多次元 0-1 ナップサック問題に対する修復オペレーターを用いた局所探索法の提案

秀方 友一郎（沼田 一道 助教授）

1. 本研究の背景と目的

現実社会に現れる最適化問題は，組合せ最適化問題として定式化されることが多い．組合せ最適化問題は，その規模が小さい場合，全列挙等の原理的方法でも厳密解を求めることができる．しかし，多くの場合，問題の規模が大きくなると計算量が爆発的に増大し，厳密解を求めることは困難になる．そこで，短時間で高精度の解を求めることができる近似解法が必要となる．

本研究ではそのような問題の例として多次元 0-1 ナップサック問題を取り上げ，それに対する近似解法として修復オペレーターを用いた 0-1 空間の局所探索法を提案し，従来の 0-1 空間の局所探索法と比較する．また，メタ戦略を組込む事により得られる近似解の精度向上を目指す．

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

n 個のプロジェクトがあり，c_j をプロジェクト j を実行することにより得られる利益，a_{ij} をプロジェクト j が必要とする第 i 資源の量，b_i を第 i 資源の利用可能量とする．プロジェクト j を実行するとき $x_j = 1$, 実行しないとき $x_j = 0$ とすると, 多次元 0-1 ナップサック問題は以下の(**P**)のように定式化される．

$$
(\mathbf{P})\quad
\left|\begin{array}{lll}
\text{maximize} & \sum_{j=1}^{n} c_j x_j \quad j = 1,2,\ldots,n & (2.1) \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \le b_{i,}\ i = 1,2,\ldots,m & (2.2) \\
 & x_j \in \{0,1\} &
\end{array}\right.
$$

目的関数と制約式の全ての係数 i, j に対して $c_j > 0, a_{ij} \ge 0, b_i > 0, \sum_{j=1}^{n} a_{ij} > b_i$ となる場合を考える．

3. ＭＫＰに対する代表的な解法

3.1 構築法

目的関数に対する局所的な評価値に基づいて実行可能解を直接構成する方法で，代表的なものとして有効勾配法や代理制約式法などが知られている．

3.2 改善法

与えられた解を簡単な操作によって改善する手続きを繰り返す方法で，代表的なものとして 0-1 空間及び順列空間の局所探索法などが知られている．

3.3 メタ戦略

局所探索などで得られた解の周辺をより丹念に探索する方法で，構築法，改善法よりも多少計算時間は掛かるがより精度の良い解を求める事ができる．代表的なものとして，遺伝的アルゴリズムやタブー探索法などが知られている．

4. アルゴリズム作成の方針

本研究では，短時間で比較的良い解を算出する 0-1 空間の局所探索法に注目する．0-1 空間の局所探索法は実行可能性の問題があるので，修復オペレーターを用いることによりスムーズに解を探索できるようにする．さらに，局所最適解にランダムな変形を加えて探索を続ける反復局所探索戦略を導入し精度の向上を目指す．

5. 局所探索法

局所探索法は，現在の解を少し変更してできる解集合（近傍）を探索し，改善解があれば更新していく方法である．そのため，どのようなものを近傍として採用するかが良い近似解を算出する上で重要となる．本研究では，解の 0-1 ベクトルを少し変更したものを近傍として採用し 0-1 空間を探索する．

従来の 0-1 空間の局所探索法は，実行可能領域のみを探索し，その中で最良なものに改善していく方法であった．本研究では，修復オペレーターと呼ばれる実行不可能解を実行可能解に対応させる操作を導入し，従来の 0-1 空間の探索と比べより丹念な探索を期待する．また，修復オペレーターはそれ程時間を要しない作業なので，計算時間を増加させずに解の精度を向上できると考えられる．

5.1 従来の 0-1 空間の局所探索法のアルゴリズム

Step1　現在の解 x の 0-1 を 1 ヶ所変更した解集合を $N_1(x)$，2 ヶ所変更した解集合を $N_2(x)$ とし $N_1(x) \cup N_2(x)$ を x の近傍 $N(x)$ とする．

Step2　制約 $\sum_{j=1}^{n} a_{ij} x_j^{'} < b_i (i=1,\ldots,m)$ を満たし（実行可能解），目的関数値が最大となるような $x^{'} \in N(x)$ を選ぶ．

Step3　$\sum_{j=1}^{n} c_j x_j^{'}$ が目的関数値を更新していれば $x = x^{'}$ とし Step1 へ戻る．更新されていなければ x を局所最適解として出力する．

5.2 本研究で提案する修復オペレーターを用いた 0-1 空間の局所探索法のアルゴリズム

Step1　現在の解 x の 0-1 を 1 ヶ所変更した解集合を $N_1(x)$，2 ヶ所変更した解集合を $N_2(x)$ とし $N_1(x) \cup N_2(x)$ を x の近傍 $N(x)$ とする．

Step2　制約 $\sum_{j=1}^{n} a_{ij} x_j^{'} < b_i (i=1,\ldots,m)$ を満たすものと，実行不可能解を修復オペレーターを用いて実行可能解に対応させたものの中で，目的関数値が最大となるような $x^{'} \in N(x)$ を選ぶ．

Step3　$\sum_{j=1}^{n} c_j x_j^{'}$ が目的関数値を更新していれば $x = x^{'}$ とし Step1 へ戻る．更新されていなければ x を局所最適解として出力する．

6. 代理制約式と修復オペレーター

修復オペレーターは CHU らが文献[2]の中で提唱した方法である．修復オペレーターを用いることで実行不可能解を実行可能解に対応させることができる．その際，何らかの方法で採択優先順序を決めておく必要がある．ここでは，以下のように代理制約式を用いて順序付けを行う．

6.1 代理制約式

MKP の連続緩和問題を解き，shadow price を用いて複数の制約式を 1 つの制約式（代理制約式）に集約し，代理制約式から単位資源あたりの利益を効率値として求める．この効率値を大きい順に並べたものを，修復オペレーターで用いる採択優先順序とする．shadow price とは最適解における各制約式の重みのことであり，この重みを各制約式に乗じて加え合わせ代理制約式を作成する．

6.2 修復オペレーターのアルゴリズム

Step1　代理制約式により求めた各プロジェクトの効率値の大きな順に，プロジェクトを並び替える．これを採択優先順序とする．

Step2　採択優先順序の小さな方から見て順番に制約条件を満たすまで 1→0 に変更(drop phase)

Step3　採択優先順序の大きな方から見て順番に制約条件を満たす限り 0→1 に変更(add phase)

修復オペレーターのアルゴリズムを小規模な MKP に適用した例を以下に示す．

$$\text{maximize} \quad 100x_1 + 600x_2 + 1200x_3 + 2400x_4 + 500x_5 + 2000x_6 \qquad (6.1)$$

$$\text{subject to} \quad 8x_1 + 12x_2 + 13x_3 + 64x_4 + 22x_5 + 41x_6 \le 80 \qquad (6.2)$$

$$3x_1 + 6x_2 + 4x_3 + 18x_4 + 6x_5 + 4x_6 \le 20 \qquad (6.3)$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \in \{0,1\}$$

局所探索で実行不可能解 $(x_1, x_2, x_3, x_4, x_5, x_6) = (1,1,1,1,0,0)$ **を評価するときのの制約の左辺は**(6.2),(6.3)**の順に** 97 ,31

代理制約式で求めたプロジェクト採択優先順序 (3,2,6,4,5,1)

		制約の左辺 (6.2) (6.3)	解の変化
drop phase	$x_1 : 1 \to 0$ **に変更**	89 , 28	(0,1,1,1,0,0)
	$x_4 : 1 \to 0$ **に変更**	25 ,10	(0,1,1,0,0,0)
add phase	$x_6 : 0 \to 1$ **に変更**	66 ,14	(0,1,1,0,0,1)
	$x_1 : 0 \to 1$ **に変更**	74 ,17	(1,1,1,0,0,1)

7. メタ戦略の適用-反復局所探索法-

反復局所探索法とは過去の探索で得られた良い解にランダムな変形を加えたものを初期解として，単純局所探索方法を反復する方法である．これにより，過去に得られた良い解の周辺をより丹念に探索するとともに，ランダムな変形によって，これまでの探索とは多少異なる領域を調べることができる．

8. 有効勾配法を用いた初期解の生成

本研究では，0-1 空間の局所探索法の初期解（出発地点）として有効勾配法を用いることにした．有効勾配法は非常に短時間で，高精度な近似解を得ることができる方法として広く知られている．

全プロジェクトを実施した時のプロジェクト j が必要とする資源ベクトル A_j の和から実行可能領域の境界面に至る最短の長さを持つベクトル r に A_j を正射影した長さ h_j が長ければより実行可能領域に近づく．また，プロジェクト j によって得られる利益 c_j が小さければより目的関数値を減らさずにすむ．そのため $g_j = c_j / h_j$ を有効勾配と定義する．この有効勾配の値の小さい方から順にプロジェクトを除外していき，資源の利用可能量を超えない範囲でできるだけ利益を最大化するプロジェクトの組合せを求める方法である．2 次元の場合の，h_j の様子を図 8.1 に示す．

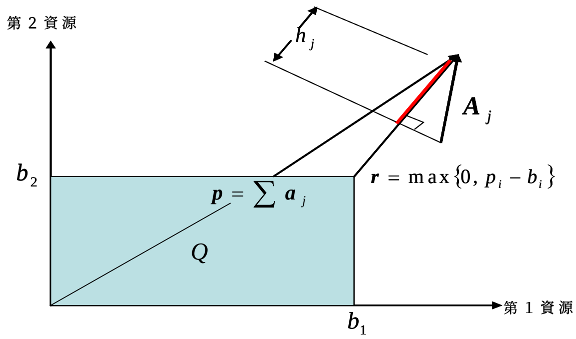

図 8.1 A_j の r 方向への正射影 h_j （2 次元の場合）

9. 実験

9.1 実験概要

MKP に対する従来の局所探索法で得られた解と本研究で提案する解法で得られた解の精度を，最適値からの乖離値（Gap）によって比較する．テストデータには OR-LIBRARY にある MKP のデータを用いた．プログラムの作成には Borland Delphi6 を用いた．

9.2 実験結果

従来の 0-1 空間の局所探索法（解法①），本研究で提案する修復オペレーターを用いた 0-1 空間の局所探索法（解法②）の実験結果を表 1 に示す．ただし，解法①と解法②は有効勾配法の解を初期解とする．

表 1 の全ての結果は実行開始とほぼ同時に出力されたので計算時間を示していない.

表 1 実験結果（解法①，解法②）

解法 / Problem Name	有効勾配法(初期解)		解法①		解法②		最適値
	目的関数値	Gap(%)	目的関数値	Gap(%)	目的関数値	Gap(%)	
Weing1(28変数2制約)	140477	0.5669	140477	0.5669	141278	0	141278
Weing3 (28変数2制約)	95627	0.0522	95627	0.0522	95627	0.0522	95677
Weing6 (28変数2制約)	130213	0.3139	130233	0.2986	130233	0.2986	130623
Weing7 (105変数2制約)	1095112	0.0304	1095112	0.0304	1095382	0.006	1095445
Weing8 (105変数2制約)	620060	0.6822	620060	0.6822	621086	0.5178	624319
Sento1 (60変数30制約)	7648	1.605	7648	1.605	7758	0.1801	7772
Sento2 (60変数30制約)	8697	0.2866	8697	0.2866	8722	0	8722
Petersen2 (10変数10制約)	8337	4.240	8337	4.240	8706.1	0	8706.1
Petersen5 (28変数10制約)	11810	4.758	11890	4.113	12400	0	12400
Petersen6 (39変数5制約)	10506	1.055	10570	0.4521	10570	0.4521	10618
100-30-1 (100変数30制約)	21375	2.602	21557	1.773	21835	0.5058	21946
100-30-2 (100変数30制約)	40058	1.739	40318	1.101	40458	0.7580	40767
100-30-3 (100変数30制約)	56677	1.421	56955	0.937	57494	0	57494

表 1 から，解法②が解法①よりも精度の良い解を出力していることが分かる．これは修復オペレーターを用いたことにより，探索の範囲が広がったためと考えられる．しかし，いくつかの問題では最適値に到達せずに探索が終了している．

そこで，解法②に反復局所探索戦略を適用した解法③を構成し，実験を行った．その結果を表 2 に示す．解法③では小規模な問題に対して短時間で最適値を求めることができた．100 変数 30 制約式の難しい問題に対しても長時間の実行の後に，最適解あるいはそれに近い解を発見することができた.

表 2 実験結果（解法③）

解法 / Problem Name	解法③			最適値
	目的関数値	Gap(%)	計算時間 (ミリ秒)	
Weing3 (28変数2制約)	95677	0	100	95677
Weing6 (28変数2制約)	130623	0	200	130623
Weing7 (105変数2制約)	1095445	0	181	1095445
Weing8 (105変数2制約)	624319	0	861	624319
Sento1 (60変数30制約)	7772	0	981	7772
Petersen6 (39変数5制約)	10618	0	901	10618
100-30-1 (100変数30制約)	21946	0	246214	21946
100-30-2 (100変数30制約)	40687	0.196	620032	40767

10. まとめ

本研究では多次元 0-1 ナップサック問題に対する新たな近似解法として，有効勾配法の解を初期解とし修復オペレーターを用いて 0-1 空間を局所探索法するというアルゴリズムを提案した．従来の 0-1 空間の局所探索法と比較したところ優位性が認められ，さらにメタ戦略を組込む事により得られる近似解の精度は向上した．難しいとされる問題例に対しても計算時間の点を除けば満足な結果を与えた．計算時間を改善することは今後の課題である.

【主要参考文献】

[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」，朝倉書店，2001

[2]P.C.CHU AND J.E.BEASLEY：“ A Genetic Algorithm for the Multidimensional Knapsack Problem ”，Journal of Heuristics,4，pp.63-86，1998

[3]J.E.BEASLEY:OR-Library(http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mknapinfo.html)，2005/12/19