メタヒューリスティクスの数理

- 2.7　：大近傍探索法
- 2.8　：探索空間平滑化法・交互平滑化法
- 2.9　：部品最適化法
- 2.10 ：多レベル法

M2　大山雄己

大近傍探索法

■大近傍探索法

・近傍のサイズが大きいほど得られる解の質は向上するが，計算時間が上昇する.

・大きな近傍を何らかのテクニックを用いて，効率的に求めることが重要.

・この方法を，**大近傍探索法**という.

1. 動的計画法
2. 容易に解ける特殊形を用いた方法
3. **ヒューリスティクス探索**
4. 分枝限定法を利用する方法

ヒューリスティクス探索の例として，**可変深度局所探索法（VDS）**を紹介する.

大近傍探索法

■可変深度局所探索法

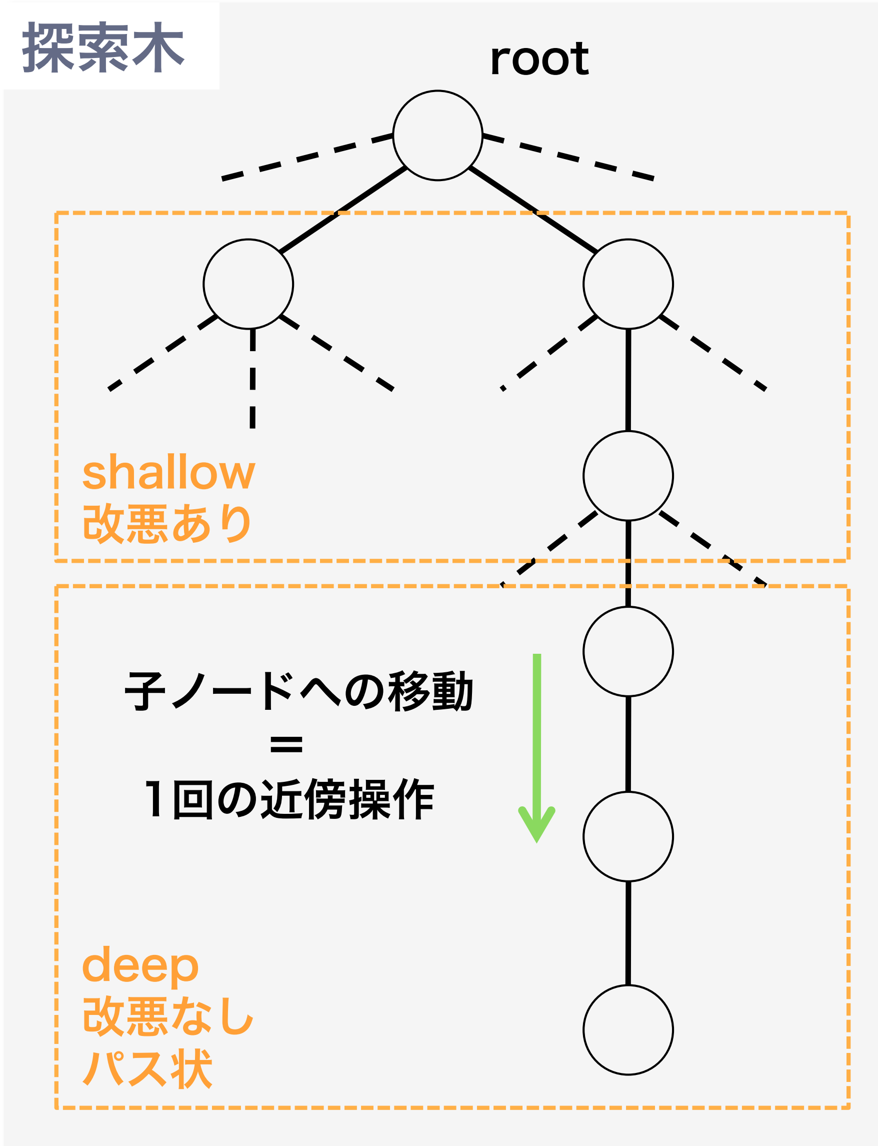
単純な近傍操作を連鎖的に複数回適用することで生成され得る新たな解集合を近傍と定義する方法.

shallow part

改悪を許しながら，近傍操作により得られる全ての解を子として持つ.

deep part

自身より評価の高いノードを即時戦略にて探索し，唯一の子とする.



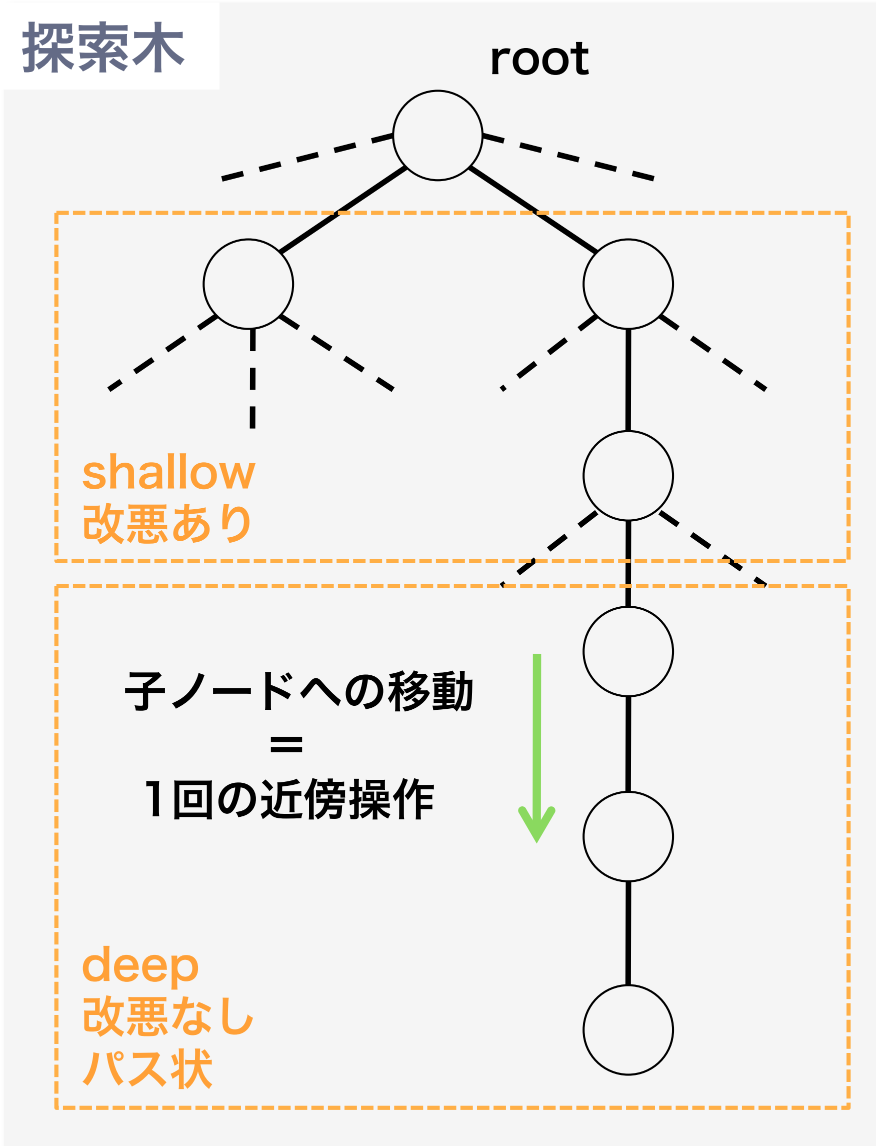
大近傍探索法

■探索木の設計

問題に応じて，木の制限方法を工夫する必要がある.

<u>制限と注意点</u>

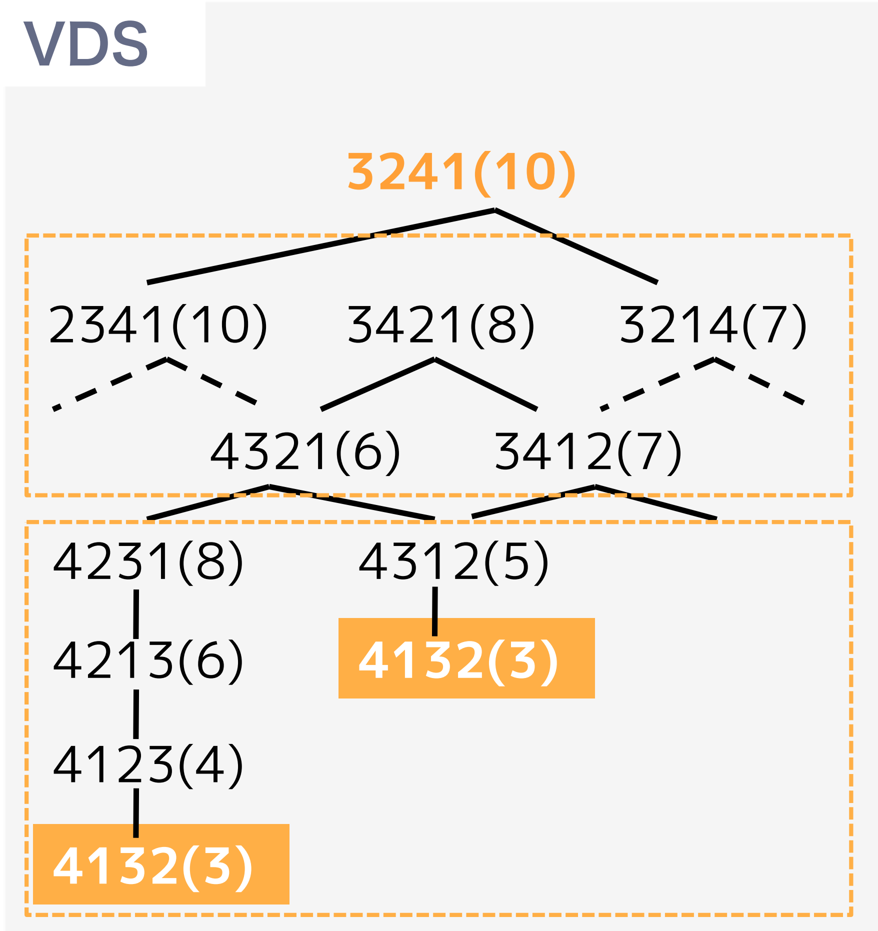
・shallow partで繰り返しを防ぐため，既に生成した解については二度目以上は生成しない.
・deep partではroot（根）への逆戻りを防ぐ工夫が必要である.
・shallow partの深さはあらかじめ決めておく必要があるため，設定には注意が必要である.



大近傍探索法

■スケジューリング問題での適用イメージ

近傍操作　＝　隣り合った1組のジョブの入れ替え



探索空間平滑化法

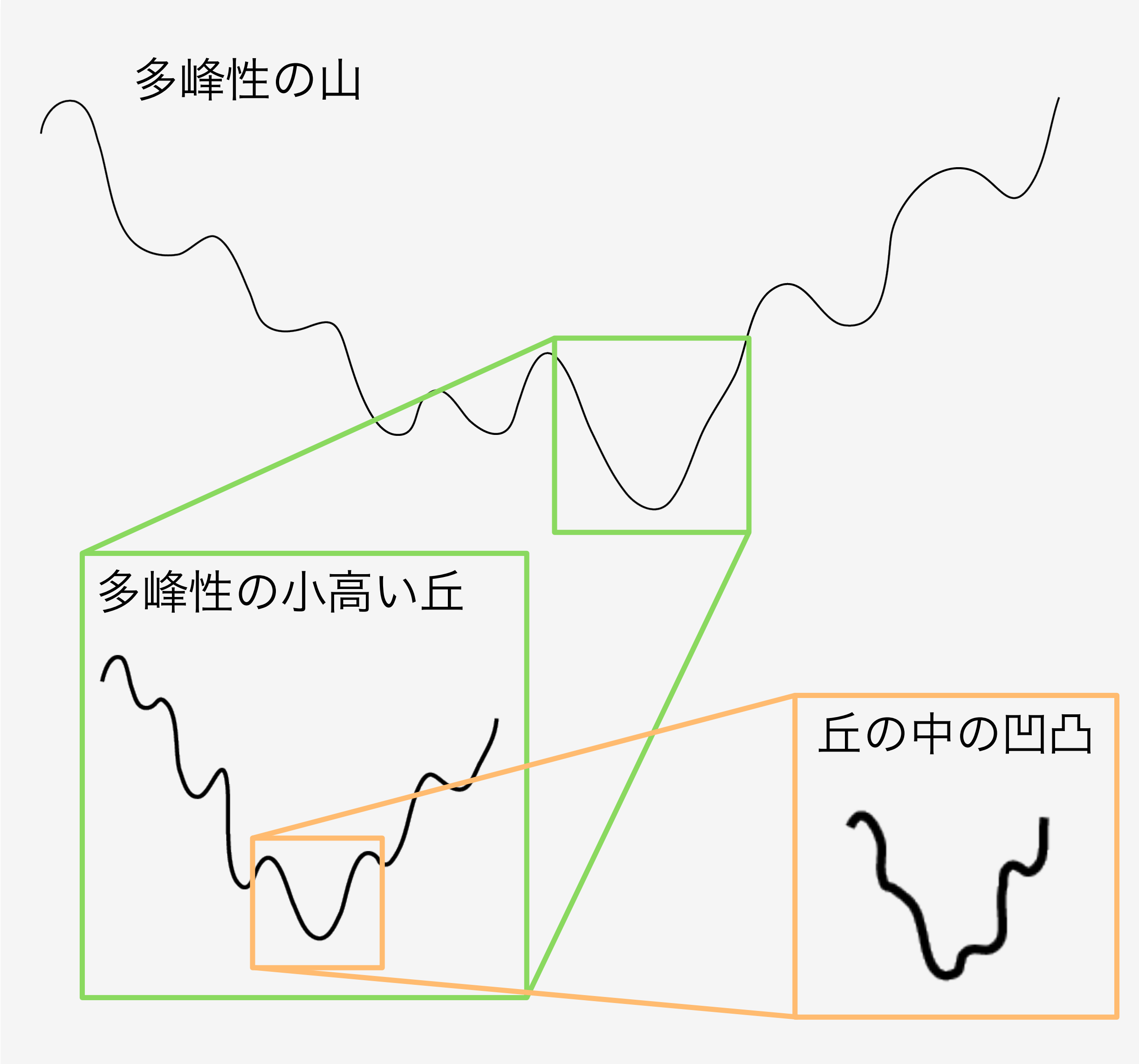
■組み合せ問題の探索空間

最適解を見つけるためには，
目的関数の小さな凹凸を消す
アイディアが必要.

・目的関数の元になる費用を
同一視してしまう

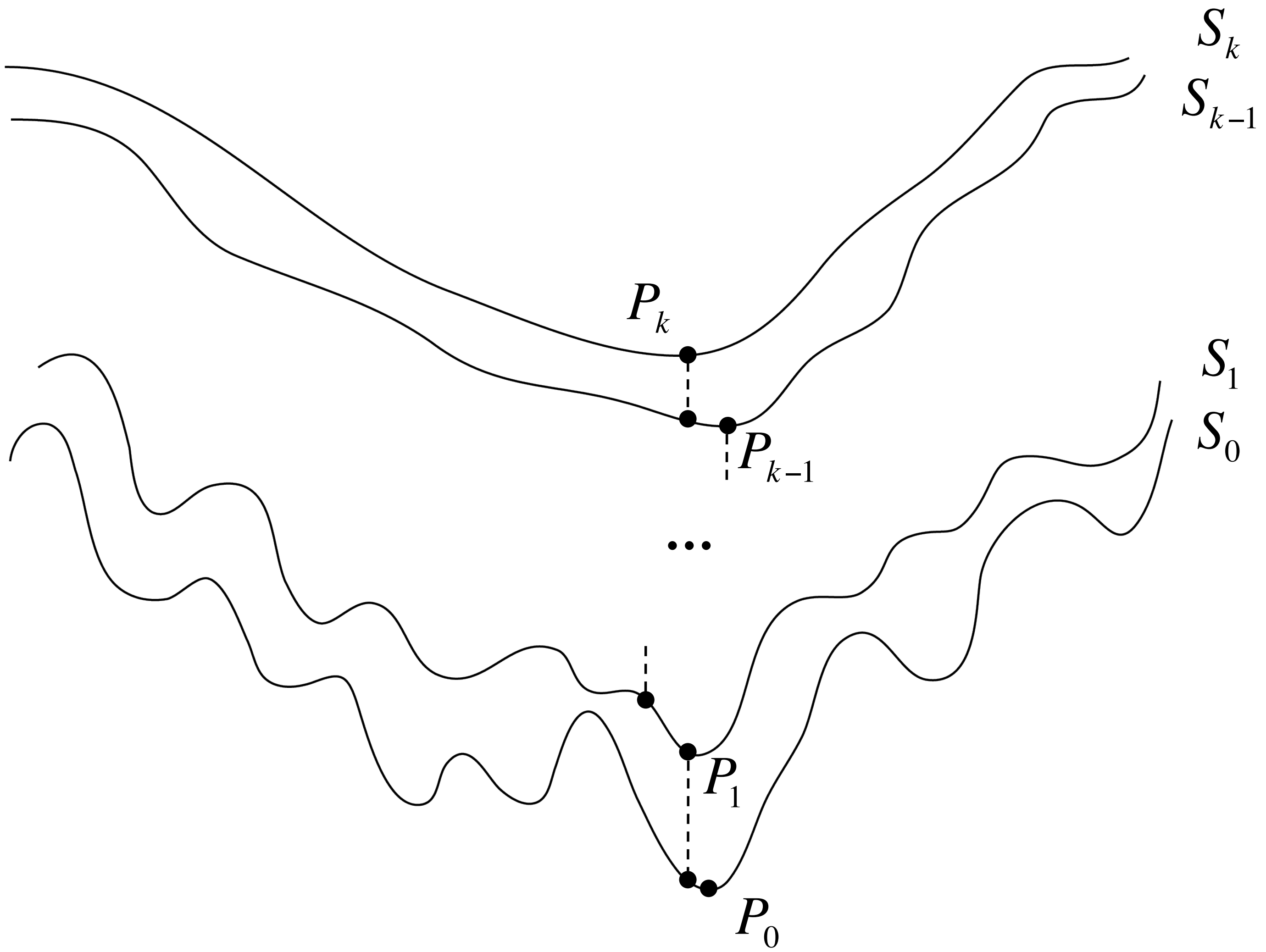
探索空間平滑化法

細かい高さの差を測定すること
ができない高度計をもとに
山を登ることを考える.



探索空間平滑化法

■探索空間平滑化法の概念



- 局所最適点が十分に少なくなるまで平滑化（k 段階）

- 最も平滑化された空間 k から近似解を求めていく.

- 上の段階の近似解を探索の開始点として，最良解を求める.

- 原問題の最良解 P_0 が求まるまでプロセスを繰り返す.

探索空間平滑化法

■Gu-Huangの方法

$$D_{ij}(\alpha) = \begin{cases} \bar{D} + |D_{ij} - \bar{D}|^{\alpha} & D_{ij} \geq \bar{D} \\ \bar{D} - |D_{ij} - \bar{D}|^{\alpha} & D_{ij} < \bar{D} \end{cases}$$

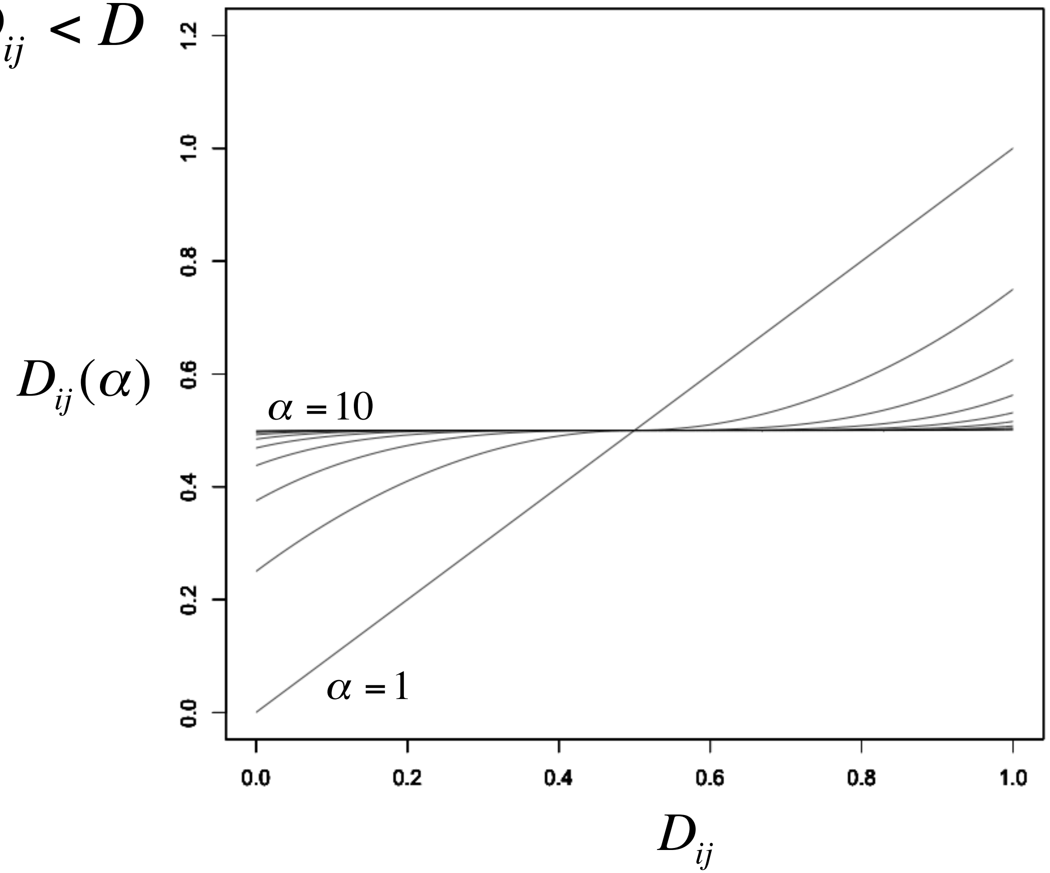
D_{ij} ：2地点 (i,j) のもともとの距離
（0〜1の値にスケーリング）

$\bar{D}$ ：2地点 (i,j) の距離の平均値

$\alpha = 1$ のとき， $D_{ij}(\alpha) = D_{ij}$

α が大きいと， $D_{ij}(\alpha) = \bar{D}$

つまり， $\alpha = n > 1$ （各自設定）から始めて， $\alpha = 1$ まで近傍探索を繰り返す.



探索空間平滑化法

■探索空間平滑化法のアルゴリズム

探索空間平滑化法

1 初期解 s^* を生成し，対応する目的関数の値を C^* とする．また,$\alpha = \alpha_0$ とする．$(1 < \alpha_0)$

2 式(1)を用いて費用 D_{ij} を$D_{ij}(\alpha)$に置き換えた平滑化問題を定義する．

3 s^* を初期点として，単純近傍探索法を用いて平滑化問題を解く．候補解 s' を生成し，$C' \le C^*$ ならば置き換え．

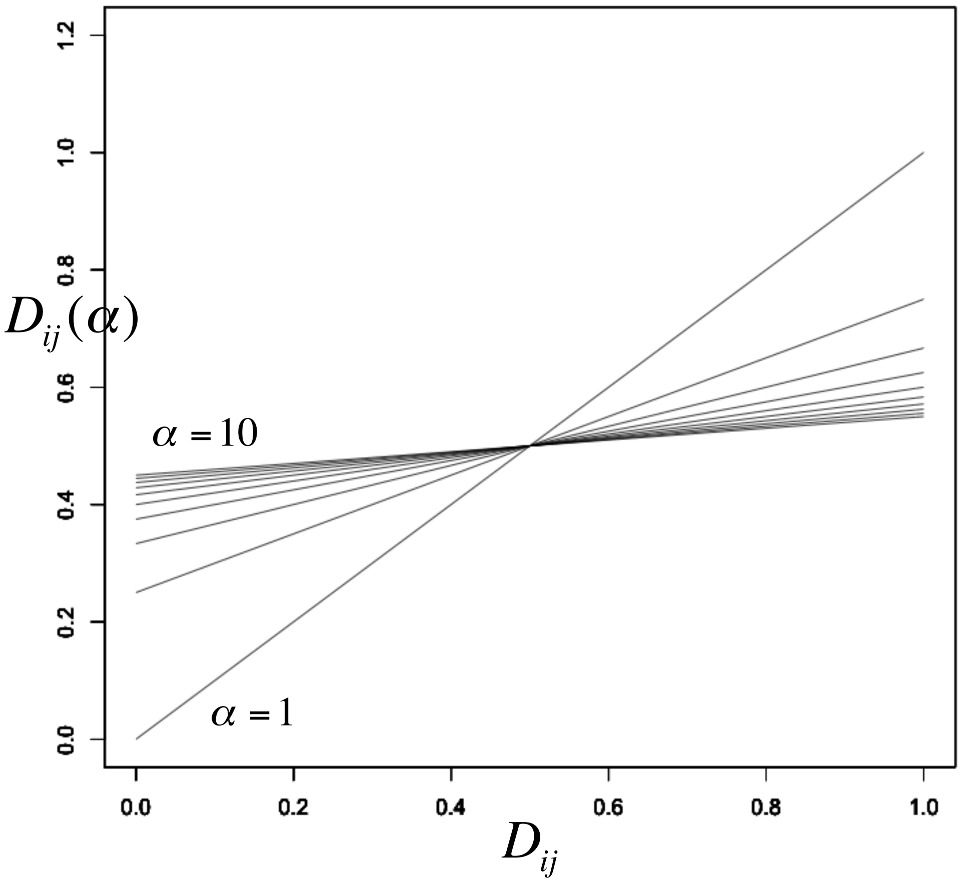
4 $\alpha = \alpha - \Delta\alpha$ とし,$\alpha < 1$ならば終了．$\alpha \ge 1$ ならばステップ3へ戻る．

$$D_{ij}(\alpha) = \begin{cases} \overline{D} + |D_{ij} - \overline{D}|^{\alpha} & D_{ij} \ge \overline{D} \\ \overline{D} - |D_{ij} - \overline{D}|^{\alpha} & D_{ij} < \overline{D} \end{cases} \qquad (1)$$

探索空間平滑化法

■「平滑化」が良かったのか？

$$D_{ij}(\alpha) = \overline{D} + \frac{1}{\alpha}(D_{ij} - \overline{D})$$



この場合，元の局所探索法よりもいい結果が得られなかった.

$$D_{ij}(\alpha) = \begin{cases} \overline{D} + |D_{ij} - \overline{D}|^{\alpha} & D_{ij} \geq \overline{D} \\ \overline{D} - |D_{ij} - \overline{D}|^{\alpha} & D_{ij} < \overline{D} \end{cases} \qquad (1)$$

もう一つの特徴として，平均より大きいところで凸関数，小さいところで凹関数というものが重要.

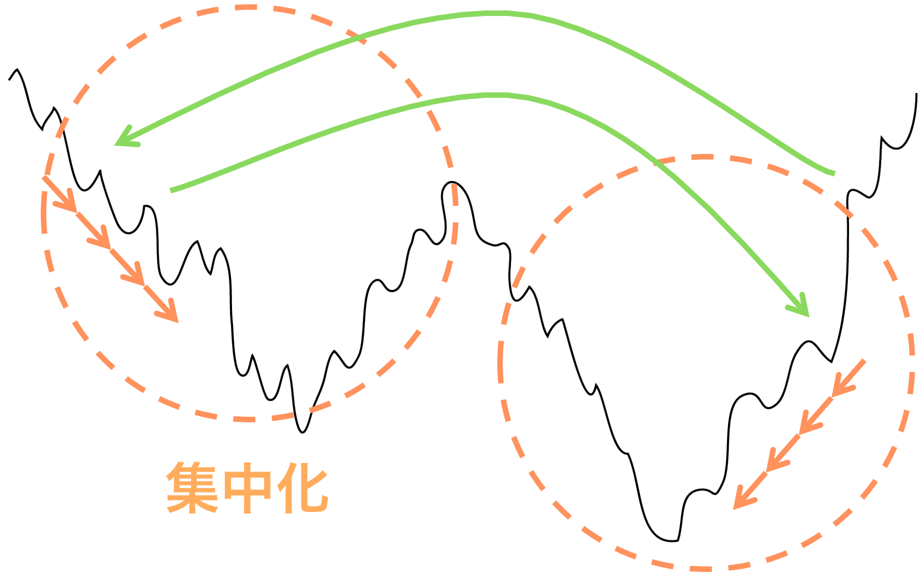
交互平滑化法

■探索の集中化/多様化

1．探索の集中化（Intensification）

「良い解同士は似通った構造を持つ」という仮定（POP）に従い，構造の似ている解を集中的に探索する方法.

- 同じ解を何度も探索
- 解の存在しない周辺を探索



2．探索の多様化（Diversification）

異なる構造を持つ解を探索する方法.

2つのバランスを取りつつ探索することが高速な局所探索設計にとって重要

平滑化の中で探索の集中化と多様化を交互に繰り返す　→　交互平滑化法

交互平滑化法

■探索の集中化/多様化

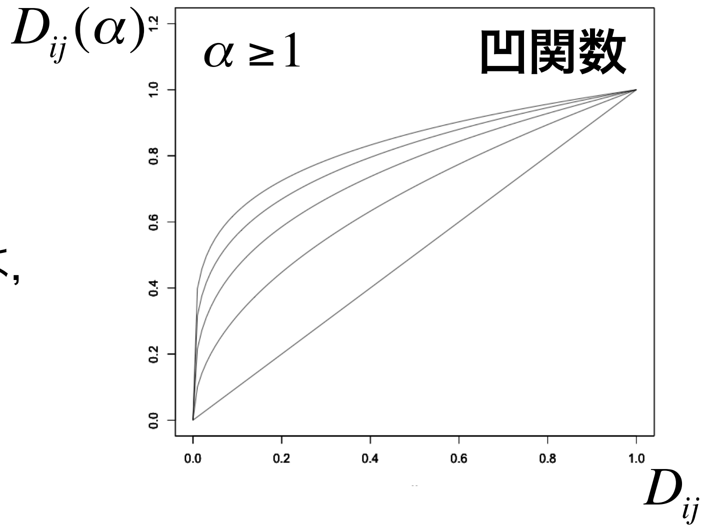
1. 探索の集中化（Intensification）

$D_{ij}(\alpha) = D_{ij}^{1/\alpha}$　　$\alpha = 2$ として,

長さ D の枝を除き, $\hat{D}$ の枝を加えた時の変化量は,

$$\sqrt{D} - \sqrt{\hat{D}} = \frac{D - \hat{D}}{\sqrt{D} + \sqrt{\hat{D}}}$$

構造の近いところで, より大きな改善を行なっている.

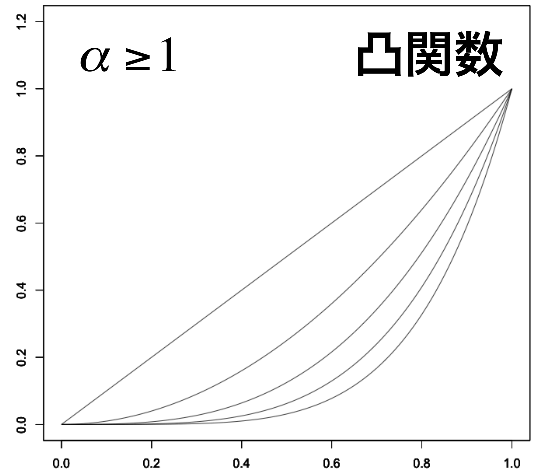


2. 探索の多様化（Diversification）

$D_{ij}(\alpha) = D_{ij}^{\alpha}$　　$\alpha = 2$ のときの変化量は,

$$D^2 - \hat{D}^2 = (D + \hat{D})(D - \hat{D})$$

構造が遠いところで, より大きな改善を行なっている.



交互平滑化法

■交互平滑化法のアルゴリズム

交互平滑化法

1　初期解 s^* を生成し，対応する目的関数の値を C^* とする．また，$\alpha = \alpha_0$とする．$(1 < \alpha_0)$

2　式(1)を用いて費用 D_{ij} を $D_{ij}(\alpha)$ に置き換えた平滑化問題を定義する．

凹関数，探索の集中化

3　s^* を初期点として，単純近傍探索法を用いて平滑化問題を解く．候補解 s' を生成し，$C' \leq C^*$ ならば置き換え．

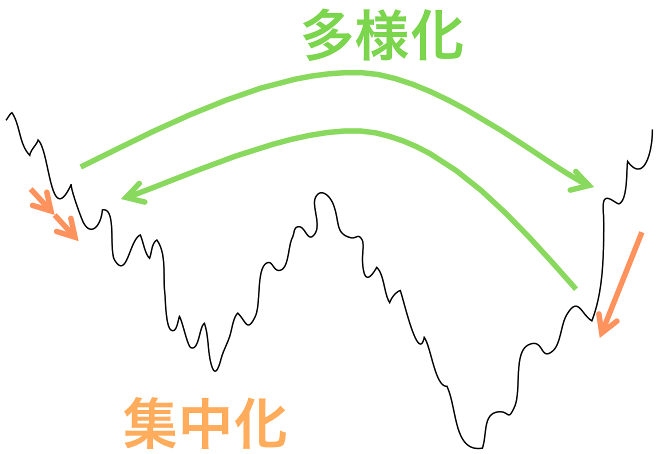
4　式(2)を用いて費用 D_{ij} を $D_{ij}(\alpha)$ に置き換えた平滑化問題を定義する．→3と同様のステップ

凸関数，探索の多様化

5　$\alpha = \alpha - \Delta\alpha$ とし，$\alpha < 1$ならば終了．$\alpha \geq 1$ ならばステップ2へ戻る．

$$D_{ij}(\alpha) = D_{ij}^{1/\alpha} \quad (1)$$

$$D_{ij}(\alpha) = D_{ij}^{\alpha} \quad (2)$$



部品最適化法

■部品

対象とする問題の解が，いくつかの「部品」に分割することができると仮定する．

解 x について，「部品（part）」を以下の部分集合として定義する．

$$p_1 \cup p_2 \cup \cdots \cup p_m = x \quad \text{かつ} \quad p_i \cap p_j = 0 \quad i \neq j$$

仮定①

$$\boxed{p_i \cup \cdots \cup p_k} \rightarrow \textbf{部分問題（子問題）}$$

の目的関数に対する最適化問題には厳密解（もしくは効率的な近似解）と解法が存在する．

仮定②

部品同士に，近さを表す何らかの尺度がある．

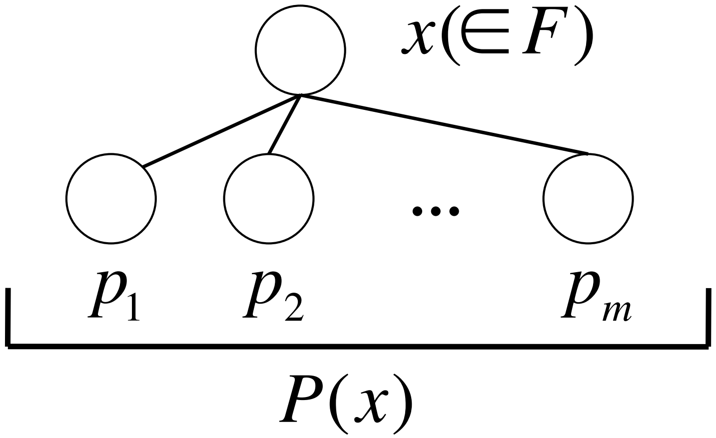
部分問題（子問題）を最適化し，部品を置き換えて徐々に大きな部分問題を解いていく．

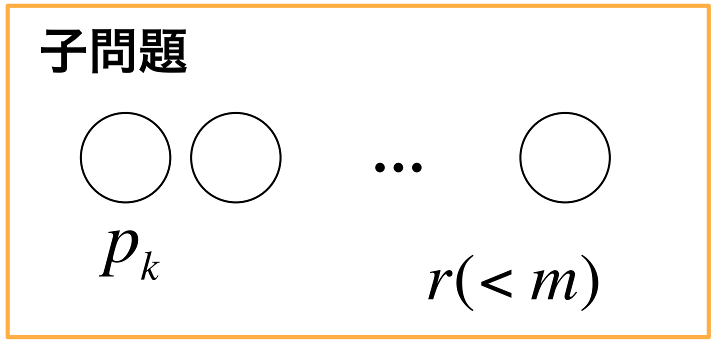
部品最適化法

■部品最適化法のアルゴリズム

部品最適化法

1. 解 x を m 個の部品に分解したものを $P(x)$ とする．
2. $P(x)$ から適当な部品 p_k と構造の近い r 個の部品からなる子問題を生成する．
3. 子問題を最適化し，改良解が得られた場合は解 x の一部を改良された部品で置き換える．
4. 終了判定基準が満たされるまで，以上のステップを繰り返す．





仮定が厳しすぎるという問題がある．

多レベル法

■探索の集中化/多様化

いくつかの小高い丘をまとめて1つの丘とみなしてしまう方法.

問題の縮小フェイズ

1　最適解に含まれる可能性の高い「部品」を次々と固定することで，問題を簡略化.

2　十分に縮小されたと判断されたら，簡略化された問題を解く.

問題の拡大フェイズ

3　縮小の逆操作（拡大）を行ない，徐々に部品を追加してオリジナルの問題に近づける.
4　拡大された問題に対して順次適当な改善法を適用し，問題を解く.

多レベル法

■グラフ分割問題