複数制約非線形ナップザック問題に対する スマートグリーディ法

亀高　哲夫・太田垣博一*・岩崎　彰典**・仲川　勇二***
成久　洋之****
岡山理科大学大学院工学研究科電子工学専攻
*岡山理科大学工学部電子工学科
**岡山理科大学情報処理センター
***関西大学総合情報学部
****岡山理科大学工学部情報工学科
（1995年 9 月30日　受理）

1．まえがき

複数制約非線形ナップザック問題は信頼性工学における最適化問題をはじめとして工学応用上頻繁に現れる重要な問題である。単一制約を持つ非線形ナップザック問題は再帰グリーディ法やスマートグリーディ法を用いて高速にかなり良い近似解を得ることができる[1)2)]。しかし，複数制約条件を持つ問題に一般的なグリーディ法を適用することは困難である。その理由は，グリーディ法は目的関数値と制約関数値の比により選択されるべき項目集合の順序の並べ換えができることを前提としているが，複数制約を持つ問題では制約式が複数あるために，そのような並べ換えを行うことができないからである。そこで，我々は代理制約法を複数制約非線形ナップザック問題（原問題）に用いて変換された単一制約問題（代理問題）にスマートグリーディ法を適用する方法を提案する。

代理制約という考え方は，Glover[3)] により初めて数理計画法の分野に取り入れられた。Luenberger[4)] は，非線形計画問題が準凸問題であるときは，代理乗数を最適化すれば，代理双対問題の解は原問題の解と一致することを示した。しかし，離散変数を含む非線形整数計画問題に代理制約法を適用した場合には，問題の離散性による代理ギャップが存在する。このため代理双対問題の厳密解は原問題の実行可能解になるとは限らないが，この解は原問題の上限値を与える。最適解を得ることが困難である場合には，上限値を求めることは得られた近似解の品質評価をするために重要である。我々は，代理双対問題の解法として代理問題に対して厳密解法であるモジュラアプローチ(MA)[5)]，代理乗数を最適化するために COP（Cut-off Polyhedron）アルゴリズム[6)] を用いた。この最適化された代理乗数を持つ代理問題にスマートグリーディ法を適用する。スマートグリーディ法は，再帰グリーディ法に複数のバランス係数を用いたグリーディ関数を導入し，得られた複数の解の中

から最善の解をスマートグリーディ解とする方法である。この方法は各制約条件について実行可能性のチェックを行い実行可能解を生成する。

本論文で提案した方法で得られたスマートグリーディ解は，品質の良いことを計算機実験によって示す。

2．代理制約法

複数制約非線形ナップザック問題は次のように定式化される。

[K]

$$\text{maximize} \quad f(\boldsymbol{x}) = \sum_{n\in\mathcal{N}} f_n(x_n), \tag{1}$$

$$\text{subject to} \quad g_m(\boldsymbol{x}) = \sum_{n\in\mathcal{N}} g_{mn}(x_n) \le b_m,\ m = 1, 2, \cdots, M, \tag{2}$$

$$x_n \in \mathcal{A}_n, \tag{3}$$

ここで，$\mathcal{N} = \{1, 2, \cdots, n, \cdots, N\}$ は変数の添字集合，$\mathcal{A}_n = \{a_{n1}, a_{n2}, \cdots, a_{nk}, \cdots, a_{nk_n}\}$ は n 番目の変数の項目集合である。

問題 [K] に対する代理双対問題 [SD] は，次となる。

[SD]

$$\min_u \max_x \{f(\boldsymbol{x}) : \varphi(\boldsymbol{u}, \boldsymbol{x}) \le \beta(\boldsymbol{u})\}, \tag{4}$$

ただし

$$\varphi(\boldsymbol{u}, \boldsymbol{x}) = \sum_{m=1}^{M-1} u_m\{g_m(\boldsymbol{x}) - g_M(\boldsymbol{x})\} + g_M(\boldsymbol{x}), \tag{5}$$

$$\beta(\boldsymbol{u}) = \sum_{m=1}^{M-1} u_m(b_m - b_M) + b_M, \tag{6}$$

$$\boldsymbol{u} = (u_1, u_2, \cdots, u_{M-1})^T \in \mathrm{R}^{M-1}, \tag{7}$$

$$\mathrm{U} = \{\boldsymbol{u} \in \mathrm{R}^{M-1} : \sum_{m=1}^{M-1} u_m \le 1, \boldsymbol{u} \ge 0\}, \tag{8}$$

$\varphi(\boldsymbol{u}, \boldsymbol{x})$ を代理制約関数と呼ぶ。また，$\boldsymbol{u}$ は代理乗数と呼ばれる。問題 [SD] の厳密解を $\boldsymbol{x}^{SD}$，そのときの $\boldsymbol{u}$ を $\boldsymbol{u}^{SD}$ とする。解 $\boldsymbol{x}^{SD}$ は問題 [K] の最適解の上限値を与える。

定理１：問題 [K] の実行可能な解を $\boldsymbol{x}^{FEASIBLE}$ とすれば次式が成立する。

$$f(\boldsymbol{x}^{FEASIBLE}) \le f(\boldsymbol{x}^{SD}). \tag{9}$$

証明：問題 [K] の実行可能領域を $\mathcal{F}^K$，問題 [SD] の実行可能領域を $\mathcal{F}^{SD}$ とすれば，

$$\forall \boldsymbol{u} \quad \text{に対して} \quad \mathcal{F}^K \subset \mathcal{F}^{SD} \tag{10}$$

が成立する。

定理1より $\boldsymbol{x}^{SD}$ が実行可能である時，$\boldsymbol{x}^{OPT}$ を問題[K]の厳密解とすると，

$$\boldsymbol{x}^{OPT} = \boldsymbol{x}^{SD} \tag{11}$$

が成立する。

3．代理双対問題を解くためのアルゴリズム

代理双対問題[SD]を解くためにアルゴリズムMAおよびCOPを用いる。代理乗数 $\boldsymbol{u}$ の形成する多面体の重心として与えられた代理乗数を持つ代理問題にMAを用いて厳密解を得る。次に得られた厳密解を用いてCOPは多面体を切断縮小する。代理乗数の定義域を初期多面体としてこれらの手順を繰り返して代理乗数を更新し，多面体が空集合となった時に停止する。最終的に得られた代理乗数 $\boldsymbol{u}^{SD}$ は代理問題の目的関数値を最小にするという意味で最適化されている。

3．1　モジュラアプローチ

MAは，与えられた離散最適化問題に対して

(1)　各変数に対して深測操作（fathoming）を行い各変数に対する項目集合 $\mathcal{A}_n$ の要素数を減らす。

(2)　複数の変数を統合して1つの変数にし，変数の数を減らす。

という2つの操作を繰り返し，最終的にただ1つの変数に統合して最適解を求める方法である。この方法は問題を厳密に解くので厳密解法である。上記の(1)の深測操作では，Branch&Boundの手法である，実行可能性操作，限界値操作，優越性操作を用いている。(2)で変数を統合するというのは次のように変数の項目集合 $\mathcal{A}_n$ の直積をとることである。

$$\mathcal{A}_{NEW} = \mathcal{A}_i \times \mathcal{A}_j, (i, j, \in \mathcal{N}). \tag{12}$$

変数の統合にはいくつかの方策が考えられるが，我々は $\mathcal{A}_i$，$\mathcal{A}_j$ の選択方策として，その集合の項目数が最大のものと最少のものを選んで統合した。この方策によればメモリ消費量が少なくしかも高速に解けることが報告されている[7]。

3．2　COPアルゴリズム

代理乗数の定義域のつくる多面体 U^i の重心として与えられた代理乗数 $\boldsymbol{u}^i$ に対する代理問題の厳密解を $\boldsymbol{x}^{i+1}$ とする。このとき，多面体 U^i を切断して得られる新しい多面体は

$$\mathrm{U}^{i+1} = \mathrm{U}^i \cap \{\boldsymbol{u} : \varphi(\boldsymbol{u}, \boldsymbol{x}^{i+1}) < \beta(\boldsymbol{u})\} \tag{13}$$

で与えられる。(13)式に基づいて，COPは $\mathrm{U}^{i+1} = \phi$ となるまで多面体を順次切断縮小して，最適な代理乗数 $\boldsymbol{u}^{SD}$ を求めることができる。

3．3 スマートグリーディ法

複数制約非線形ナップザック問題に対する実行可能な近似解を得るために，MA および COP によって最終的に得られた最適な代理乗数を用いた代理問題に対してスマートグリーディ法を適用する。スマートグリーディ法は，整数優越性および LP 優越性の原理を用いて最適解の候補として不要な項目を各変数の項目集合から除外し，隣り合う項目集合に対する目的関数と制約関数との増分の比（ゲイン率）が単調減少となるように項目集合を並べ変える。そして，縮小され並べ換えられた項目集合

$$\mathcal{A}_n = \{a_{n1}, \cdots, a_{nk_n}\}, \qquad (k_n \leq K_n,\ n = 1, \cdots, N), \tag{14}$$

に対してバランス係数 α_l $(0 \leq \alpha_l < \infty)$ によって指定されたグリーディ関数

$$\psi_n(a_{nk}, a_{nk+1}) = \Delta f_n \times \left(\frac{1-\alpha_l}{\Delta M} + \frac{\alpha_l}{\Delta g_n^S}\right),\ (l = 1, 2, \cdots, L) \tag{15}$$

ただし

$$\Delta f_n = f_n(a_{nk+1}) - f_n(a_{nk}), \tag{16}$$

$$\Delta g_n^S = g_n^S(a_{nk+1}) - g_n^S(a_{nk}), \tag{17}$$

$$g_n^S(a_{nk}) = \sum_{m=1}^{M-1} u_m^{SD}\{g_{mn}(a_{nk}) - g_{Mn}(a_{nk})\} + g_{Mn}(a_{nk}), \tag{18}$$

$$\Delta M = \frac{1}{N^+}\sum_{n \in \mathcal{N}^+} \frac{g_n^S(a_{nk_n}) - g_n^S(a_{n1})}{k_n - 1}, \tag{19}$$

を最大とする変数に対して増分配分を行なうことを繰り返す。ただし，ここで a_{nk} は暫定解の n 番目の変数を構成する項目，a_{nk+1} はその変数の増分配分の候補となる項目である。Δf_n および Δg_n^S はそれぞれ目的関数および代理制約関数の増分を表す。また，ΔM は代理制約関数 g_n^S の項目集合全体についての増分の平均を表す。ここで N^+ は増分配分が可能な変数の添字集合 $^+$ の項目数を表す。スマートグリーディ法は複数のバランス係数 $\alpha_l (0 \leq \alpha_l < \infty, l = 1, 2, \cdots, L)$ を用いて複数のグリーディ解を生成してその中から最善の解をスマートグリーディ解とする。この方法は各制約条件について解の実行可能性のチェックを行なって解が実行可能になる時増分配分を停止するので得られた解は原問題の実行可能解となり，その解は原問題の下限値を与える。

4．計算機実験

本実験に用いた計算機は NEWS-5000VI（CPU R4000SC）である。本実験では目的関数 f_n，制約関数 g_{mn} が整数値をとるように問題を乱数を用いて発生した。制約数 $M = 3$，問題の規模は変数の数 $N = 10, 20, 50, 100$，変数に対する構成項目数 $K_n = 10, 20, 50, 100$ について，10問ずつ解いた結果を表1に表す。ここで，上限値計算は MA，COP によって

表1 計算機実験結果

$K_n =$		10	20	50	100
$N = 10$	(a)	<1s	<1s	1s	2s
	(b)	0.5	1	1.7	1.7
$N = 20$	(a)	<1s	<1s	2s	6s
	(b)	0.7	1	1.6	1.9
$N = 50$	(a)	1s	3s	11s	51s
	(b)	1.2	1.9	2.5	2.4
$N = 100$	(a)	5s	14s	1m08s	3m16s
	(b)	4.1	3.7	6.2	4

(a)：上限値計算に要した平均時間
(b)：上限値と下限値の差の平均

上限値を求める計算，下限値計算は本論文で提案したスマートグリーディ法によってスマートグリーディ解を求める計算を表す。表1に，各々の問題に対して上限値計算に要した時間の平均と，上限値と下限値との差の平均を表す。下限値計算に要した時間はすべて1秒以内で，高速にスマートグリーディ解が得られている。この表において，問題の規模が大きくなると上限値と下限値との差が大きくなっているが，これは問題の規模の増大に伴い目的関数値も大きくなるためであり，上限値と下限値との相対誤差で評価すれば小さくなっており解の品質は十分良いことが判る。

5．むすび

本論文では，複数制約非線形ナップザック問題にスマートグリーディ法を適用する方法を提案した。この方法は，厳密解法であるMAと代理乗数を最適化するCOPとを用いて最終的に得られた代理乗数を持つ代理問題にスマートグリーディ法を適用するという方法である。得られたスマートグリーディ解は元の複数制約非線形ナップザック問題の品質の良い実行可能解であり，この解が高速に得られることが計算機実験により確かめられた。

参考文献

1) Ohtagaki H., Nakagawa Y.："A Recursive Greedy Procedure for Solving Nonlinear Knapsack Class of Reliability Optimization Problems", Systems and Control, Topics in Theory and Applications, Mita Press, pp.293-305 (1991).
2) Ohtagaki H., Nakagawa Y., Takatsuka H. and Narihisa H.："Smart Greedy Procedure for Solving a Class of Nonlinear Knapsack Problems", Proc. of the Austr-Jpn Workshop on Stochas. Models in Engng., Tech. and Manag., World Scientific, pp.454-462 (1993).
3) Glover F.："A multiphase-dual algorithm for the zero-one integer programming problem", Oper. Res., **13**, pp.879-919 (1965).
4) Luenberger D. G.："Quasi-convex programming", SIAM J. of Appl. Math., **16**, pp.1090-1095

(1968).
5) 仲川勇二："離散最適化問題の新解法"，信学論 J-73A，pp.550-556 (1990).
6) 仲川勇二，疋田光伯，鎌田　弘："代理双対問題を解くためのアルゴリズム"，信学論 Vol. J67-A，pp. 53-59 (1984).
7) 仲川勇二，疋田光伯，岩崎彰典："多重選択ナップザック問題の高速厳密解法"，信学論 J-75A，pp.1752-1754 (1992).

An Application of Smart Greedy Procedure to Multi-dimensional Nonlinear Knapsack Problems

Tetsuo KAMETAKA, Hirokazu OHTAGAKI*, Akinori IWASAKI**
Yuji NAKAGAWA*** and Hiroyuki NARIHISA****

Graduate School of Engineering,
**Department of Electronic Engineering, Faculty of Engineering,*
***Information Processing Center,*
*****Derartment of Information and Computer Engineering,*
Faculty of Engineering,
Okayama University of Science
1-1 Ridai-cho, Okayama 700, Japan

****Faculty of Informatics, Kansai University*
Takatsuki-shi, 569 Japan

(Received September 30, 1995)

Smart Greedy procedure is proposed for solving multi-dimensional nonlinear knapsack problems.

Introducing a surrogate multiplier, the multi-dimensional nonlinear knapsack problem can be translated to a surrogate problem, which is an one-dimensional nonlinear knapsack problem. With the surrogate multiplier that is a centroid of the polyhedron of surrogate multiplier, the algorithm MA solves the surrogate problem, and then the algorithm COP optimizes the surrogate multiplier by reducing the polyhedron, repeatedly, untill the polyhedron becomes empty. Smart Greedy generates the Smart Greedy solutions of surrogate problems with the optimal surrogate multiplier. The solutions obtained are feasible to original multi-dimensional nonlinear knapsack problems.

The computational experiments show that the proposed method provides high quality solutions.