多次元0-1ナップサック問題に対する近似解法の研究

石井 宏明　　伊藤 祐介（沼田 一道 助教授）

1．はじめに

現実社会に現れる様々な最適化問題の多くは組合せ最適化問題として定式化される．組合せ最適化問題は，問題の規模が小さい場合には全数探索によって厳密解を求めることが可能である．しかし，問題の規模が大きくなってしまうと組合せ数が爆発的に増え，厳密解を求めることは困難になる．よって，短時間で高精度な解が得られる近似解法が必要となる．

本研究では，標準的な組合せ最適化問題の1つである多次元0-1ナップサック問題を取り上げ，それに対する新しい近似解法として順列空間の局所探索法を提案し，既存の0-1ベクトル空間の探索と比較する．また，局所探索をもとに，適用するメタ戦略を計算実験によって検討し，より良い近似解算出のアルゴリズムを提案する．

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

いくつかのプロジェクトがあり，それらに対する資源制約が複数あるとき，「資源制約を満たした上で利益を最大化するにはどのような組合せでプロジェクトを採用すればよいか」という問題が多次元0-1ナップサック問題である．

2．1　MKPの定式化

多次元0-1ナップサック問題は，一般的に次のように定式化される．

$$
(\mathrm{MKP})\begin{cases}
\text{maximize} & z=\sum_{j=1}^{n} c_j x_j \quad j=1,2,\ldots,n & (1)\\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i=1,2,\ldots,m & (2)\\
 & x_j \in \{0,1\} &
\end{cases}
$$

c_j はプロジェクト j を実施することによって得られる利益，a_{ij} はプロジェクト j が必要とする第 i 資源の量，b_i は第 i 資源の利用可能量，x_j はプロジェクト j を採用する（$x_j=1$）かしない（$x_j=0$）かを表す決定変数である．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．アルゴリズム作成の方針

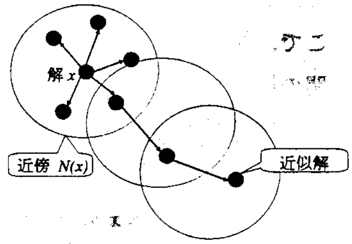

図１：近傍と局所探索

本研究では，局所探索法の枠組みをアルゴリズムの中心としてMKPに適用する．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，現在の解より良い解があれば改善していく方法である（図1）．よい解の近傍によりよい解が見つかる可能性が高いと考えられるので，初期解はなるべく精度が良い解を採用すべきである．そこで，まず構築法によって良い解を求めておき，局所探索法によって逐次改善することを考える．

次に，これまでに得られた良い解と似通った構造を持つ解を集中的に探索することによって，局

所最適解の周辺に潜んでいるよりよい解を発見しようとする．しかし，似通った構造の解を集中的に探索すると，同じ解を何度も探索してしまって，無駄が多くなる恐れがある．よって，ときどきは，これまで生成してきた解とは構造の異なる解を生成することも必要である．そこで，メタ戦略であるタブー探索と適応的多スタート法を導入する．

4．局所探索法の出発点となる初期解の生成方法

本研究では，局所探索法の初期解の生成方法として既存の有力な構築法である有効勾配法と代理制約式法を試みる．

4．1　有効勾配法

全プロジェクトを実施したと仮定したときに，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 $g_j = c_j / h_j$ を有効勾配と定義し，有効勾配を局所的な評価値としてプロジェクトを除外していき，そのときの資源要求量が資源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集合を求める．

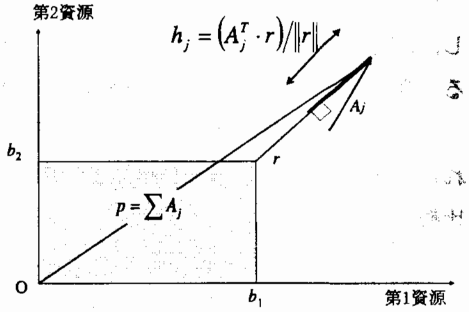

図２：A_j の r への正射影の長さ h_j（2次元の場合）

4．2　代理制約式法

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）にし，代理制約式から単位資源あたりの利益を評価値として，欲張り法で解く方法である．shadow priceとは各制約式の重みのことであり，この重みを各制約式に掛けて代理制約式を作成する．

5．局所探索法

本研究では，既存の局所探索の評価を通して新しい局所探索法を提案し，数値実験によってその性能を比較する．局所探索法では，近傍を探索し，現在の解より良い解があれば改善していく．そのため，どのようなものを近傍として採用するかが初期解の選択と並んでよい近似解を出す上で重要である．一般的な局所探索では，解の0-1ベクトルを少し変更したものを近傍として0-1ベクトル空間の探索を行う．この探索は2^nのオーダで探索を行うため，短時間で解を求めることが可能である．しかし，解の求め方が実行可能領域だけでなく実行不可能領域も探索するため無駄が多い．そこで，新たな探索方法としてプロジェクト採用の優先順序を少し変更したものを近傍とする順列空間の探索を取り上げる．この探索のオーダは$n!$であるが，実行可能領域のみを探索するためより細かい探索が可能であると考えられる．

5．1　0-1空間の探索のアルゴリズム

Step1. 現在の解 x の0-1を1ヵ所変更した解集合を$N_1(x)$，
2ヵ所変更した解集合 $N_2(x)$ とし，
$N_1(x) \cup N_2(x)$ を近傍 $N(x)$ とする（図３）．

Step2. 制約 $\sum_{j=1}^{n} a_{ij} x'_j < b_i \ (i = 1, \cdots, m)$ を満たし，

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

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

図３：0-1ベクトル空間における近傍の例

目的関数値 $\sum_{j=1}^{n} c_j x'_j$ 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,

目的関数値が更新されていなければ x を解として出力する.

5．2． 順列空間の探索のアルゴリズム

Step1. プロジェクト採用の優先順序 $s(k)$ を2ヵ所交換し,

順列 s から新しい順列集合$N(s)$を作る(図４).

Step2. $a_{is'(k)} < b_i \, (i = 1,2,\ldots,m)$ ならば $x_{s'(k)} := 1$,

$b_i := b_i - a_{is'(k)}$ として暫定解をだし,

目的関数値 $\sum_{j=1}^{n} c_{\sigma'(j)} x_{\sigma'(j)}$ が最大となるような $\sigma' \in N(x)$ を選ぶ.

Step3. $\sum_{j=1}^{n} c_{\sigma'(j)} x_{\sigma'(j)}$ を計算し目的関数値が更新されていれば, $s := s'$としてStep1へ,

更新されていなければ $x_{s(k)}$ を解として出力する.

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

⬇

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

図４：順列空間における近傍の例

6．メタ戦略(metaheuristics)の適用

本研究では，メタ戦略としてタブー探索と適応的多スタート法を用いる．

6．1 タブー探索(tabu search)

タブー探索法は，現在の解 x を除く近傍$N(x)$ 内の最良の解を次の解として選ぶ．これによって現在の解が仮に局所最適解であっても他の解への移動が強制される．これによって幅広い探索が可能になる．しかし，これをそのまま実行すると，x から他の解 $x' \in N(x)$ に更新されたのち，$N(x')$ 内の最良の解を求めると，再び，x に戻ってしまう可能性が高い．そこで，このようなサイクリングをさけるため，タブーリストという解集合 T を作り，$N(x) \setminus (\{x\} \cup T)$内の最良解 x'へ移るという解法である．

6．2 適応的多スタート法(adaptive multi-start method)

適応的多スタート法とは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を初期解とする方法である．こうすることにより，多スタート法の異なる初期解に局所探索法を適用して得られる複数の局所最適解の中から最良のものを選ぶことによって，1回の局所探索による危険性を避けるという利点に加え，過去に得られた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．

7．実験の概要

MKPに対する既存の解法で得られた解と新しい解法で得られた解の精度を比較する．なお，プログラム言語には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．Petersen0*は，C.C.Petersenの「Computational experience with variants of the Balas algorithm applied to the selection of R&D projects」からのテスト問題である．また, WEING*は，P.C.ChuおよびJ.E.Beasleyの「A genetic algorithm for the multidimensional knapsackproblem」の中で解決された問題であり, 30.100_**の最良値とは, その中で与えられたアルゴリズムによって見つかった最良値である．

7．1　実験結果・考察

初めに，局所探索法の初期解として適用する有効勾配法・代理制約式法それぞれについて，数値実験の結果を表１に示す．

表1:数値実験結果

解法 / Problem Name	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

表１より，有効勾配法の目的関数値と代理制約式法の目的関数値を比較すると同等の精度が出た．

次に，有効勾配法の解を初期解とした0-1空間の探索（解法①）・代理制約式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順列空間の探索（解法③）それぞれについて，数値実験の結果を表２に示す．表２より，0-1空間の探索では初期解の設定を2種類で試みたが優劣がつけ難い．そこで，代理制約式法の解が適用できる順列空間の探索を行うと，0-1空間の探索と順列空間の探索では後者の方が精度の良い解を求めることができた．これは，解の求め方が実行可能領域内の有望な解の含まれる領域のみを探索していることによるものと思われる．

さらに，代理制約式法の解を初期解とする順列空間の探索をもとにタブー探索（解法④）と適応的多スタート法（解法⑤）を実行した(表２)．局所探索法だけで最適値が得られなかったデータについても，メタ戦略を導入することにより解が改善され最適値を得ることができた．

表2:数値実験結果

解法 / Problem Name	解法①		解法②		解法③		解法④	解法⑤
	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	目的関数値	目的関数値
Petersen06	10507	0.9895	10584	0.9968	10584	0.9968	10618	10618
Petersen07	16440	0.9941	16443	0.9943	16499	0.9977	16537	16537
WEING4	119337	1.0000	116622	0.9772	117452	0.9842	119337	119337
WEING6	130233	0.9970	130233	0.9970	130233	0.9970	130623	130623
WEING8	620060	0.9932	619766	0.9927	623612	0.9989	624319	624319

そこで，さらに難しい問題について解法④・⑤で解き，その実験結果を表３に示す．解法④では，タブー期間や反復回数が非常に多く，計算時間もかかったので，順列空間の探索とタブー探索は相性が悪いといえる．しかし，解法⑤では数値実験を10回行った結果，30.100_00と30.100_20についてはそれぞれ最良値を求めることができ，30.100_10についても最良値を求めることはできなかったが，タブー探索より良い解が得られた．

表3:数値実験結果

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

8．おわりに

本研究では，実験結果から多次元0-1ナップサック問題に対して「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」というアルゴリズムを提案した．このアルゴリズムは，ある程度の大きさの問題に対しては非常にすばらしい精度を持っていると言える．

今後は，適応的多スタート法を改良し，さらに精度のよい解を出すことが課題である．

【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，2001.

[2]今野 浩，鈴木 久敏：整数計画法と組合せ最適化；日科技連，1982.