多次元0-1ナップサック問題に対する近似解法の研究

東京理科大学工学部経営工学科

沼田研究室

4499007　石井 宏明

4499011　伊藤 祐介

発表構成

1. はじめに
2. 定式化
3. 研究目的
4. 局所探索法
5. 局所探索法の初期解の生成
6. 実験①
7. メタ戦略
8. 実験②
9. まとめ
10. 今後の課題

1．はじめに(1)

現実に現れる最適化問題

- 意思決定
- 生産計画

↓

組合せ最適化問題として定式化できることが多い

- 巡回セールスマン問題
- <u>ナップサック問題</u>
  - 0-1ナップサック問題
- スケジューリング問題

1．はじめに(2)

問題の規模が小さい場合

全数探索が可能 → 厳密解が求まる

問題の規模が大きい場合

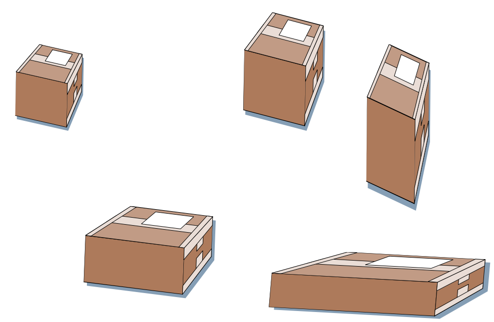
厳密解を求めるのは困難 → 近似解法が必要

⇒ 本研究では，多次元0-1ナップサック問題に対する近似解法を扱う.

0-1ナップサック問題

最大積載量のあるナップサックに荷物を詰める問題

- 荷物は複数ある
- 荷物はそれぞれ重さが違う
- 荷物はそれぞれ価値が違う



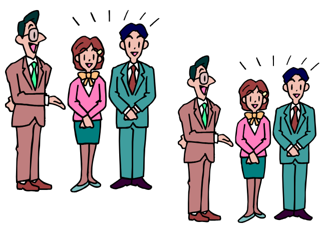
⇒ **最大積載量以内でなるべく価値を最大にする.**

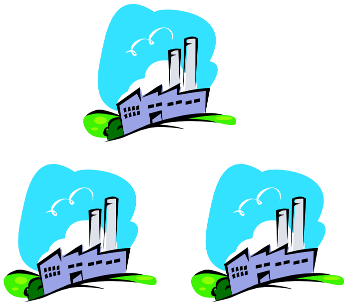
多次元0-1ナップサック問題
(MKP:Multidimentional Knapsack Problem)

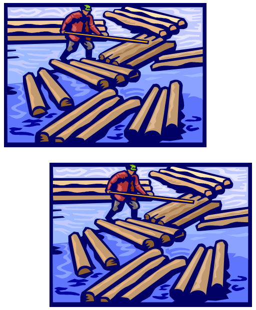
0-1ナップサック問題の制約が2つ以上ある問題

複数のプロジェクトを実施する場合

- 人の制約
- 場所の制約
- 材料の制約
- 時間の制約







全ての制約を満たしてなるべく価値を最大にする.

2．定式化(1)

定式化に使う記号

$$x_j = \begin{cases} 1, & \text{プロジェクト} j \text{を実施するとき} \\ 0, & \text{プロジェクト} j \text{を実施しないとき} \end{cases}$$

c_j ：プロジェクト j を実施することにより得られる利益($j = 1,\ldots,n$)

a_{ij} ：プロジェクト j が必要とする第 i 資源 の量($i = 1,\ldots,m$)

b_i ：第 i 資源の利用可能量

2．定式化(2)

多次元0-1ナップサック問題

$$
\begin{cases}
\text{最大化} & z = \sum_{j=1}^{n} c_j x_j & j = 1,2,\ldots,n \\
\text{制約} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & i = 1,2,\ldots,m \\
 & x_j \in \{0,1\} &
\end{cases}
$$

目的関数と制約式の係数すべての i, j に対して $c_j > 0, a_{ij} \geqq 0, b_i > 0, \sum_{j=1}^{n} a_{ij} > b_i$ となるMKPを考える.

3．研究目的

MKPに対する近似解法として**新しい局所探索法**を提案し，その性能を既存の方法と比較する．

さらに，解の改善を図るために**新しい局所探索**に**メタ戦略**を導入し，MKPに対するよりよいアルゴリズムを提案する．

4．局所探索法

解の近傍を探索し，近傍内に現在の解より良い解があれば解を改善していく方法．

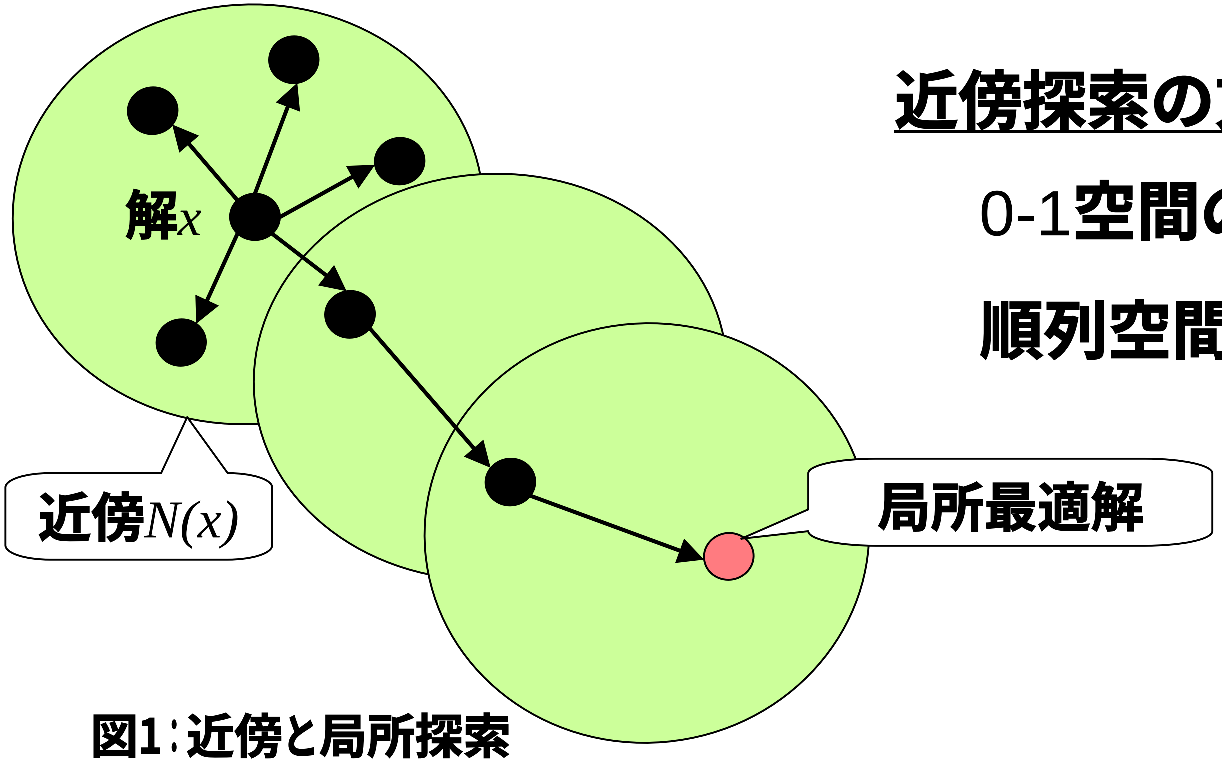


図1：近傍と局所探索

<u>近傍探索の方法</u>

0-1空間の探索（既存の方法）

順列空間の探索（新しい方法）

４．１ 0-1空間の局所探索(1)

解 x の任意の2ヶ所以下の0-1を変えて次の解を生成する方法.

0-1空間の探索における近傍

1反転近傍

解 $x = (0,1,0,0,\underline{0},1)$

$\downarrow$

解 $x' = (0,1,0,0,\underline{1},1)$

2反転近傍

解 $x = (0,1,\underline{1},0,\underline{0},1)$

$\downarrow \quad \downarrow$

解 $x' = (0,1,\underline{0},0,\underline{1},1)$

4．1 0-1空間の局所探索(2)

2^nの解空間を探索する．

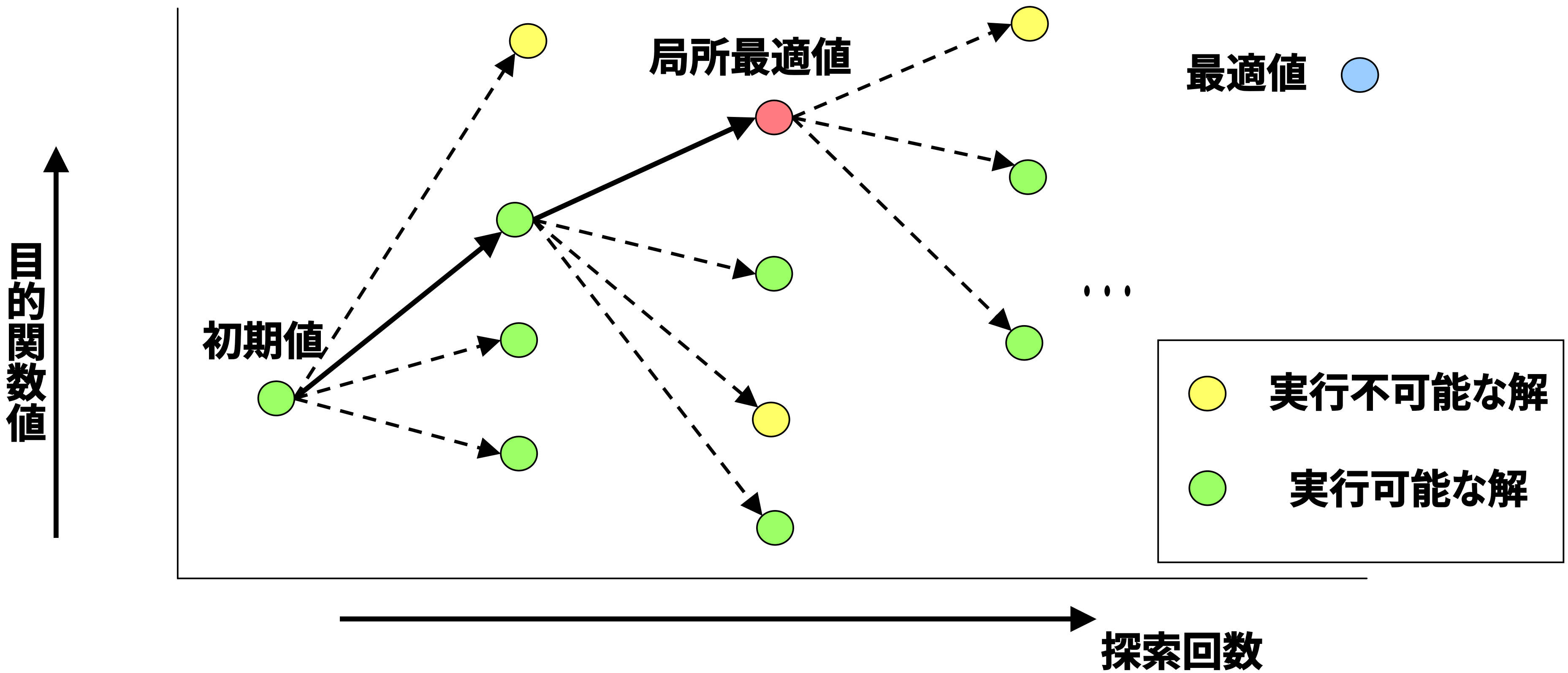


図2：0-1ベクトル空間の探索過程

4．2 順列空間の局所探索(1)

プロジェクト採用の優先順序を2ヶ所を交換して解く方法.

<u>順列空間の探索における近傍</u>

x_j プロジェクト番号

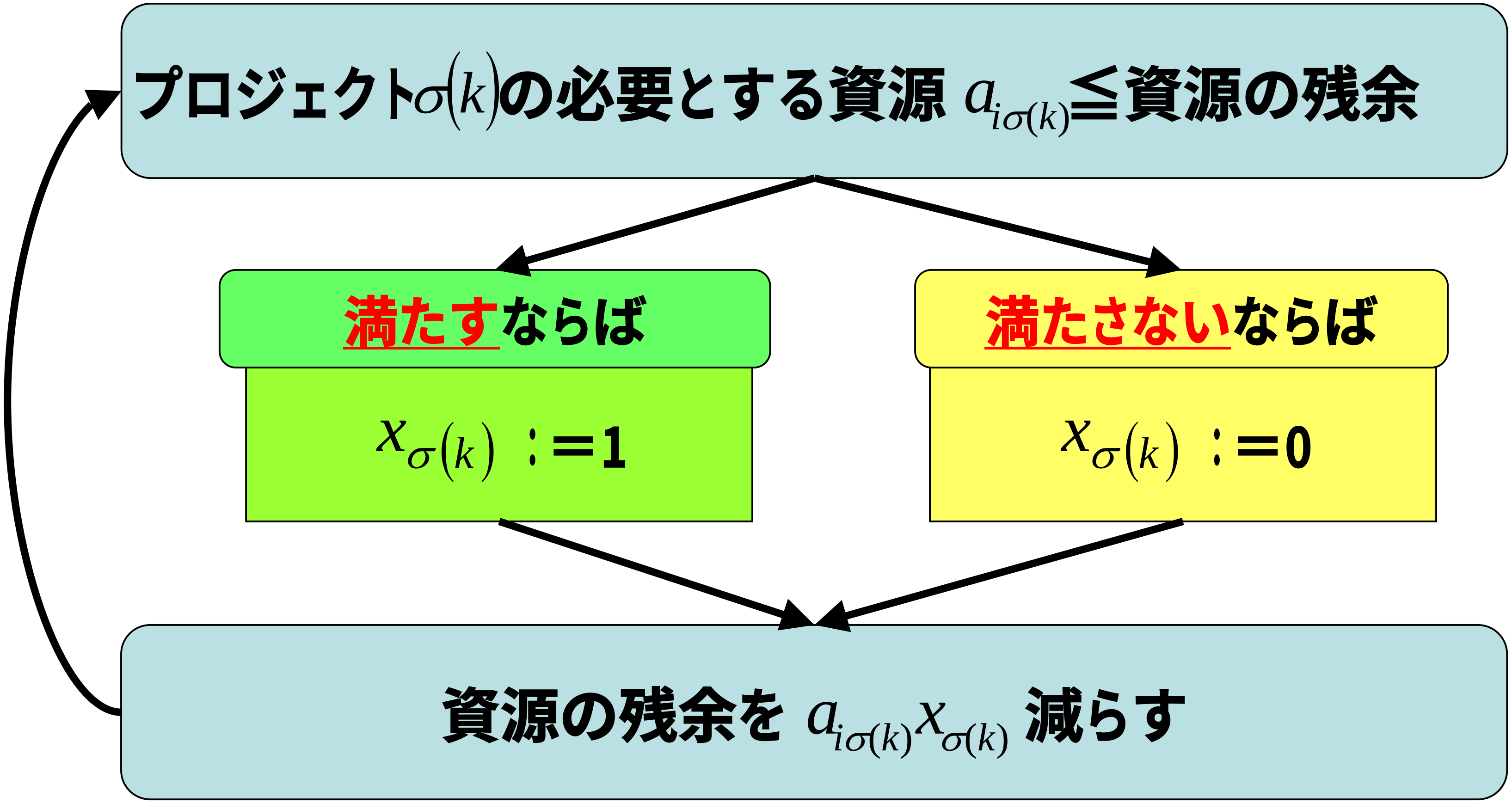
優先順序 $\sigma = (4,\underline{2},6,\underline{5},1,3)$

↓

優先順序 $\sigma' = (4,\underline{5},6,\underline{2},1,3)$

4．2 順列空間の局所探索(2)

プロジェクトの採用･不採用の決定



$\sigma(k)$ ：k番目に優先されるプロジェクト番号

4．2 順列空間の局所探索(3)

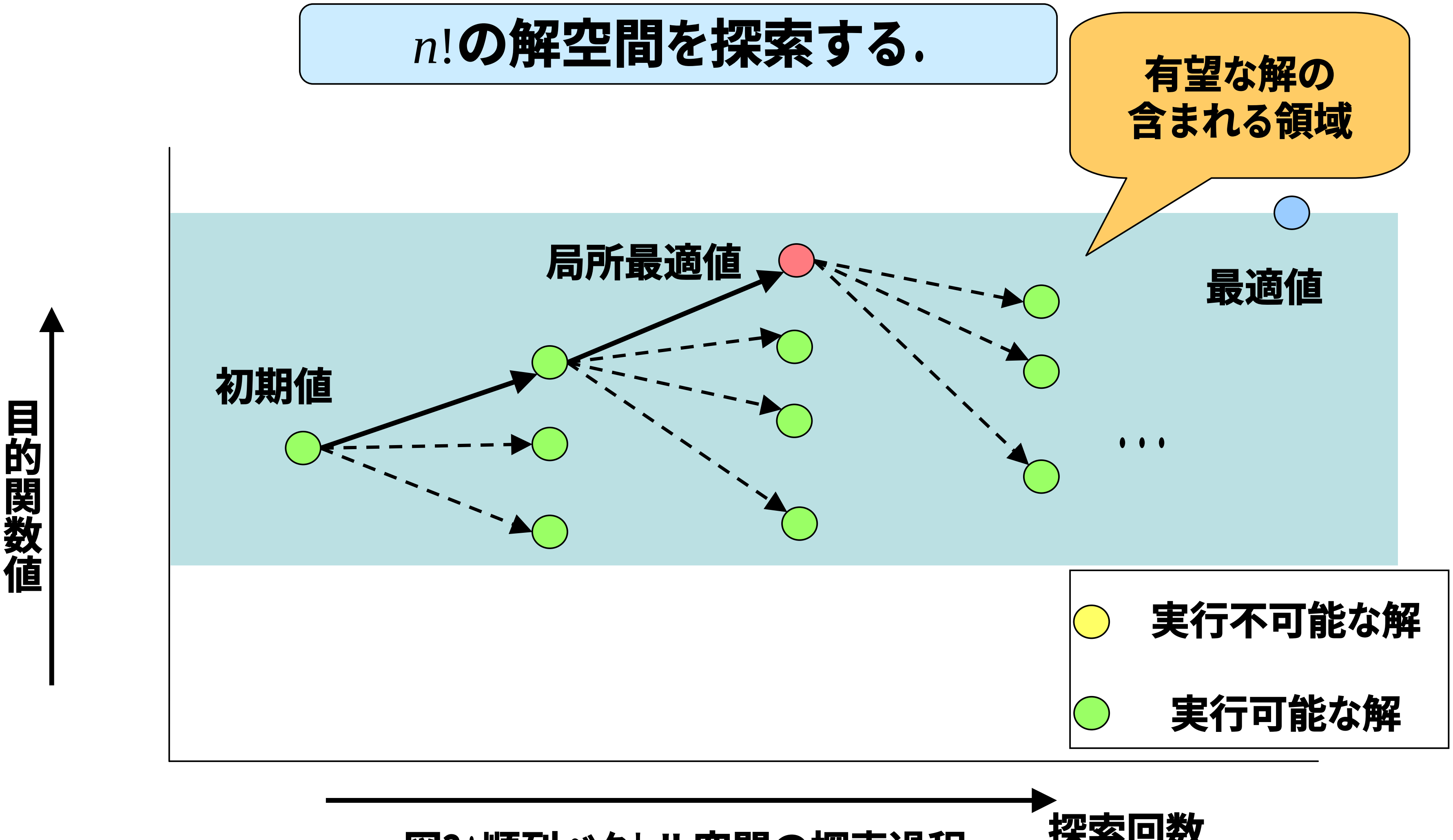


図3：順列ベクトル空間の探索過程

5．局所探索法の初期解の生成

局所探索をするには
初期解が必要

⇒

優良な解を求めるには
どのような初期解を採用するべきか？

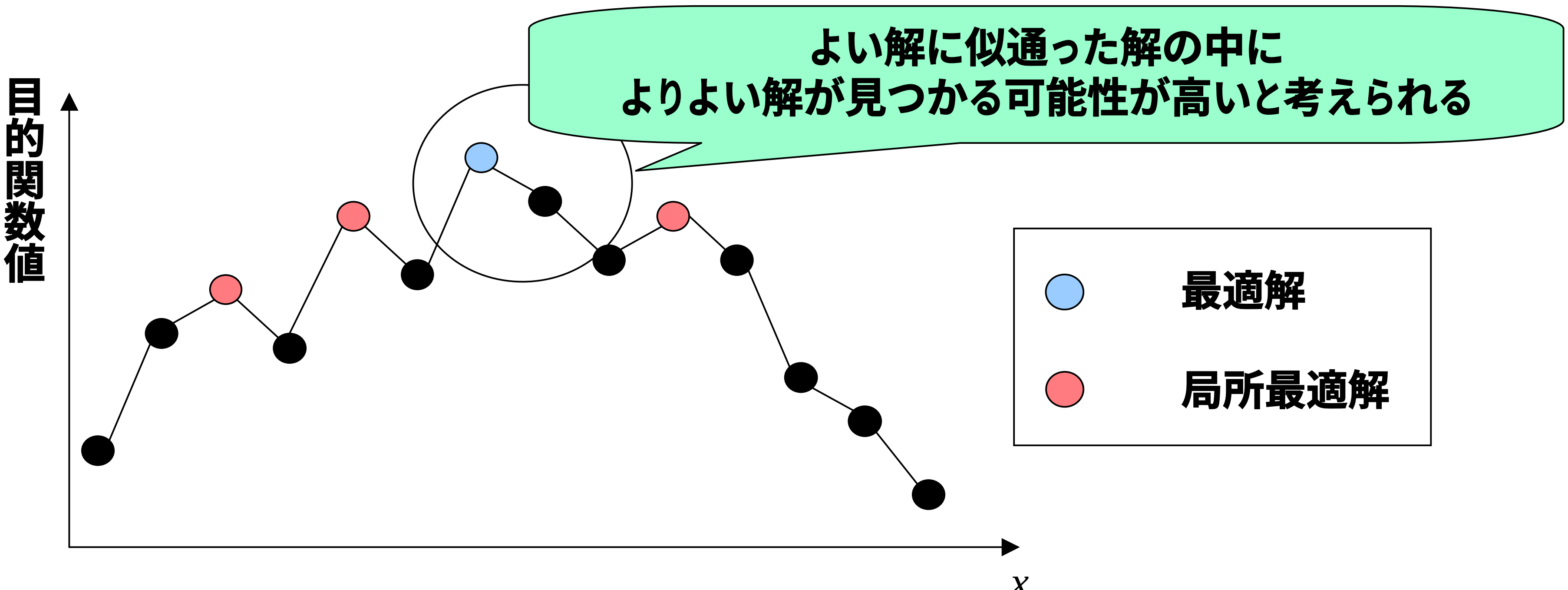


図4：解と目的関数値

初期解はなるべく精度が
良い解を採用すべき

⇒ 卒業論文審査

構築法によって初期解を求める

5．1 構築法

構築法：目的関数への貢献度を示す<u>局所的な評価値</u>に基づいて，実行可能解を直接構成する．

有効勾配法

x=(1,1,…,1) から探索スタート

代理制約式法

x=(0,0,…,0) から探索スタート

5．2 有効勾配法

局所的な評価値

有効勾配 c_j/h_j

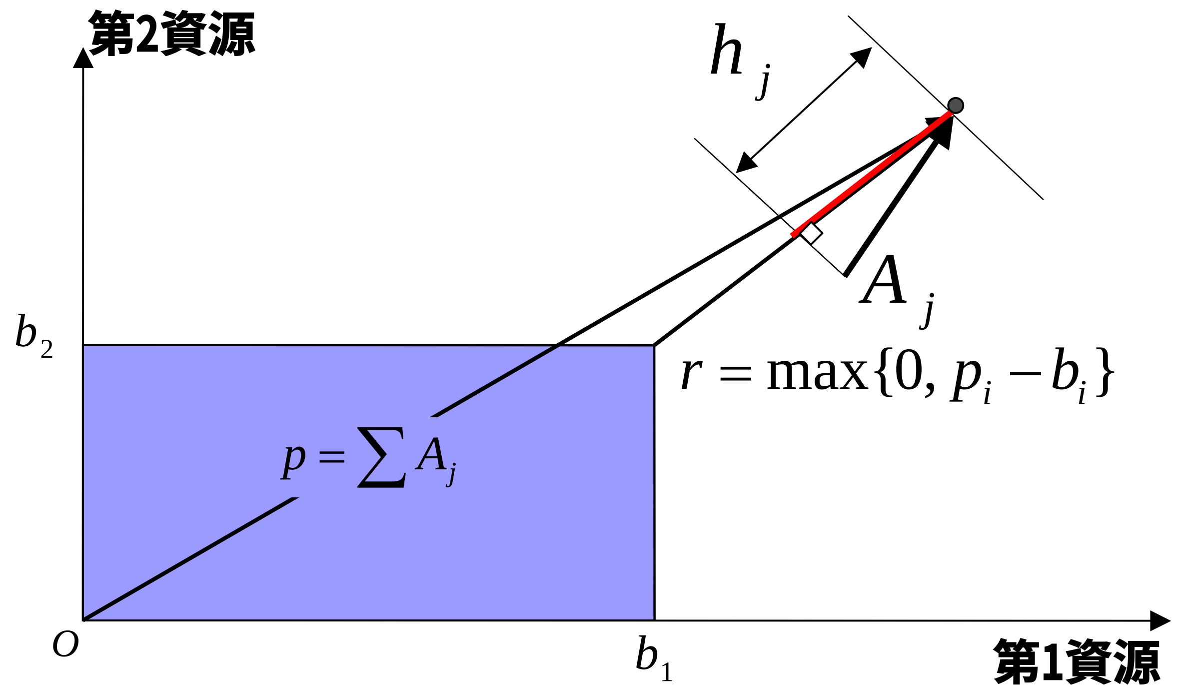


図5：A_j の r への正射影の長さ h_j (2次元の場合)

有効勾配を基準にできるだけ利益が大きくなるプロジェクト集合を求める．

5．3 代理制約式法

MKPの連続緩和問題を解き，shadow price(*sp(i)*)を用いて複数の制約式を1つの制約式にし，欲張り法で解く．

↓

局所的な評価値

↓

単位資源あたりの利益

$h_j = c_j / a'_j$

$$sp(1)\times(a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n) \le sp(1)\times b_1$$

$$\cdots$$

$$+)\quad sp(m)\times(a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n) \le sp(m)\times b_m$$

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m} sp(i)\cdot a_{ij}\right)\cdot x_{ij} \le \sum_{i=1}^{m} sp(i)\cdot b_i$$

（ここで，$\sum_{i=1}^{m} sp(i)\cdot a_{ij} = a'_j$ とする）

5．4　予備実験の結果･考察

プログラムはBorland社のDelphi 6を用いて作成し，データはOR-LIBRARYにあるMKPのデータを用いた．

表1：数値実験結果

Problem Name ＼ 解法	有効勾配法		代理制約式法		最適値
	目的関数値	有効勾配法/最適値	目的関数値	代理制約式法/最適値	
Petersen06 (5 knapsacks, 39 objects)	10506	0.9895	10547	0.9933	10618
Petersen07 (5 knapsacks, 50 objects)	16440	0.9941	16436	0.9939	16537
WEING4 (2 knapsacks, 28 objects)	119337	1.0000	104799	0.8782	119337
WEING6 (2 knapsacks, 28 objects)	130213	0.9969	130233	0.9970	130623
WEING8 (2 knapsacks, 105 objects)	620060	0.9932	619101	0.9916	624319

実験結果より，有効勾配法と代理制約式法は優劣がつけ難い．

そこで，2つの解法で求まった解を局所探索法によって逐次改善する．

6．実験①

0-1空間の探索（既存の近似解法）

初期解 ← 構築法（代理制約式法，有効勾配法）の解

性能の比較をおこなう

順列空間の探索（新しい近似解法）

初期解 ← 構築法（代理制約式法）の解

6．1 実験結果･考察①

表2：数値実験結果

解法 / Problem Name	解法①		解法②		解法③	
	目的関数値	「0-1」/最適値	目的関数値	「0-1」/最適値	目的関数値	「順列」/最適値
Petersen06	10507	0.9895	10584	0.9968	10584	0.9968
Petersen07	16440	0.9941	16443	0.9943	16499	0.9977
WEING4	119337	1.0000	116622	0.9772	117452	0.9842
WEING6	130233	0.9970	130233	0.9970	130233	0.9970
WEING8	620060	0.9932	619766	0.9927	623612	0.9989

解法①：有効勾配法の解を初期解とした0-1空間の探索
解法②：代理制約式法の解を初期解とした0-1空間の探索
解法③：代理制約式法の解を初期解とした順列空間の探索

実験結果より，代理制約式法の解を初期解とする順列空間の探索が良い解が導かれた．

７．メタ戦略(1)

順列空間の探索（新しい近似解法）

初期解 ← 構築法（代理制約式法）の解

さらに解の改善を図るために

↓

メタ戦略を組み込んだ局所探索法

初期解 ← 構築法（代理制約式法）の解

7．メタ戦略(2)

局所探索では

局所最適解で探索が終了してしまうため，
他によりよい解があっても移動できない

この問題を解決するために

メタ戦略では

多少は時間がかかるが，より良質の解を求めることが出来る

タブー探索法，**適応的多スタート法**を導入する．

7．1 タブー探索法(1)

現在の解の近傍 $N(x)$ から最良の解を次の解として選ぶ方法.

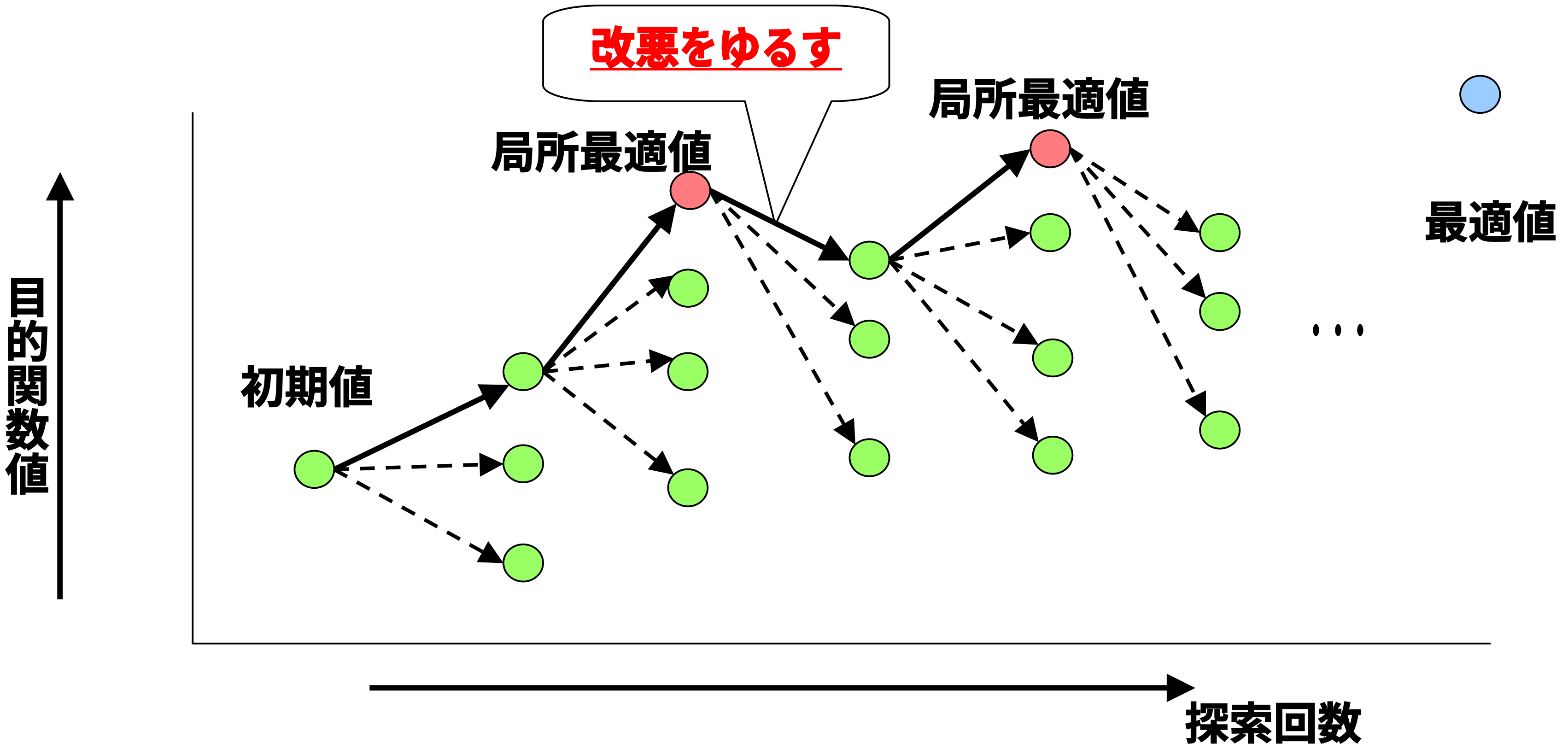


図6：タブー探索

7．1 タブー探索法(2)

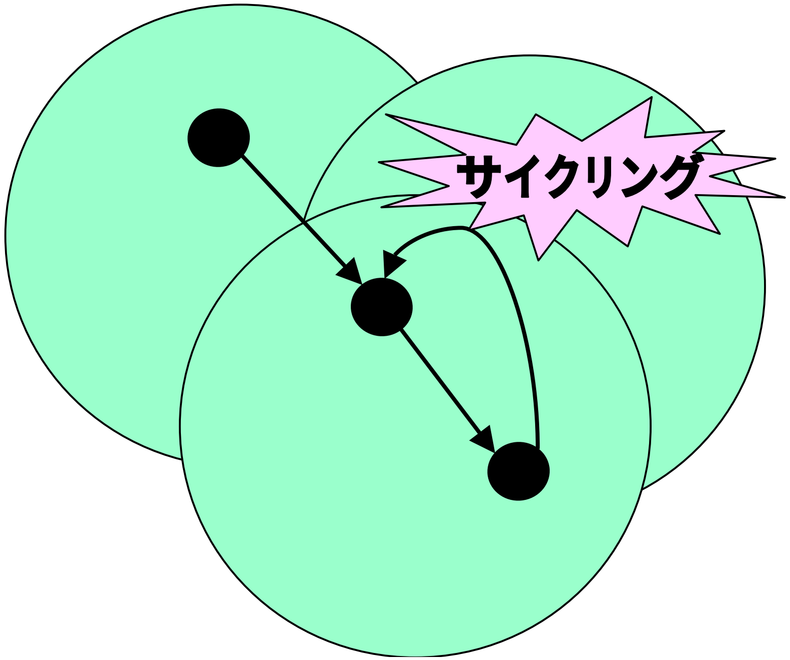


図7：サイクリング

サイクリングを避けるために，タブーリストTと呼ばれる解集合を用意し，これに含まれる解への移動を禁止する．

７．２ 適応的多スタート法(1)

過去の探索で得られたよい解にランダムな変形を加えたものの複数個を初期解とする方法．

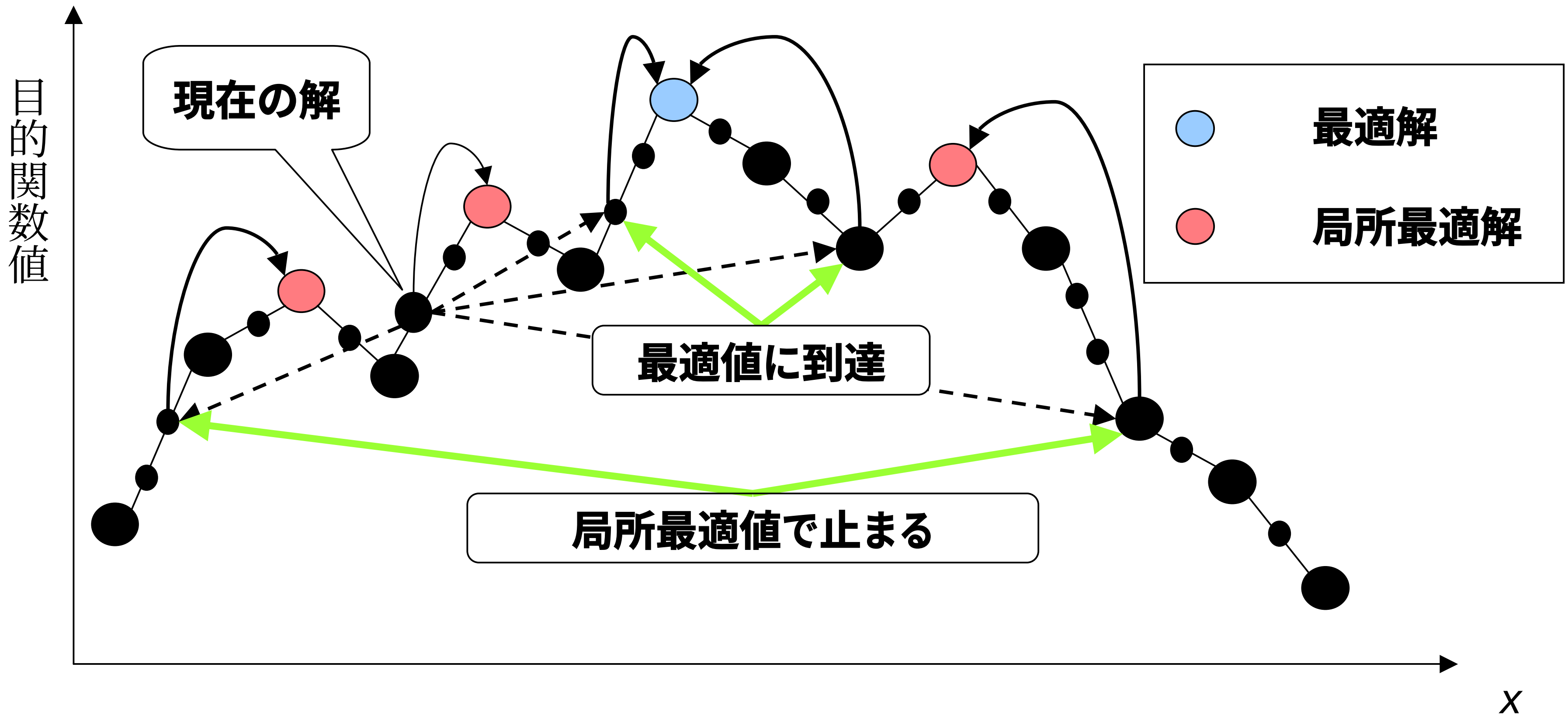


図8：多スタート法

7．2 適応的多スタート法(2)

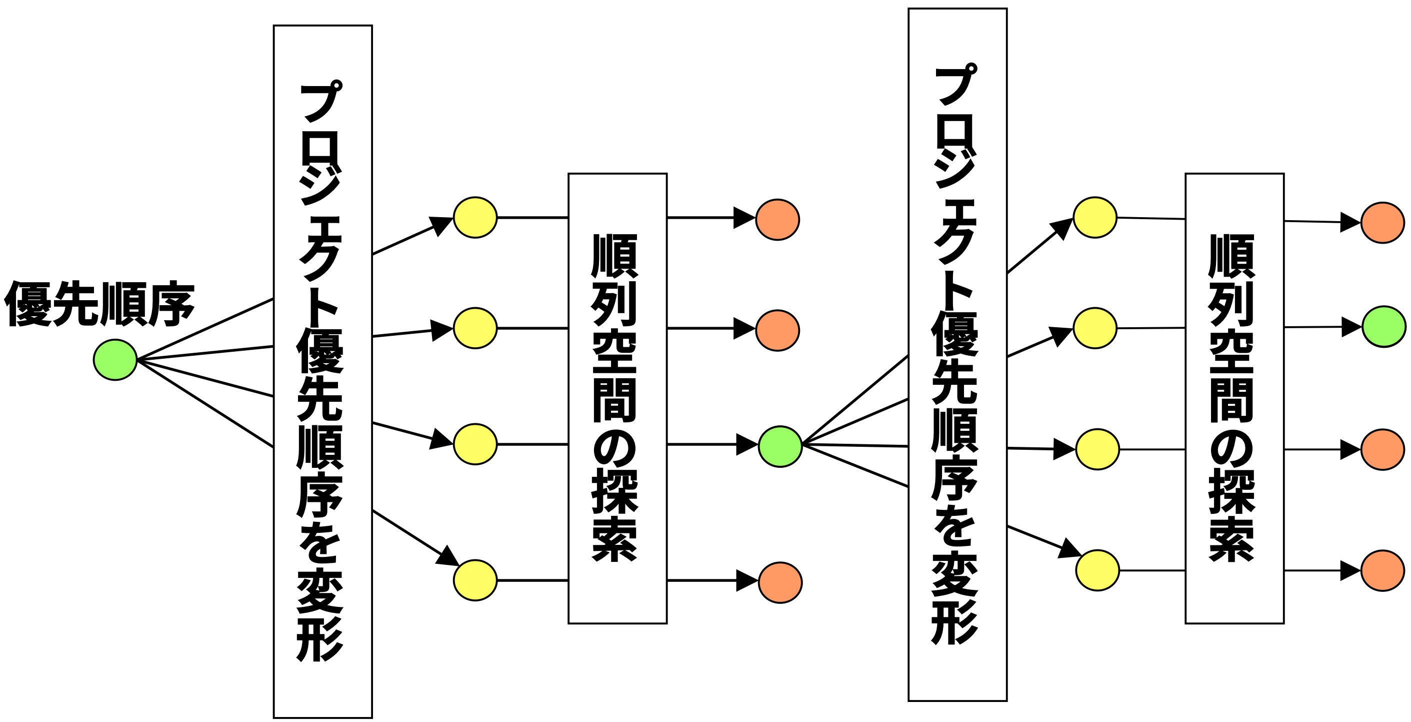


図9：適応的多スタート法と順列空間の局所探索

8．実験②

<u>タブー探索を組み込んだ順列空間の探索（解法④）</u>

初期解　←　<u>構築法</u>（代理制約式法）の解

性能の比較をおこなう

<u>適応的多スタート法を組み込んだ順列空間の探索（解法⑤）</u>

初期解　←　<u>構築法</u>（代理制約式法）の解

8．1 実験結果・考察②(1)

表3：数値実験結果

Problem Name ＼ 解法	解法④ 目的関数値	解法⑤ 目的関数値	最適値
Petersen06	10618	10618	10618
Petersen07	16537	16537	16537
WEING4	119337	119337	119337
WEING6	130623	130623	130623
WEING8	624319	624319	624319

解法④：代理制約式法の解を初期解としてタブー探索を導入した順列空間の探索

解法⑤：代理制約式法の解を初期解として適応的多スタート法を導入した順列空間の探索

全ての問題に対して，メタ戦略を導入することで最適値を求めることができた．

次に，さらに難しい問題を解いた．

8．2 実験結果・考察②(2)

表4：数値実験結果

解法 / Problem Name	解法④	解法⑤		最良値
	目的関数値	目的関数値	最良値が求まった回数	
30.100_00 (30 knapsacks, 100 objects)	21764	21946	2/10	21946
30.100_10 (30 knapsacks, 100 objects)	40607	40628	0/10	40767
30.100_20 (30 knapsacks, 100 objects)	57170	57494	10/10	57494

解法④：代理制約式法の解を初期解としてタブー探索を導入した順列空間の探索
解法⑤：代理制約式法の解を初期解として適応的多スタート法を導入した順列空間の探索

実験結果より，タブー探索より適応的多スタート法を導入した方がよい解が得られた．

9．まとめ

・新しく提案した順列空間の探索は既存の解法より，良い性能を示した．

・MKPに対する有効なアルゴリズムとして「代理制約式法の解を初期解として適応的多スタート法を導入した順列空間の探索」を提案した．

１０．今後の課題

－提案したアルゴリズムの改善

－計算時間の短縮

【参考文献】

[1]柳浦 睦憲, 茨木 俊秀 : 組合せ最適化ーメタ戦略を中心としてー ; 朝倉書店, 2000.

[2]今野 浩, 鈴木 久敏 : 整数計画問題と組み合わせ最適化 ; 日科技連, 1982.

[3]J.E.Beasley : OR-LIBRARY, http://www.ms.ic.ac.uk/info.html