θ は **現在時刻 t** と **改善幅 $w(P') - w(P)$** に依存する
- 時間とともに 許容確率 θ を小さくしていく
 - 時間が経つにつれて改善操作のみを受け入れるようになっていく

2-opt法とは異なり, 悪くなる操作も受け入れることで, 局所最適解から脱出できる可能性がある.

焼きなまし法の詳細

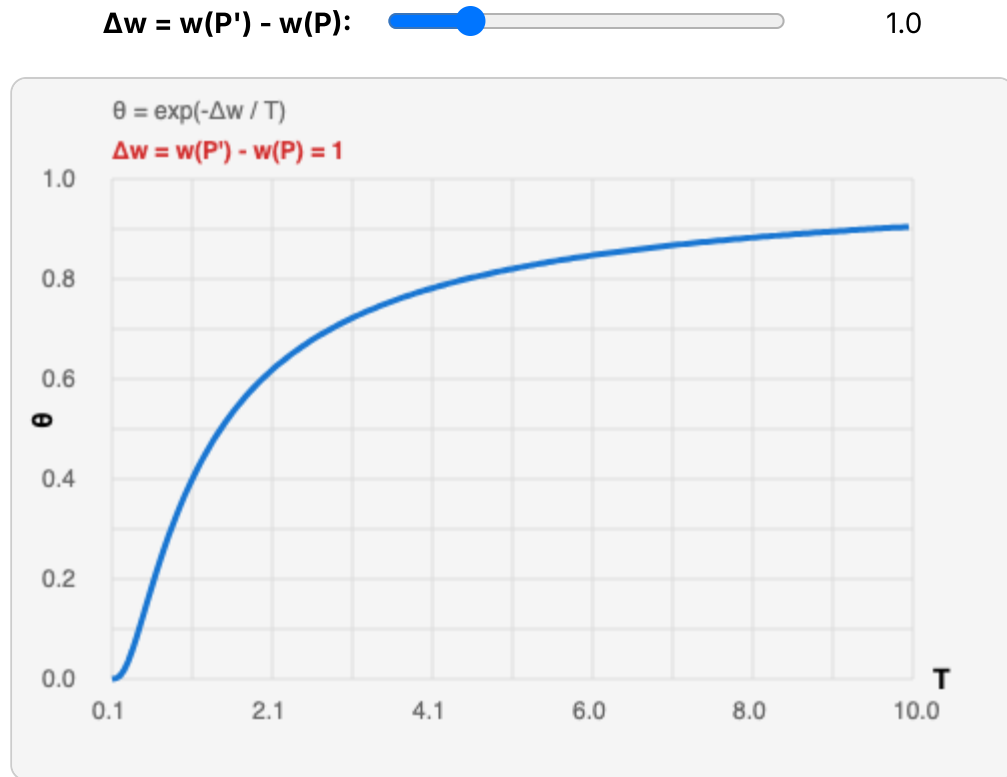
- TSPの例では, 次のようになる:

アルゴリズム

1. 初期巡回路 P を適当に構成し, 時刻 $t \leftarrow 0$ で初期化
 2. P の2つの辺をランダムに選び, それらの一つの端点を入れ替えることで得られる巡回路 P' を考える
 3. 確率 $\min \left\{ 1, \exp \left(-\frac{w(P') - w(P)}{T(t)} \right) \right\}$ で $P \leftarrow P'$ と更新
 4. $t \leftarrow t + 1$ とし, ステップ2に戻る
- $T(t)$ は 温度関数 と呼ばれ, 時刻 t とともに減少する関数
 - 例えば, $T(t) = \frac{T_0}{\log(2+t)}$ など (T_0 は大きな定数)

更新確率の可視化

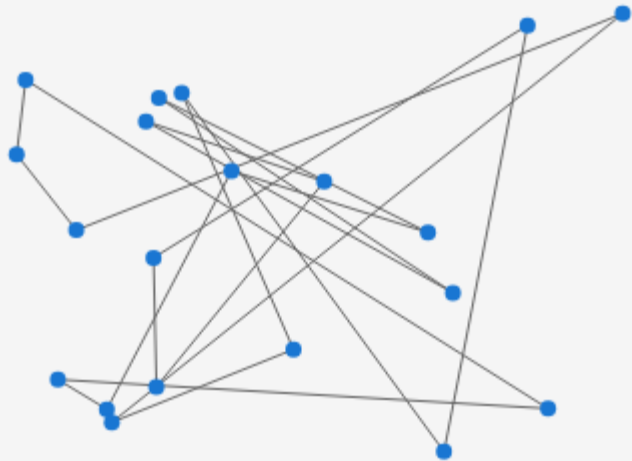
- 更新確率 $\theta = \min \left\{ 1, \exp \left(-\frac{\Delta w}{T} \right) \right\}$
 - $\Delta w = w(P') - w(P)$ を固定値として設定可能
 - パラメータ T を横軸、 θ を縦軸としてプロット
- T が大きいほど θ が大きくなる（悪化操作も受け入れやすい）
- T が小さいほど θ が小さくなる（改善操作のみ受け入れる傾向）
- $\Delta w < 0$ ならば $\theta = 1 \rightarrow$ 必ず更新



2-opt vs. 焼きなまし法

2-opt法

距離: 3215.86 反復回数: 0



頂点数:

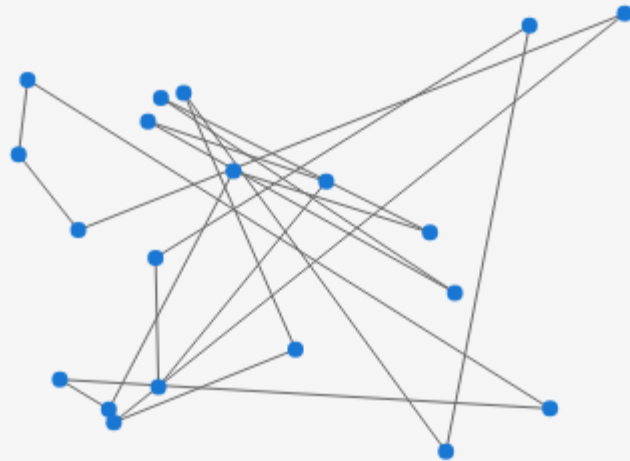
20 ▼

温度スケジューリング:

対数スケール ($T(t) = T_0 / \log(2+t)$) ▼

焼きなまし法

距離: 3215.86 反復回数: 0



2-opt vs. 焼きなまし法

- 時刻 t が小さいところでは, 2-optの方が改善して焼きなまし法は改悪しやすい
- t を大きくすると焼きなまし法の方が良い解に収束することが多い (反復回数を大きくすることが大事)
- 実は焼きなまし法は **最適解** への収束が保証されてる

定理

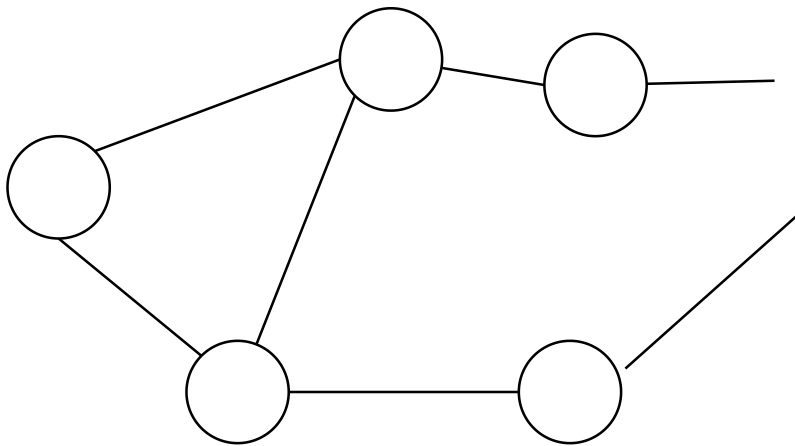
焼きなまし法は**十分にゆっくり温度を下げれば**, 最適解に収束する. 具体的には, 十分大きな $T_0 > 0$ に対し

$$T(t) = \frac{T_0}{\log(2 + t)}$$

とすれば, $t \rightarrow \infty$ のとき, 確率1で最適解に収束する.

直感的な説明

- TSPの局所探索は**全ての順回路 P 全体の集合を頂点集合とみなした**メタ的なグラフ上の探索とみなせる
 - メタグラフの頂点数は $(n - 1)!/2$ 個 (巡回路の数)
 - 二つの頂点 P, P' が隣接 \iff 2-optの更新で行き来できる



直感的な説明

- 焼きなまし法はメタグラフ上の**ランダムウォーク**とみなせる
 - ランダムウォーク: 各ステップで隣接頂点にランダムに移動する乱択探索
 - 焼きなまし法: ランダムな2辺を選択 → 受理確率に基づき遷移
- メタグラフの各頂点(=順回路) P に対し以下の値を考える:

$$\mu_t(P) = \frac{\text{ランダムウォークが時刻 } t \text{ までに頂点 } P \text{ を訪問した回数}}{t}$$

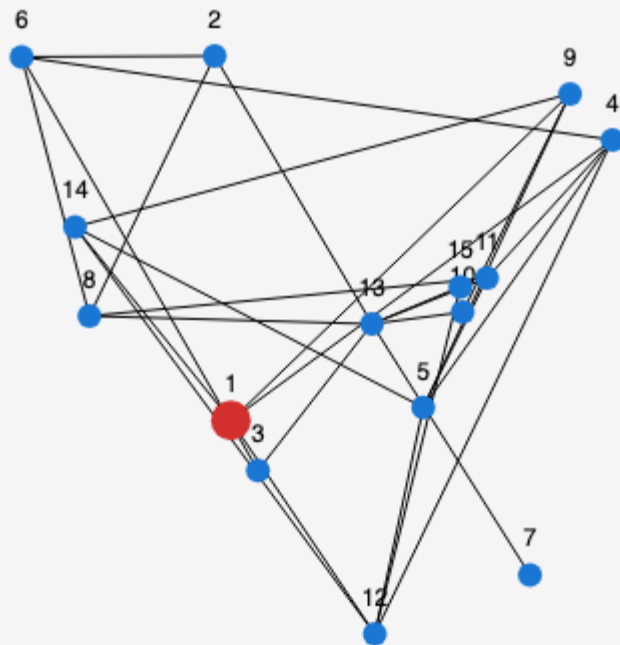
- $\mu_t(P)$ が一定の場合大きい \Rightarrow ランダムウォークが P を訪問しやすい

注釈

$\mu_t(P) \geq 0$ かつ $\sum_P \mu_t(P) = 1$ なので, 各 μ_t は順回路全体の集合上の分布を定める

ランダムウォークのシミュレーション

時刻: 0



時刻: 0

$\mu_t(i)$

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

頂点番号

ランダムウォークの定常分布

定理

温度 T が一定の場合, ある分布 π (定常分布) が存在して, 焼きなまし法の任意の初期巡回路 P_0 に対し,

$$\lim_{t \rightarrow \infty} \mu_t(P) = \pi(P).$$

- メタグラフが連結ならランダムウォークは収束する
- 温度を固定して焼きなまし法を動かすと, 十分に時間が経った後の巡回路 P の分布はほぼ π
 - 実際には温度を非常にゆっくり動かすので, T を固定した状況とほぼ同じ
- 定常分布 π は以下を満たす (ボルツマン分布という):

$$\pi(P) \propto \exp\left(-\frac{w(P)}{T}\right)$$

焼きなまし法の定常分布

- 焼きなまし法では

$$\mu_t(P) \rightarrow \pi(P) \propto \exp\left(-\frac{w(P)}{T}\right)$$

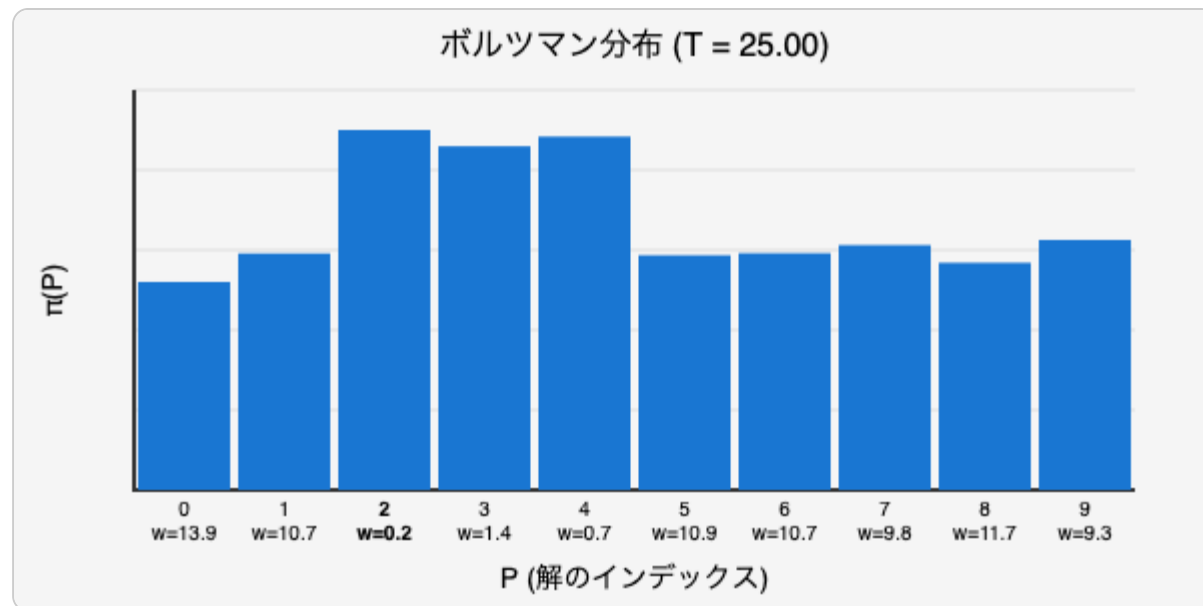
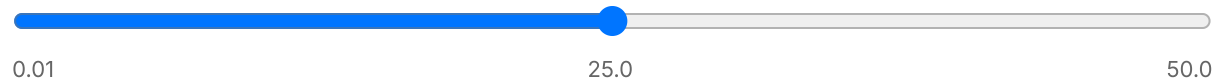
- つまり, 順回路 P の訪問回数は $\exp\left(-\frac{w(P)}{T}\right)$ に比例
 - $w(P)$ が小さいほど訪問回数が多い
 - $T \rightarrow \infty$ のとき, 一様分布に収束 (温度が高い状態)

問題

$T \rightarrow 0$ のとき, 定常分布 π はどうなる?

焼きなまし法の定常分布

温度 T: 25.00



焼きなまし法の定常分布

- P が最適解じゃないならば, $\pi(P) \rightarrow 0$
 - よって, $T \rightarrow 0$ の極限では定常分布 $\pi(P)$ は**最適解上の一様分布**となる
- ただし, $T = 0$ としてしまうと2-opt法と一致
 - ランダムウォークの収束性が失われる
 - 実際, **初期温度が十分高くないと**最適解への収束が保証できない

注釈

実際には **T を動かしながらランダムウォークを行う** が, その変化は非常にゆっくりなので, 温度 T での定常分布 π_T に収束する. しかしその証明はとても難しい.

焼きなまし法のまとめ

- 焼きなまし法では, 温度パラメータによって二つの性質を緩やかに変化させられる
 - 高温状態 \approx メタグラフ上の一様ランダムな隣接点への遷移
 - 低音状態 \approx 2-opt法
- 温度変化がゆっくりだと, それぞれの温度帯で訪問回数が定常分布(ボルツマン分布)に収束
 - 低音状態では, 最適解上に確率が集中する

注釈

最適解への収束性は保証されているが, その収束のスピードは非常に遅いため, 実用では**温度は早く下げる**ことが多い (それでも経験上, 良い解が得られることが多い).

- 理論保証: $T(t) = \frac{T_0}{\log(2+t)}$ (T_0 は十分大きな値)
- 現実: $T(t) = \frac{T_0}{t}$

メタヒューリスティクス

- 焼きなまし法はTSP以外にも様々な問題に対して応用可能
 - 原理的には「メタグラフ」上での遷移が定義できれば実装可能

注釈

このような, 多くの問題に汎用的に適用できる発見的解法を **メタヒューリスティクス** と呼ぶ.

- 他にも **遺伝的アルゴリズム**, **タブーサーチ**, **アリコロニー最適化**, **粒子群最適化** など
- これらは全て, 局所探索の枠組みを拡張したものであるが, **理論保証がない**
- 個人的には, 少なくとも研究の場面では, 複雑で凄そうなアルゴリズムを使うよりも, まずは単純な局所探索や焼きなまし法で試すことをお勧めする
 - 「これじゃなきゃいけない」理由を説明できないアルゴリズムは避けるべき

今日のまとめ

1. 理論と実用のギャップ

- 最悪時計算量は「いじわるな」入力に引っ張られる
- 実用的な計算量を捉えるための考え方: 平均時計算量, 平滑化解析

2. k-means法

- Lloydのアルゴリズムは最悪時では指数時間かかるが、平滑化解析の意味では多項式時間
- 平滑化解析: いじわるな入力にノイズを加えた上での計算量を評価

3. 局所探索と焼きなまし法

- 局所探索: 初期解から少しずつ解を更新して最適解に近づく
- 焼きなまし法: 悪化する操作も確率的に受け入れることで局所最適解から脱出
- 温度を十分ゆっくり下げれば最適解に収束することが保証される (ボルツマン分布への収束)