2001年度日本オペレーションズ・リサーチ学会
春季研究発表会

排他制約付きナップサック問題の近似解法と厳密解法*

01700900 防衛大学校情報工学科 山田 武夫[†] YAMADA Takeo
01107880 防衛大学校情報工学科 片岡 靖詞 KATAOKA Seiji

1 はじめに

ナップサック問題（knapsack problem: KP）は OR や情報工学分野の基本問題の一つとして, 数多くの研究がある [1] が, 本稿では商品間に相性の善し悪しがある場合を考察する. 一般の KP の場合と同様に, n 個の商品があり, その価値と重量が p_j, w_j であるとする. これらの商品を容量 C のナップサック（以下, KS と略記）に入れるのであるが, 商品の組のいくつかは相性が悪く, 同時に KS に入れることが出来ない. このような商品の組を $E = \{(i,j)|1 \leq i \neq j \leq n\}$ とすると, 問題は次の排他制約付き **KP**（disjunctively constrained KP: DCKP）のように定式化される.

$$\text{Maximize} \quad z(x) := \sum_{j=1}^{n} p_j x_j \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \leq C, \tag{2}$$

$$x_i + x_j \leq 1, \quad \forall (i,j) \in E \tag{3}$$

$$x_j \in \{0,1\}, \quad \forall j \tag{4}$$

ここで, x_j は商品 j を採択するか否かを表す決定変数である.

本稿では, 以下の仮定の下に DCKP の近似解法と厳密解法を検討する.

A$_1$. p_j, w_j, C はすべて正整数.
A$_2$. $\sum_{j=1}^{n} w_j > C$, $\exists w_j \leq C$.
A$_3$. 商品は相対価値の降順に整列済み. すなわち

$$p_1/w_1 \geq p_2/w_2 \geq \ldots \geq p_n/w_n. \tag{5}$$

2 近似解法

DCKP の近似解法としては, 短時間に実行可能解を得るグリーディ法と, 解をさらに改善する **2-opt** 法を考える.

2.1 グリーディ解

これは, 以下を $i = 1, 2, \ldots, n$ の順に実行する.

(a) 商品 i は KS に収容可能か？
(b) KS 中に i と相性の悪い商品が含まれていないか？
⇒ (a), (b) とも Yes なら, 商品 i を KS に入れる.

2.2 2-opt 解

KS 内の商品 i と KS 外の j のあらゆる組合せについて, 以下を可能な限り反復する.

(c) 商品 i, j の入れ換えは可能か？
(d) 上の入れ換えにより目的関数値は増加するか？
⇒ (c), (d) とも Yes なら $i \leftrightarrow j$ の入れ換えを行う.

2.3 例題

商品数 $n = 200$, p_j, w_j は $[1,100]$ でランダム, $C = 1000$ とし, 排他制約は全商品の組から 2%の比率で, 合計 407 本発生させた. その結果, グリーディ解, 2-opt 解の目的関数値はそれぞれ $\underline{z}_1 = 2871$, $\underline{z}_2 = 2879$, 計算時間は HP9000 B132L で 0.00 秒, 0.01 秒であった.

3 上界値

DCKP の上界値としては, 例えば次のものが考えられる.

- (4) を連続緩和する.
- (4) を連続緩和, (3) をラグランジュ緩和する.
- (4) を連続緩和し, (3) を落とす.

これらにより得られる上界値をそれぞれ $\bar{z}_1, \bar{z}_L, \bar{z}_2$ とする.

ラグランジュ緩和問題は

$$\text{Maximize} \quad L := \sum_{j=1}^{n} (p_j - \sum_{e \in \partial j} \lambda_e) x_j + \sum_{e \in E} \lambda_e \tag{6}$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \leq C \tag{7}$$

$$0 \leq x_j \leq 1, \quad \forall j \tag{8}$$

となるが, これは容易に解くことが出来て, 1つの上界値を与える. $e = (i,j) \in E$ とすると（微分可能のとき）

$$\partial L / \partial \lambda_e = 1 - x_i - x_j \tag{9}$$

*法政大学, 2001.5.1-2
[†]E-mail: yamada@nda.ac.jp

であることに注意して, 劣勾配法によりラグランジュ乗数を調整すると, 比較的短時間に良質の上界値 $\bar{z}_L$ を得ることが出来る. これらの上界値の間には次の関係が成り立つ.

命題 1.

$$\bar{z}_1 \leq \bar{z}_L \leq \bar{z}_2 \tag{10}$$

証明： 左側の不等式は明らか. ラグランジュ乗数をすべて 0 としたものが $\bar{z}_2$ なので右側の不等式もいえる. ▮

例 3.1 前節の例題については, $\bar{z}_1 = 2924$, $\bar{z}_2 = 3271$, ラグランジュ緩和は（34 回の反復で）$\bar{z}_L = 2948$ で, それぞれの計算時間は 1.98, 0.01, 0.00（秒）であった.

4 厳密解法

4.1 全列挙法と陰伏列挙法

$1, 2, \ldots, n$ の順に商品 i までの採否が決定された状態を$< x_1 x_2 \ldots x_i >$と記すと, これらの全体は 2 分木をなす. これをすべて列挙すればDCKP の解が得られるが, これは計算量が $O(2^n)$ で実用的でない.

状態を全て列挙するのではなく, それらのいくつかをうまく枝刈りして計算効率を高めようというのが陰伏列挙法であるが, ここでは以下のような枝刈り条件が考えられる.

C$_1$. $< x_1 x_2 \ldots x_i >$が KS の容量をオーバーしている場合.

C$_2$. $< x_1 x_2 \ldots x_i >$が相性の悪い商品の組を含む場合.

C$_3$. 商品 $i+1$ 以降から, 現在手持ちの解以上の総利益が見込めない場合.

C$_1$, C$_2$はほとんど自明であるので, C$_3$について補足する. 今, DCKP の 1 つの実行可能解 x が手許にあって, その利益を $z := z(x)$ とする. すなわち, z は 1 つの下界値である. 商品の KS への採否が状態$< x_1 x_2 \ldots x_i >$まで進んだものとして, この条件下で見込まれる利得の条件付き上界値を$\bar{z}(x_1, \ldots, x_i)$ とすると,

$$\bar{z}(x_1, \ldots, x_i) \leq z \tag{11}$$

であればこの状態を見切ることが出来る.

実際に$\bar{z}(x_1, \ldots, x_i)$ を求めるには, 前節の上界値計算法を若干修正すればよいが, 実装したプログラムでは i が 10 の倍数の場合のみラグランジュ緩和を行い, それ以外では計算時間が少なくてすむ $\bar{z}_2$ を用いた.

例 4.1 2.3 節の例では, $z = 2879$（2-opt 解）とすると, 43,373 個の状態が生成され, 84.54 秒で最適解 $z^* = 2917$ を得た.

4.2 区間縮小法

前節では, z を DCKP の下界値としたが, この値は大きいほど生成される状態数が少なくなり, 計算時間も短くてすむ. しかし, 最適値 z^* より大きい z を指定すると, 陰伏列挙アルゴリズムは計算が進行するにつれすべての状態を見切ってしまい, 最適解を見出すこと無く終了してしまう. そこで, 次のような区間縮小法を提案する.

1. 適当な下界値 $\underline{z}$ と上界値 $\bar{z}$ をとる.
2. $z := \alpha \underline{z} + (1 - \alpha)\bar{z}$.
3. z を用いて陰伏列挙法を実行する. 最適解が得られればそれを表示して終了.
4. $\bar{z} := z$として, ステップ 2 へ戻る.

例 4.2 2.3 節の例で, $\underline{z} = 2879$（2-opt 解）, $\bar{z} = 2948$（ラグランジュ緩和）から出発し, $\alpha = 0.7$ とすると, $z = 2927$ となるが, ここでは解が見つからず, 改めて $z = 2912$ として例 4.1 と同じ最適解を得た. 全体としての生成状態数は 3102, 計算時間は 2.95 秒であった. ちなみに, 同じ問題を整数計画法ソルバー [2] で解いた時の計算時間は 166.7 秒であった.

5 数値実験

表 1 に $C = 1000$ の場合の実験結果を示す. 各行は 200 ～ 1000 の各 n についてランダムに発生した 10 例題の平均値で, m は排他制約数, #states は生成状態数である. 表より, $n = 1000$ までの問題が数十秒以内で解けている.

表 1. 実験結果

n	m	$\bar{z}_2$	$\bar{z}_L$	z^*	#states	CPU
200	21.1	3488.5	3505.4	3502.6	2463.0	1.75
400	80.6	4946.0	4958.7	4956.0	2173.4	1.35
600	180.6	6150.5	6157.4	6154.9	3083.8	11.38
800	321.2	6997.0	7011.8	7007.6	9481.7	49.60
1000	502.1	7788.1	7805.3	7802.8	14008.7	69.72

参考文献

[1] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley & Sons, 1990.

[2] 茨木俊秀, 福島雅夫, FORTRAN77 最適化プログラミング, 岩波書店, 1991.