プログラム設計とアルゴリズム
第11回 (12/6)

早稲田大学高等研究所　講師
福永津嵩

(前回の復習)最小全域木問題

- 重みつき無向グラフG=(V, E)が与えられているとする。
  Gの部分グラフであって木であり、かつGの全ての頂点を含むものを
  全域木と呼ぶ。
- 全域木Tの重みは、それに含まれる辺の重みの総和とする。
  この時、最小全域木問題とは、重みが最小の全域木を求める問題である。

図15.1

(前回の復習)クラスカル法

- 単純な貪欲法によって、最小全域木を求めることが可能である。

$\boxed{\text{最小全域木を求めるクラスカル法}}$

(前回の復習)クラスカル法

図15.2

(前回の復習)グラフの基本用語:カット

- グラフG=(V, E)のカットとは、頂点集合XとYへのVの分割を意味する。ただし、X∪Y = V、X∩Y = ϕ とする。

- XとYにまたがる辺をカット辺と呼び、カット辺全体の集合をカットセットと呼ぶ。

図15.3

(前回の復習) 辺連結度

- ２頂点sからtへのパスにおいて、辺を共有しないs-tパスの最大値を辺連結度と呼ぶ。また、互いに辺を共有しないことを辺素と呼ぶ。

図16.1

(前回の復習)カットの容量

- カット(S, T)においてカットセットに含まれる辺の本数をカット(S, T)の容量と呼び、c(S, T)と表す。

図16.2

(前回の復習)最小カット問題

- $s \in S$、$t \in T$を満たすカットをs-tカットと呼ぶ。

$\boxed{\text{最小カット問題(辺の容量が1の場合)}}$

- グラフGから辺を数本取り除いてs-t間を分断する時、
  取り除く最小の本数を求めているとも言える。

(前回の復習)辺連結度に関する強双対性

- ある主問題の最小値が、その補問題(双対問題)の最大値になるとき、強双対性が成り立つと呼ぶ。

- 辺連結度については、次の強双対性が成り立つ

辺連結度に関する問題の強双対性

(前回の復習)残余グラフ

- 残余グラフを考えることで効率的に問題を解くことが出来る。
- 残余グラフとは、通ったs-tパスの辺の向きを逆方向にしたグラフである。

図16.5

(前回の復習)辺連結度を求めるアルゴリズム

- 辺連結度を求めるアルゴリズムは次の通りとなる。

２頂点s-t間の辺連結度を求めるアルゴリズム

(前回の復習)応用例: 二部マッチング問題

- 二部グラフが与えられたときに、２カテゴリ間で最大何組のペアを作ることができるかという問題。
- 新たに頂点sとtを追加し、また向きも一方向に定めることで、辺連結度を求める問題と同様の問題となる。

図16.15

第十七章
PとNP

問題の難しさ

- ある問題はアルゴリズムを工夫することで線形時間で解けるが、別の問題はどう頑張っても指数時間のアルゴリズムしか構成できないかもしれない。

- 問題の難しさを考えるときは、まず多項式時間で解けるかどうかが重要なポイントとなる。よって、問題の考え方は次の通り

  - 多項式時間アルゴリズムを与えます (それができれば，次は計算量を少しでも改善します)
  - 多項式時間アルゴリズムでは解けそうもないことを示します

多項式時間帰着

- 多項式時間では解けない(だろう)証明は、次のように行われる

$\boxed{\text{多項式時間帰着}}$

- よって、多項式時間で解けない(だろう)Y以上にXが難しいことを示す。

多項式時間帰着

図17.1

問題のクラス

- 問題の難しさに応じて、問題をクラス分けする。ここでは、判定問題(YesかNoで答えられる問題)のクラスを考える。

- 最適化問題(最大値や最小値を求める問題)は判定問題ではないが、最適化問題を判定問題に変更できることは多い。

- 「~~という場合の最小値を求めよ」
→
「~~という場合の最小値がWを超えるかどうかを判定せよ」

クラスP

- クラスP:
  多項式時間アルゴリズムが存在する判定問題の集合

  例:
  1. 与えられた配列の中に特定の値が存在するかを判定する。
  2. 与えられた無向グラフが二部グラフであるかを判定する。

- 一方、次に紹介するハミルトンサイクル問題は、
  多項式時間アルゴリズムは見つかっていないが、多項式時間アルゴリズムが存在しないことも証明されていない。

  つまり、クラスPかどうかはまだわかっていない。

ハミルトンサイクル問題

ハミルトンサイクル問題

図17.2右部

- なお、全ての「辺」をひとつずつ含む(同一の頂点を何度通っても良い)サイクルはオイラーサイクルと呼ばれる。
- オイラーサイクルの判定問題はクラスPである。

クラスNP

- クラスNP:
判定問題の答えがYesならば、そのYesである証拠を与えると、
それがYesであることが多項式時間で検証できる判定問題の集合

- クラスPに属する問題は、明らかにクラスNPである。

- また、ハミルトンサイクル問題は、もし答えがYesであり、その答えであるハミルトンサイクルが与えられたならば、全ての頂点を通っていることは多項式時間で検証できるので、クラスNPである。

クラスNPの誤解と補足

- NP = not Pと誤解されることが多いが、NPはPを含むので、明らかに誤りである。

- NPは指数時間アルゴリズムで解ける問題のクラスであるとも誤解されるが、それはEXPであり、NPを含むクラスである。

- Pは
  「決定性チューリングマシンにより多項式時間で解ける判別問題」
  NPは
  「非決定性チューリングマシンにより多項式時間で解ける判別問題」
  とも言える。この意味を理解するためには、オートマトン理論を勉強する必要がある。

P≠NP予想

- クラスNPとクラスPが同一の集合なのか、同一ではないのかはまだわかっていない。

- P = NPとはつまり、「Yesの証拠が与えられたときに多項式時間でYesであることを判定できる(クラスNPに属する)ならば、そもそも判定問題を多項式時間で解くことが出来る(クラスPに属する)」という主張。

- P ≠NPとはつまり、「クラスNPに属する問題でも、その判定問題を多項式時間では解けない問題が存在する。」という主張。

P≠NP予想

- つまり、ハミルトンサイクル問題が多項式時間では解けないことを証明したら、P≠NPの証明が成り立つ。
- これは現代数学において最も重要な未解決問題の一つであり、100万ドルの賞金がかけられている「ミレニアム7問題」の一つである。
- 多くの人がP≠NPであろう、とは予測している。

NP完全

- クラスNPに所属する問題の中で、最も難しい問題のクラスをNP完全と呼ぶ。

NP完全

- ハミルトンサイクル問題は実はNP完全問題である。

NP困難

- 判定問題以外の問題について問題のクラスを考えたとき、
  NP完全問題に多項式時間帰着できる問題のクラスをNP困難と呼ぶ。

$\boxed{\text{NP困難}}$

- つまり、NP困難問題は、NP完全問題と同等かそれ以上に難しい
  問題となっている。

巡回セールスマン問題(TSP)

- NP困難である問題の一つとして、非常に有名な問題

巡回セールスマン問題(TSP)

- 頂点を都市、枝の重みを移動にかかる時間として「全ての都市を一度ずつ周り、そのうち最短の時間になる経路」を求める問題として知られる

- ハミルトンサイクル問題に多項式時間帰着できる。

第十八章
難問対策

メタヒューリスティクス

- NP困難な問題にたいして最適解を求める事は難しいことが多い。
一方実用上は、近似的に最適であれば十分であることも多い。
近似的に最適な解を発見する手法をヒューリスティクスという。

- メタヒューリスティクスとは、多様なNP困難問題に対して、
形式上どのような問題に対しても利用可能なヒューリスティクス
のことをいう。

- 焼きなまし法、タブー探索、遺伝的アルゴリズムなど
様々な手法が提案されている。

山登り法(TSP)

1. まず最初にランダムに初期解xを生成し、そのスコアf(x)をbest scoreとする。

   (全ての都市を回るルートをランダムに生成し、その時間をスコアとする)

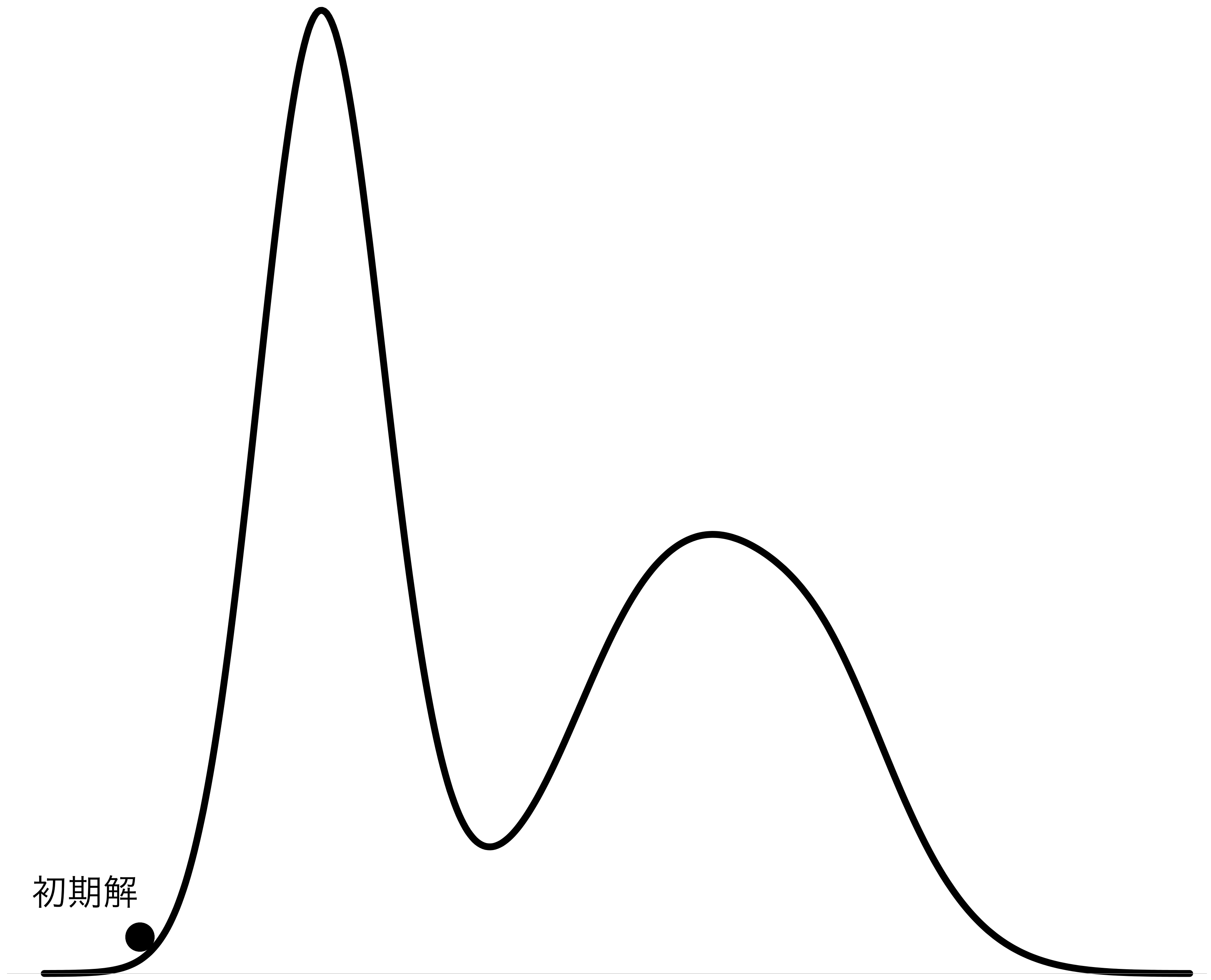
2. xの近傍x'の中で、最もスコアの良いx'に注目する。f(x')がbest scoreよりも良ければ、解とbest scoreを更新する。

   (都市を回る順序を微妙に入れ替えて、より良いルートがあればそちらにルートを更新する。近傍の定義は自由だが、A→B→C→D→Eの近傍は、D→B→C→A→EやA→E→C→D→Bなどと定義するのが一つのやり方)

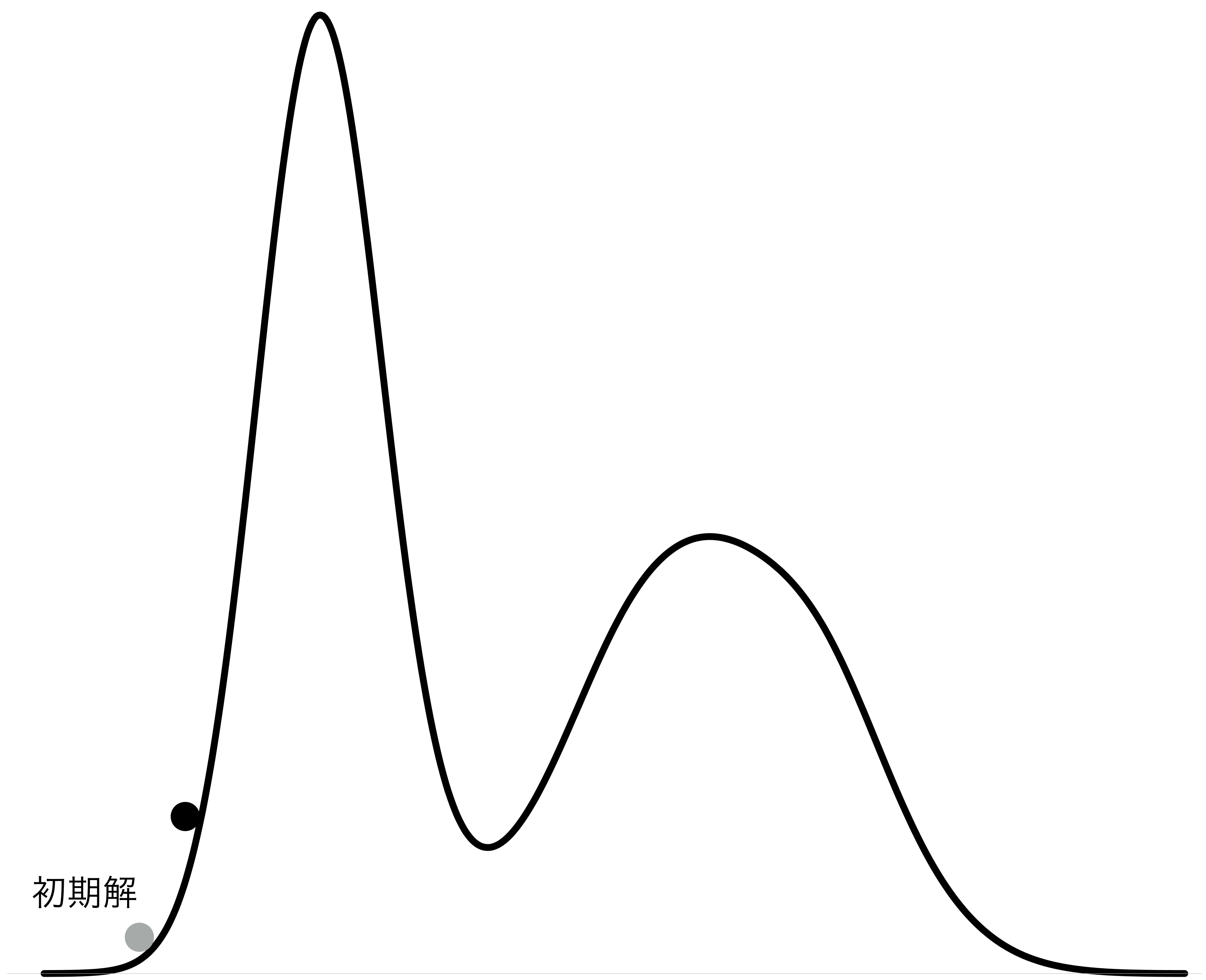
3. 2を繰り返す。

4. 解の更新が終了したら、探索を終了する。

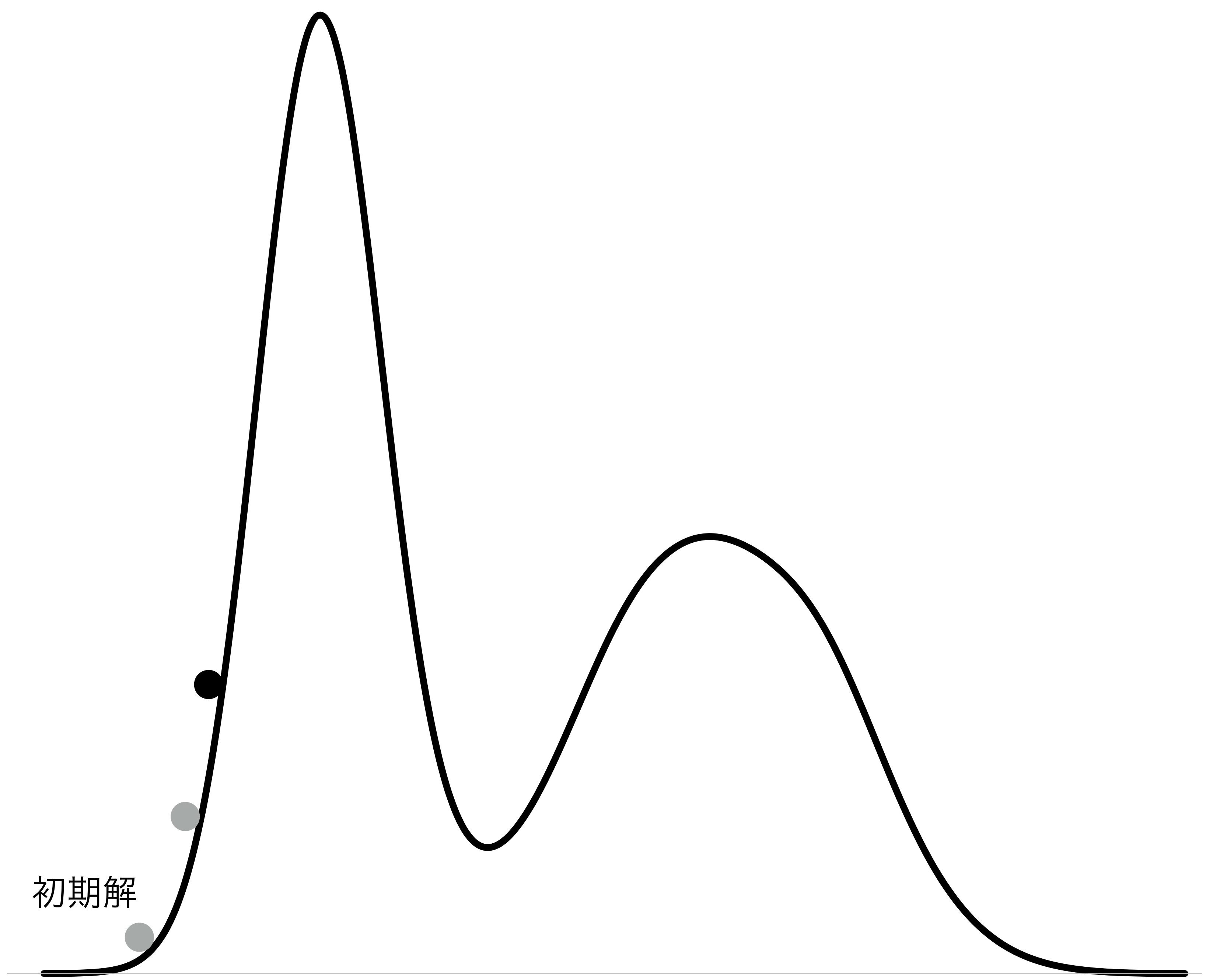
山登り法のイメージ図



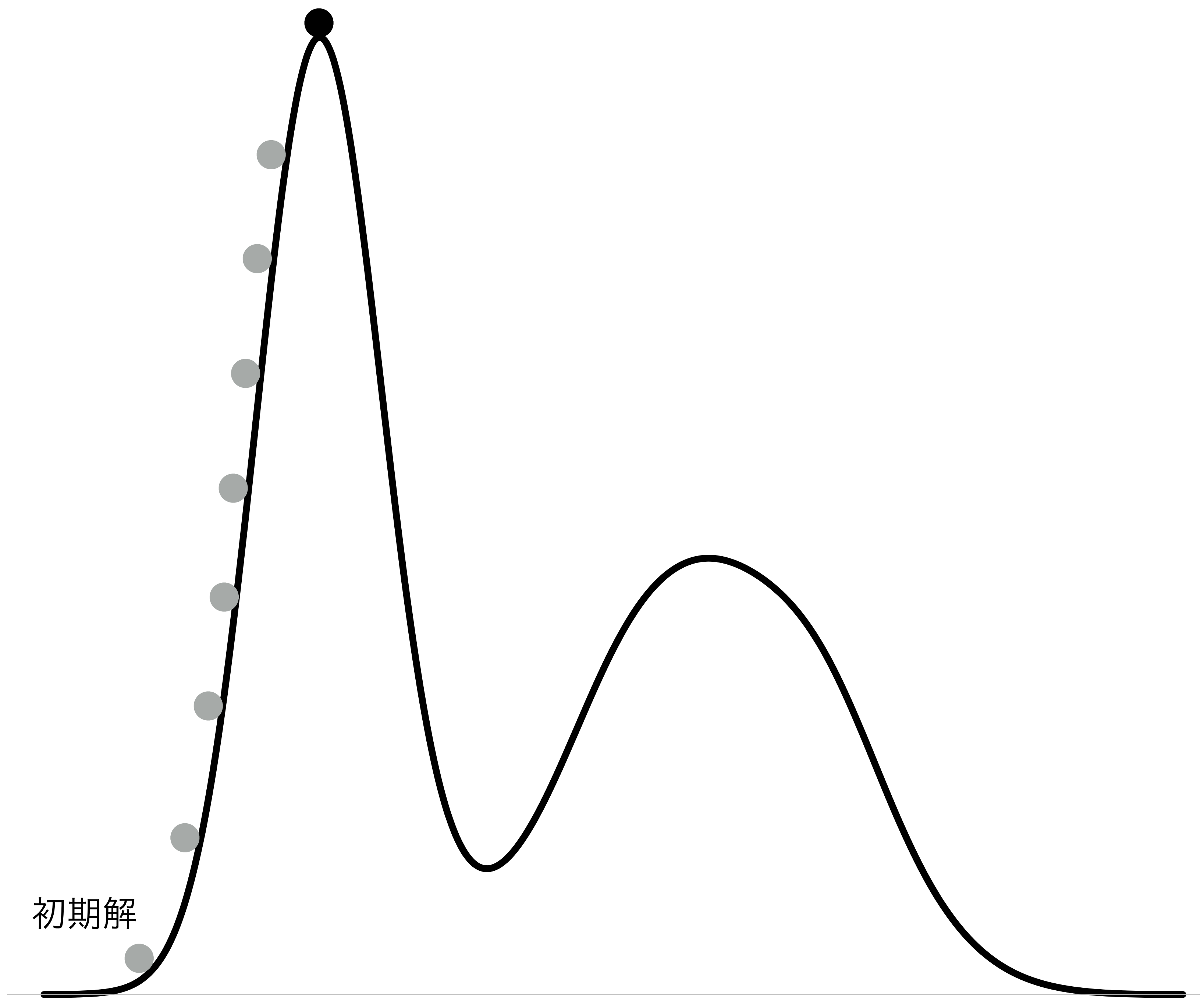
山登り法のイメージ図



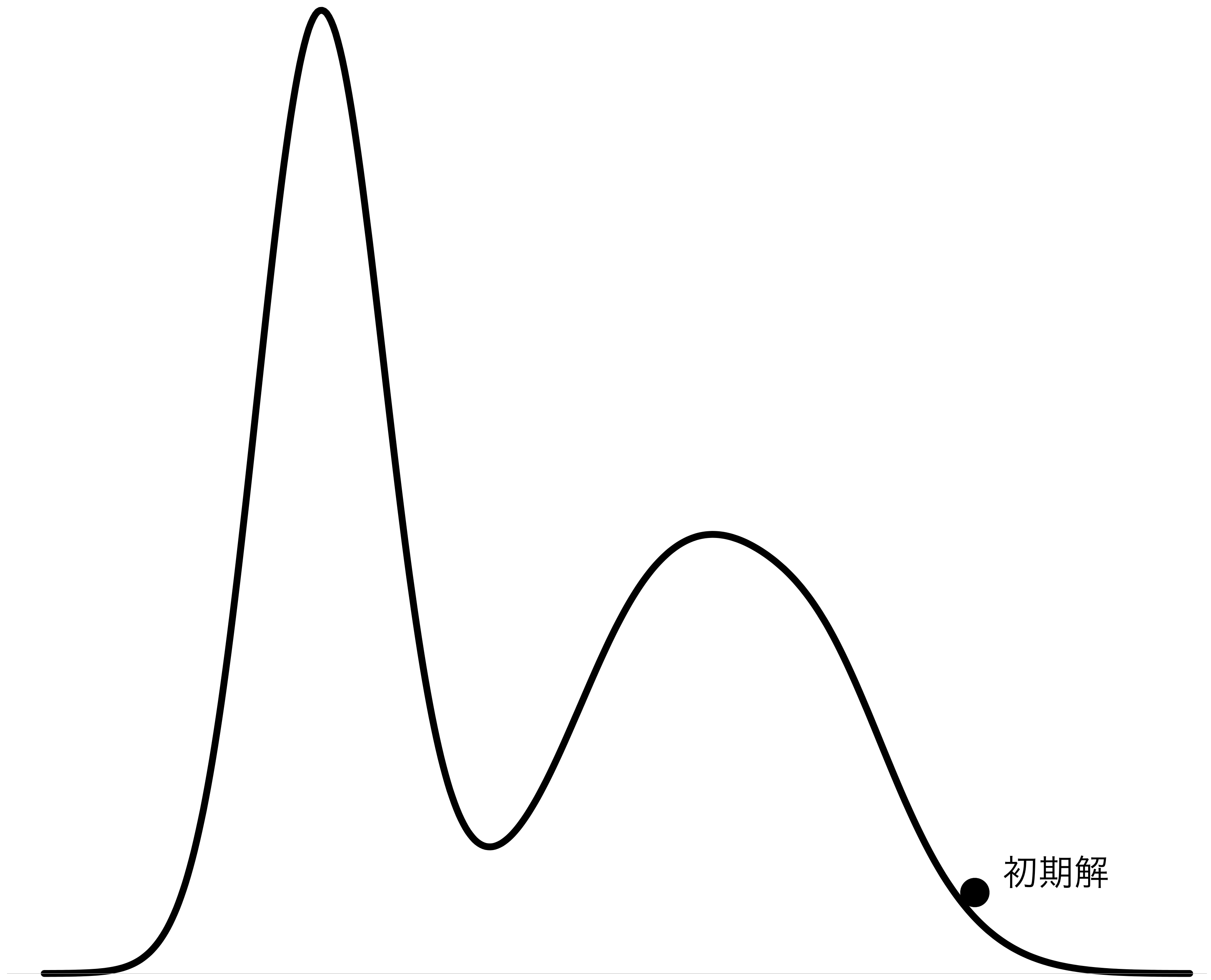
山登り法のイメージ図



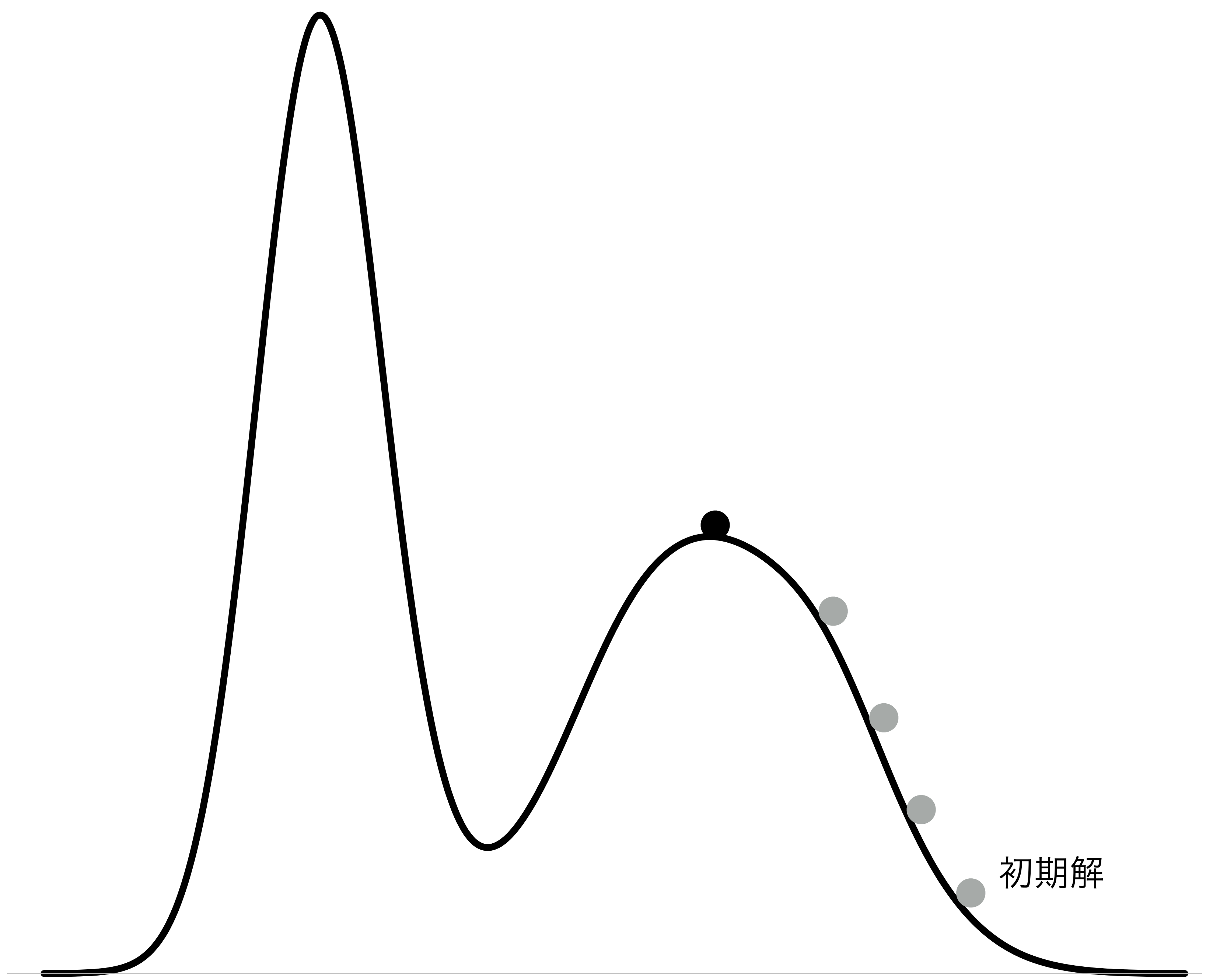
山登り法のイメージ図



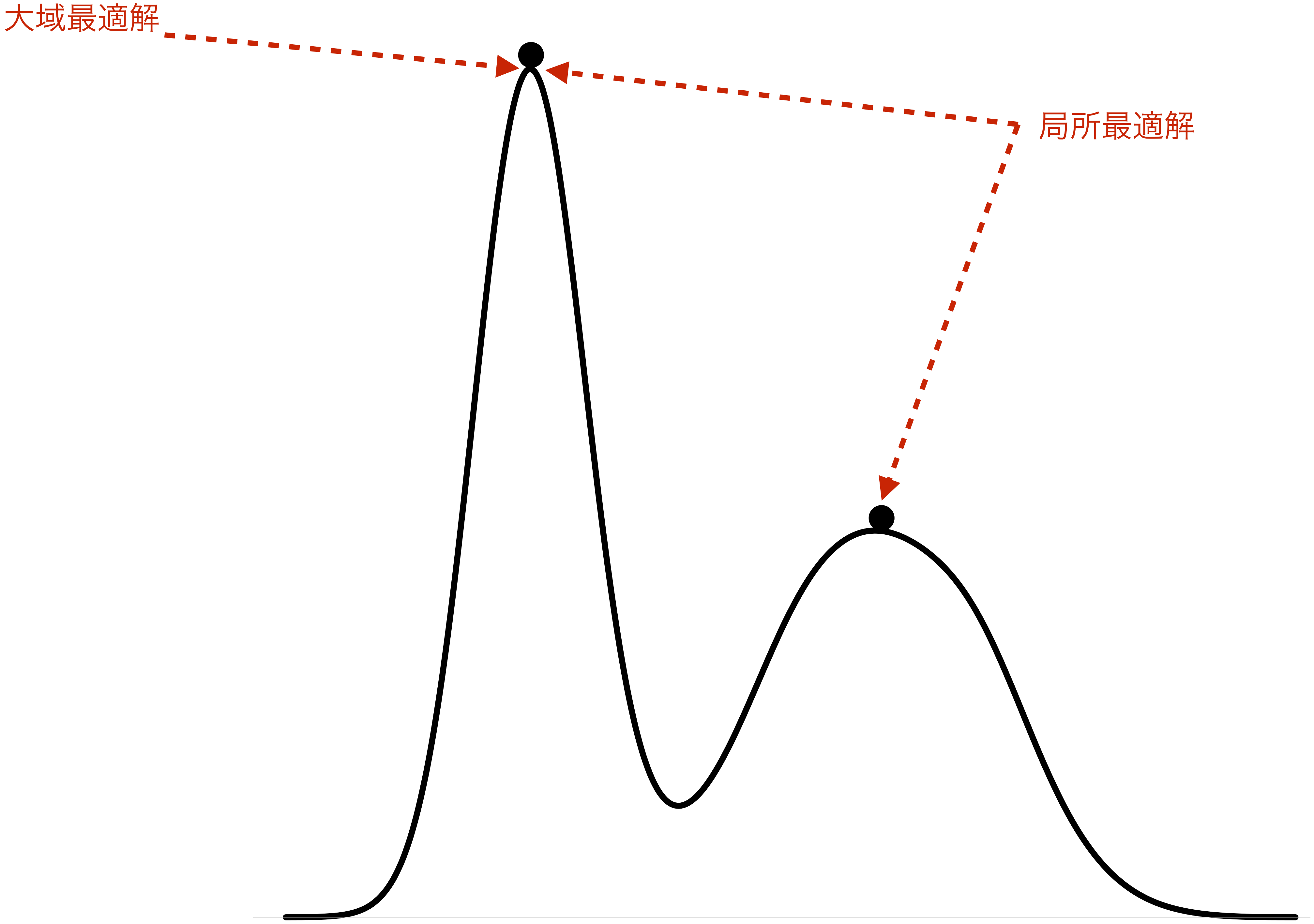
山登り法のイメージ図



山登り法のイメージ図



局所最適解と大域最適解



局所最適解と大域最適解

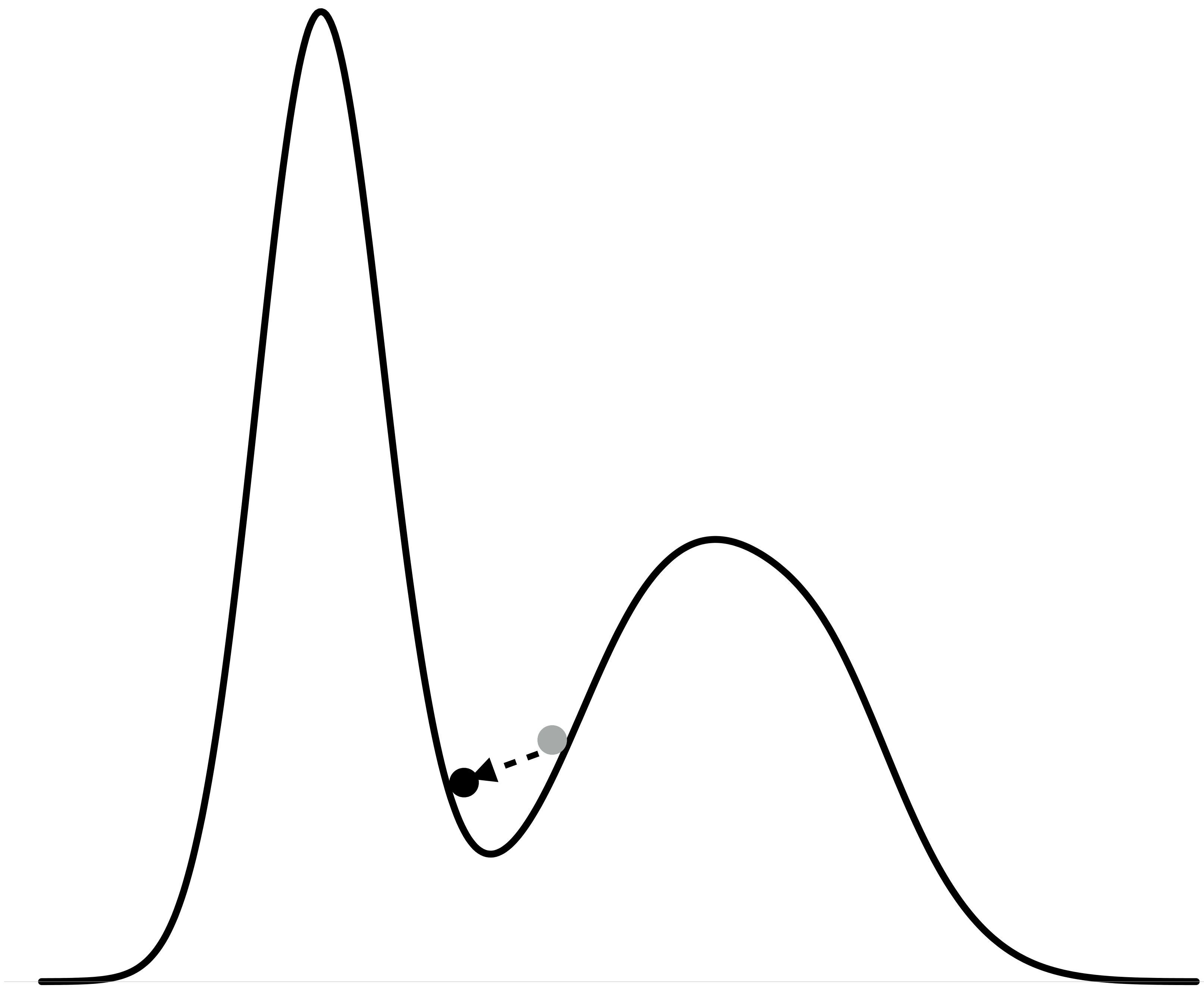
- 局所最適解とは、その解の近傍と比べると最もスコアの良い解。
大域最適解とは、あらゆる解の中で最もスコアの良い解。

- 山登り法では、局所最適解を得る事は保証されるが、局所最適解が大域最適解であるとは限らない。(一般的にはそうでないことが多い)

- 局所最適解は最初に与えた初期解に依存するため、様々な初期解を与えて山登り法を繰り返し行い、最も良かった解を採用するというのが考えられる。
(多スタート局所探索法というメタヒューリスティクスの一つ)

局所最適解を抜け出す

- 局所最適解に止まることを避けるためには、時に「スコアを下げる」遷移を手続き中に含める必要がある。



焼きなまし法

- 現在の解よりも低いスコアの解があったとしても、一定確率以上で遷移する事で、より最適な解を発見する。

1. まず最初にランダムに初期解xを生成し、そのスコアf(x)をbest scoreとする。温度Tと冷却率r (0 < r < 1)を設定する。
2. xの近傍の中からrandomにx'を選ぶ。
3. f(x')がbest scoreよりも良ければbest scoreをf(x')に更新する。
4. f(x')がf(x)より良ければ、x'をxに代入する。
   f(x')がf(x)より悪い時は、exp((f(x') - f(x))/T)の確率でx'をxに代入する。
5. 温度を更新する。(T=rT)
6. 2-5を繰り返す。
7. 一定回数繰り返すか、または温度が閾値を下回ったら繰り返しを終了し、best scoreを出力する。

焼きなまし法

- 温度が高い時(探索の初期段階)は、高い確率で低いスコアへの遷移が起こる。温度が低くなるにつれ(探索の後期段階)、山登り法に近づいていく。
- 実際には、rとTを適切に設定するのは容易ではない。
(ハイパーパラメータのチューニング)
しかし適切な設定をすると、単純に山登り法を繰り返すよりも、同一時間で良い解が得られることが多い。

- 機械学習において、解析的に解けない分布から代表点をサンプルする手法として、メトロポリス法と呼ばれる手法があるが、考え方としては類似している。

タブー探索

- 最近訪れたxをタブーリストに入れておき、タブーリストには遷移しないようにする方法。

1. 最初にランダムに初期解xを生成し、そのスコアf(x)をbest scoreとする。
2. xの近傍x'の中で、タブーリストに含まれていないx'のうち、最もスコアの良いx'に注目する。f(x')がbest scoreよりもbest scoreを更新する。
3. xをタブーリストに含める。タブーリストのうち、最も古い解を取り除く。x’に(現在の状態より悪くなっても)遷移する。
4. 2-3を一定回数繰り返したのち、best scoreを得る解を出力する。

遺伝的アルゴリズム

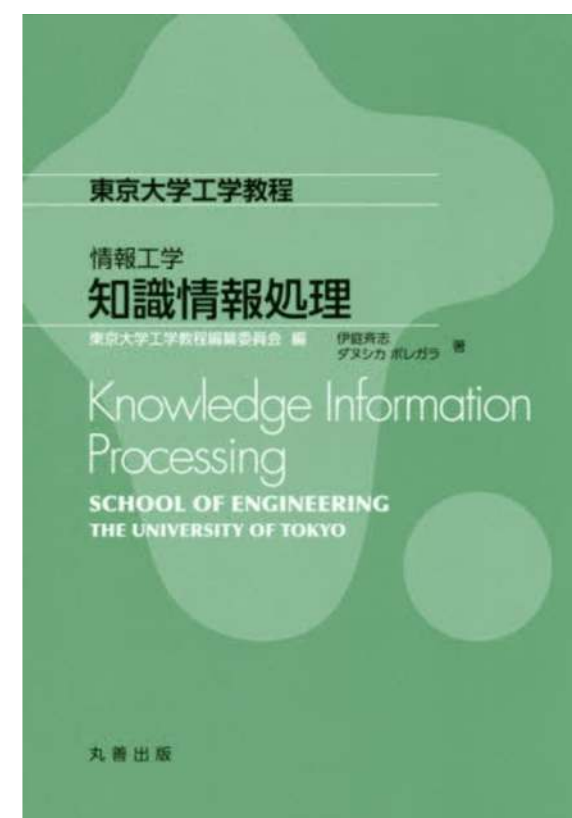
- 生物の進化を模倣する事で解を探索するアルゴリズム。

1. ランダムに初期世代の解の集団を生成する。集団にはN個の解が含まれるとする。
2. 集団内の全ての解に対して、f(x)を計算する。
3. この時集団内から解を選択し、次のいずれかの操作(GAオペレータ)を行い、
得られた解を次世代の集団に追加する。次世代の集団個数がN個になるまで、
操作3を繰り返す。また、f(x)の高い個体を選択する確率が高まるようにする。
(1) 変更を加えない。
(2) 解の一部に変更を加える。
(3) 2つの解を選択し、2つの解を混合させて1つの解を作る。
4. 定められた回数分、2-3を繰り返す。

ノーフリーランチ定理

- 様々なメタヒューリスティクスが提案されているが、どのアルゴリズムを使うと平均的に最も精度が良いのだろうか？

- ノーフリーランチ定理として、「どんな問題に対しても平均的に効率よく探索出来るアルゴリズムは存在しない」ということが証明されている。



近似アルゴリズム

- メタヒューリスティクスは実用上高い性能が得られるが、
  どれくらい最適解に近いのかという理論的な保証はない。

- 最大値を求める問題において、少なくとも最適解の1/kの解が得られる
  ことが理論的に保証されるアルゴリズムを、k-近似アルゴリズムと呼ぶ

固定パラメータアルゴリズム

- NP困難な問題は、全ての可能な入力ケースを考えると現実時間で厳密には解く事が難しいが、現実の問題では厳密に解ける事がある。
- なので、現実的な入力に対して高速に解きうるアルゴリズムの研究が最近盛んである。

- 固定パラメータアルゴリズムとは、入力データに依存するあるパラメータをk (例:グラフにおける木幅) とした時、O(*f(k)poly(n)*)で解くアルゴリズムのことである。
- ここで、f(k)はkに対して指数時間であり、poly(n)はnに対して多項時間であるとする。kが現実データにおいては十分に小さいならば、現実時間で厳密に解きうる。

分枝限定法

- 問題を部分問題に分割し(分枝)、各部分問題の解の上界と下界を求め、その下界が別の部分問題の上界より大きければ探索を打ち切る(限定)ことで計算を高速化する。

- ナップサック問題を例にとる

ナップザック問題(再掲)

分枝限定法

$\boxed{図18.5}$

- 分枝限定法は、計算量を変化させるわけではないが、実用上高速になることが多い。また、近似解ではなく正しい解を得られる。

まとめ

- 多項式時間で解ける判別問題のクラスをP、答えがYesである時に証拠が与えられればYesであることを多項式時間で検証できる判別問題のクラスをNPと呼ぶ。PとNPが等しいか否かは未解明問題である。

- NPの中で最も難しい問題をのクラスをNP完全、少なくともNP完全より同等以上に難しい問題をNP困難と呼ぶ。

- NP困難問題を解く事は難しいが、入力の性質が良い場合は
固定パラメータアルゴリズムや分枝限定法で実用時間内に
厳密解が得られる。

- また、メタヒューリスティクスや近似アルゴリズムを利用すれば、
近似的な最適解が得られる。