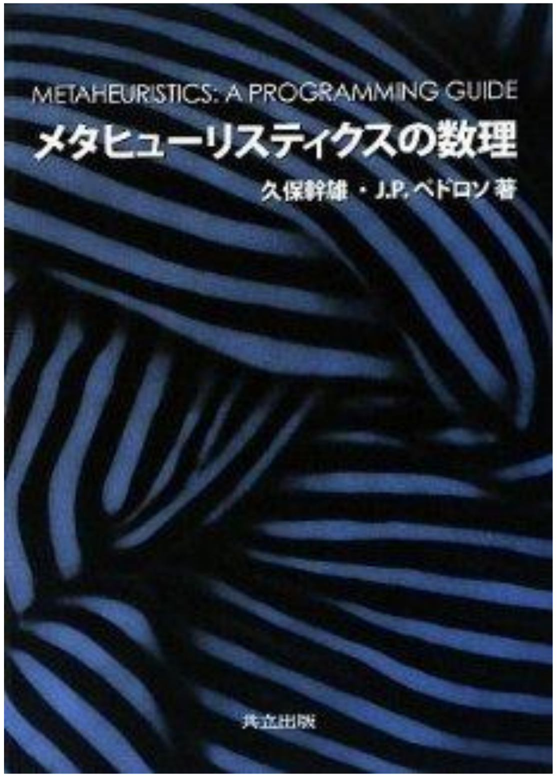
メタヒューリスティクスの数理 (Metaheuristics: A programming guide)
久保幹雄, J. P. ペドロソ 著
共立出版, 2012.



３章
数理計画とメタヒューリスティクスの融合
の前半(3.1～3.4)

2013年7月10日
M2　伊藤 創太

3章の内容

最適化問題を解きたい！

ソルバーを使って求解

× 問題が大きいと解けない

or

アルゴリズムを
1から設計

× いちいち全部設計するのは手間

→ 両方使う！ ←

ソルバーの計算
- 3.1 分枝限定法
- 3.2 使い分けと融合の必要性

実行可能解を構築
- 3.3 変数固定法
- 3.4 不完全分枝限定法と飛び込み法
- 3.5 緩和固定法
- 3.6 容量スケジューリング法

実行可能解を改善
- 3.7 MIP近傍局所探索法
- 3.8 局所分枝法

構築＋改善
- 3.9 MIP併合法

(branch and bound method) 分枝限定法

・最適化問題の厳密解を求める方法

・(ここでは)混合整数計画問題の解法

混合整数計画問題(MIP問題, mixed integer programming)

$$\min \ \underline{fx} + \underline{gy} \quad \text{目的関数}$$

$$\underline{Ax} + \underline{By} \le b \quad \text{制約条件}$$

$$\underbrace{x \in \mathbf{R}^n}_{\text{整数}}, \quad \underbrace{y \in \mathbf{Z}^p}_{\text{実数}}$$

目的関数は線形関数

制約条件は線形関数

整数制約がなければ解くのは簡単

例：ナップザック問題

$$\max 10x_1 + 7x_2 + 25x_3$$

$$2x_1 + x_2 + 6x_3 \le 7$$

$$x_j \in \{0,1\} \ (j = 1,2,3)$$

意味：7ドル持ってお買い物

1
10個入り
$2

2
7個入り
$1

3
25個入り
$6

最適化問題の表し方

・実行可能領域 F と目的関数 f の組み合わせで書く

最適化問題 $P = (F, f)$　　　最適解の目的関数値 $Z(P)$

・問題を小問題に分割する

実行可能領域 F を、排反な領域 $F_1, F_2, \ldots, F_n$ に分ける
分割 とよぶ

例：ナップザック問題

商品1,2,3の組み合わせ最適問題
→商品1を買うときの商品2,3の組み合わせ最適問題＋
　商品1を買わない時の商品2,3の組み合わせ最適問題　　に分割

$$F = \{x_1, x_2, x_3\} \quad x_1, x_2, x_3 \in \{0,1\} \quad \rightarrow \quad \begin{array}{ll} F_1 = \{1, x_2, x_3\} & x_2, x_3 \in \{0,1\} \\ F_2 = \{0, x_2, x_3\} & x_2, x_3 \in \{0,1\} \end{array}$$

分解の基本原理

分解した実行可能領域 F_i に対応する問題を P_i とする

$P_1, P_2, \ldots, P_n$ を P の分割とすると、

$$Z(P) = \min_{i=1,\ldots,n} Z(P_i)$$

例：ナップザック問題

$$\max 10x_1 + 7x_2 + 25x_3$$

$$2x_1 + x_2 + 6x_3 \le 7$$

$$x_j \in \{0,1\}\ (j = 1,2,3)$$

商品1,2,3の組み合わせ最適問題の解

$\cdots\ Z(P) = 32 \qquad x = \{0,1,1\}$

商品1を買う時の解

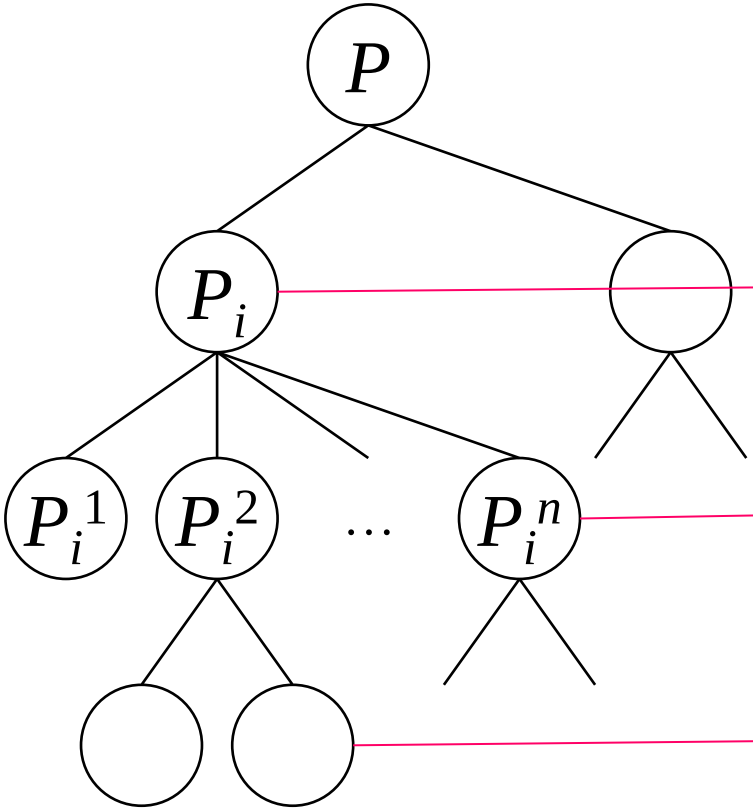
$\cdots\ Z(P_1) = 17 \qquad x = \{1,1,0\}$

商品1を買わない時の解

$\cdots\ Z(P_2) = 32 \qquad x = \{0,1,1\}$

$$Z(P) = \min(Z(P_1), Z(P_2))$$

(enumeration tree) 列挙木



問題の分割を表す

親点 (parent node)

子点 (child node)

親点の問題を分割した問題

葉 (leaf)

子点を持たない点

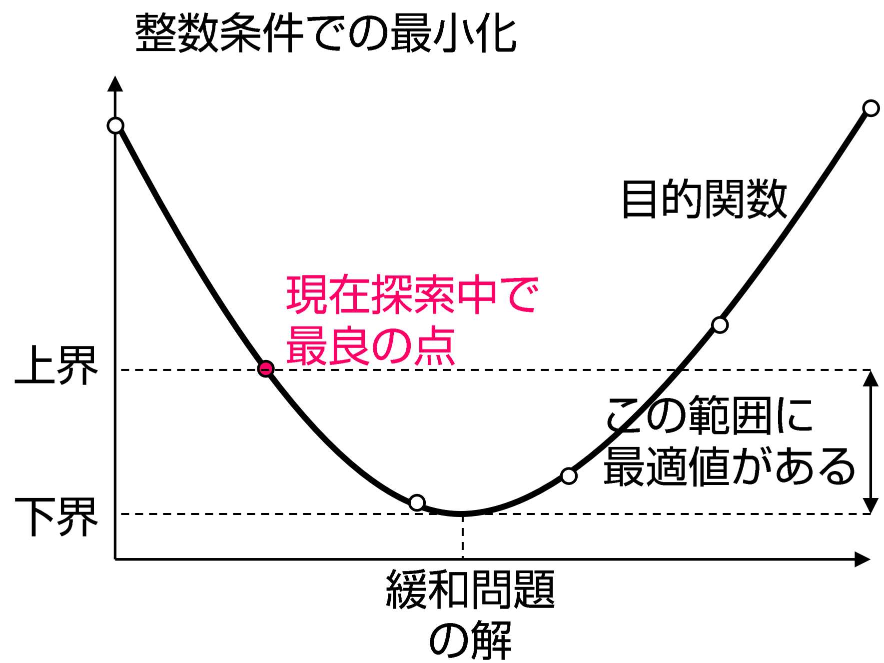
完全列挙法：全ての葉を調べる

分枝限定法：探索中に不必要な枝を効率的に除く

(bound) 限界値

最小化問題を解くとき

- **上界 *UB***　探索中の実行可能解の中で最良のもの
- **求めたい最適な目的関数値**
- **下界 *LB***　問題の整数条件を緩和したときの解



例：ナップザック問題の整数制約を緩和する（ここでは最大化問題）

$$\max 10x_1 + 7x_2 + 25x_3$$
$$2x_1 + x_2 + 6x_3 \le 7$$
$$x_j \in \mathbf{R}\ (j = 1,2,3)$$

P に対する下界

$LB(P) = 33.67$　$x = \{1.00, 1.00, 0.67\}$のとき

（P の整数計画問題の最適解）

$Z(P) = 17$　$x = \{1, 1, 0\}$のとき

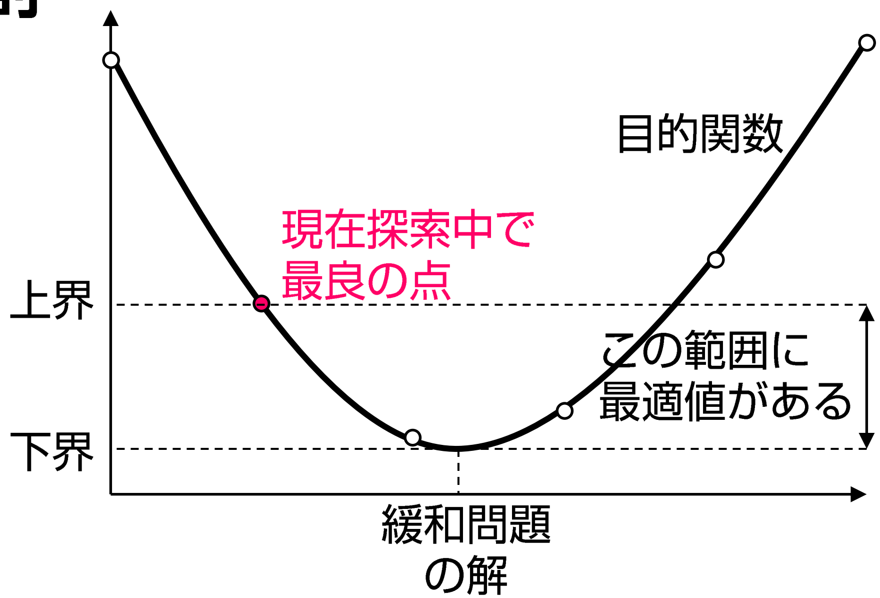
(bound) 限界値

緩和問題解による問題の分割

緩和問題の解を $\bar{y}_i$ とすると、

$$F_1 = F \cap \{y \in R^p \mid y_i \leq \bar{y}_i\}$$

$$F_2 = F \cap \{y \in R^p \mid y_i \geq \bar{y}_i\}$$



例：ナップザック問題の分割

$$\max 10x_1 + 7x_2 + 25x_3$$

$$2x_1 + x_2 + 6x_3 \leq 7$$

$$x_j \in \mathbf{R}\ (j = 1,2,3)$$

P の緩和問題の解

$LB(P) = 33.67 \quad x = \{1.00, 1.00, 0.67\}$のとき

F_1 ：$x_3 = 0 \leqq 0.67$ としたときの問題

F_2 ：$x_3 = 1 \geqq 0.67$ としたときの問題

分枝限定法アルゴリズム

アルゴリズム

1 評価前リスト L_1 に問題 P を入れる。評価後リスト L_2 を空集合、最良の上界 UB を ∞ とする

2 評価前リスト L_1 と評価後リスト L_2 がともに空集合ならば終了
そうでなければ、3~5を繰り返す

3 評価前リスト L_1 から問題を1つ選んでその問題を L_1 から除外
緩和条件を解いて評価

例：ナップザック問題

$L_1 = \{P\}$、 $L_2 = \{\emptyset\}$
$UB = \infty$

$L_1 = \{P\}$、 $L_2 = \{\emptyset\}$

P を選んで緩和問題を解く
$L_1 = \{\emptyset\}$
$LB(P) = 33.67$
$x = \{1.00, 1.00, 0.67\}$のとき

分枝限定法アルゴリズム

アルゴリズム

4 除去条件を満たせば除去
そうでなければ評価後リスト L_2 に
加える。実行解ならば UB を更新

5 評価後リスト L_2 から1つ選択して
その問題を L_2 から除外
問題を分解して、分解された問題を
L_1 に入れる

例：ナップザック問題

$L_1 = \{\emptyset\}$、 $L_2 = \{P\}$
$UB = \infty$

解が整数解ではないので
UBはそのまま

P を分解
P_1 ： x_1=1に固定した問題
P_2 ： x_1=0に固定した問題

$L_1 = \{P_1, P_2\}$ 、 $L_2 = \{\emptyset\}$

以降繰り返し

ソルバー/メタヒューリスティクス融合

最適化問題を解きたい！

ソルバーを使って求解	or	アルゴリズムを １から設計
× 問題が大きいと解けない		× いちいち全部設計するのは手間

実際の典型的な流れ

1 モデルを記述する

2 ソルバーにかける

→ 解けたら終了

→ 解けなかったら　1）あきらめる

2）<u>強い定式化やパラメータチューニング</u>

実務的にはむずかしい

3 <u>近似解法（メタヒューリスティクス）の設計</u>

変数固定法

緩和問題解をもとに変数を固定

アルゴリズム

1 N_1, N_0 を空集合とする。固定されていない添え字集合 J を N とする

2 $J = 0$ になるまで繰り返し

2a J から部分集合 J' をとる

2b 適当な方法で変数を固定。y_j を固定した変数として制約条件とする

2c 制約を付加した問題を解く

2d 実行可能であれば、J から J' を削除する

例：ナップザック問題の例

P の緩和問題の解

$LB(P) = 33.67 \quad x = \{1.00, 1.00, 0.67\}$

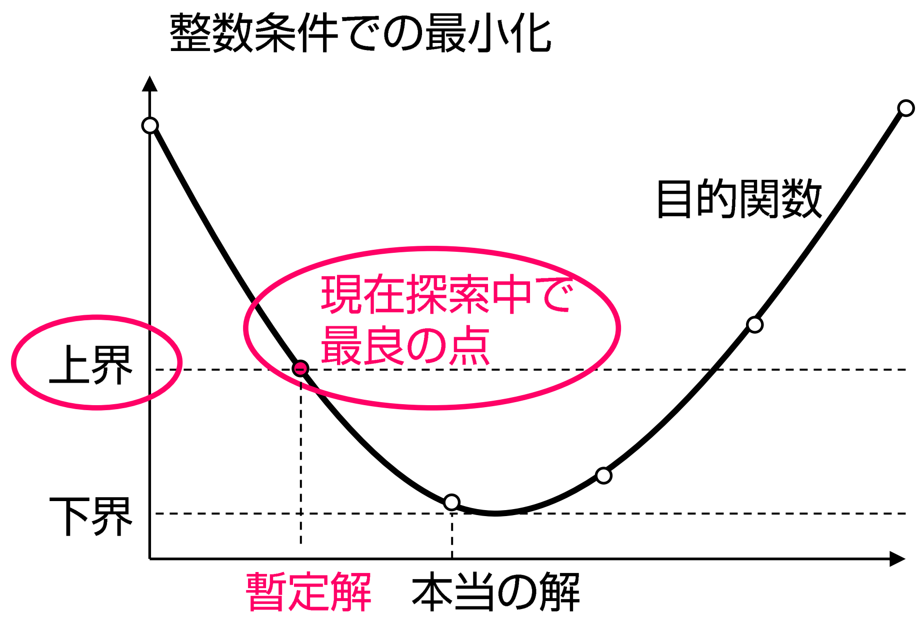
Jから3を選んだとき

$x_3 = 1$ と固定して解く

(truncated branch and bound method)打ち切り分枝限定法

分枝限定法の計算を、
決めた時間で打ち切る

上界と暫定解が求まる



早く実行可能解を見つける工夫が望ましい

→**飛び込み法**・・・緩和解で整数値に近い点から優先して固定

例：ナップザック問題の例

P の緩和問題の解

$LB(P) = 33.67 \quad x = \{1.00, 1.00, 0.67\}$

x_1 または x_2 を優先して、
$x_1 = 1$, $x_2 = 1$ と固定して解く