# Maximin 型の目的函数を持つナップサック問題について

飯 田 浩 志

## 概　　要

本稿では，maximin 型の目的函数を持つ，二つのナップサック問題を概観する．一つはナップサック配分問題，もうひとつは max-min 0-1ナップサック問題である．後者は前者の拡張であり，後者の前者からの具体的な構成法にも言及する．

## Abstract

In this paper we give an overview of two knapsack problems with maximin type objective function, one the knapsack sharing problem and the other the max-min 0-1 knapsack problem. The latter is an extension of the former, and we also give a concrete construction of the latter with the former.

**キーワード**: 組合せ最適化 (combinatorial optimization); 0-1ナップサック問題 (0-1 knapsack problem); ナップサック配分問題 (knapsack sharing problem); max-min 0-1ナップサック問題 (max-min 0-1 knapsack problem)

## 1　はじめに

古典的な組合せ最適化問題である0-1ナップサック問題（以下，簡単の為KPという）は，1970年代から盛んに研究されてきた．KPは，唯一の制約条件しか持たない整数計画問題の形をしており，一見容易に解き得るように思えるが，実はNP困難な問題であることは良く知られている．さて，KPでは，価値と重量なる二つの属性を持ついくつかの項（品物）と，それらを運ぶ為のナップサックが与えられる．但し，ナップサックには重量制限があり，すべての項を運べる訳ではない．この制約の下，その価値の総和を最大にする項の組合せを見つけることが，KPの目的である．KPは，次のように定式化される：

$$\begin{array}{lll} \text{(KP)} & \text{最 大 化} & \displaystyle\sum_{j\in N} p_j x_j \\ & \text{制約条件} & \displaystyle\sum_{j\in N} w_j x_j \le c \\ & & x_j \in \{0,1\},\ j \in N. \end{array}$$

本稿を通じて，$N := \{1, 2, \ldots, n\}$であり，$n$が項数を示すものとする．$p_j$, $w_j$をそれぞれ，項$j\,(\in N)$の価値，重量と呼ぶ．$c$はナップサックの重量制限を示し，0-1変数$x_j$が，項$j$の選択$(x_j=1)$/非選択$(x_j=0)$を示す．KPでは，選択可能な項のみを対象とする為に，すべての項$j$の重量について$w_j \le c$を，先に述べたように，問題自体を意味あるものにする為に$\sum_{j\in N} w_j > c$を前提とするのが通例である．KPの詳細については，例えば，Martello and Toth [5]を参照されたい．

このKPに拡張を施した問題は，数多く存在する．その特殊な場合としてKPを含むが故に，KPを拡張した問題はすべてNP困難であり，一般に容易には解き得ない．本稿では，KPの拡張の中でも特に，特徴的なmaximin型の目的関数（ある集合の中の最小値を最大化する）を持つ二つのナップサック問題に焦点を当て，これまでに公刊された文献をもとに，それらを概観する．一つはナップサック配分問題，もうひとつはmax-min 0-1ナップサック問題で

ある．後者は前者の拡張であり，後者の前者からの具体的な構成法にも言及する．

## 2 ナップサック配分問題

本節では，ナップサック配分問題（Knapsack Sharing Problem，以下 KSP という）を紹介する．KSP は，次の式で与えられる：

$$\text{(KSP)} \quad \text{最 大 化} \quad \min_{1\leq k\leq r} \sum_{j\in N_k} p_j x_j \tag{1}$$

$$\text{制約条件} \quad \sum_{j\in N} w_j x_j \leq c \tag{2}$$

$$x_j \in \{0,1\},\ j \in N \tag{3}$$

$$\bigcup_{k=1}^{r} N_k = N, \quad N_k \cap N_l = \emptyset,\ k \neq l. \tag{4}$$

KSP では，KP の定義に加えて，(4)：項の集合 $N$ を，$r$ ヶの排他的なクラスに分割する．その上で，(1)：選択した項の価値の和をクラスごとに算出し，その中の最小値を最大化する，という点が KP の拡張となっている．実際，$r=1$，i.e. クラスが唯一の時，KSP は KP に一致する．ここでは簡単の為，$c$ 及びすべての $p_j$，$w_j$ は正の整数と仮定する．

**例 1**．500円玉一つ持って，学食に行くとする（$c=500$）．各品目 $j$ は，赤，緑，黄の食品群に分けられており，その指数は $p_j$ であるとする．但し，**各品目は唯一つの群にのみ属する**とする．また，各品目には値段 $w_j$ 円も与えられている．ここで，500円以内で選択した品目から算出した三つの指数それぞれの合計の内，最小のものを最大にするのが KSP である．大雑把に言えば，どれか一つの群の指数の合計のみが突出しても最適解にはなり得ない．バランス良く品目を選び，赤，緑，黄全体を底上げするのが肝要である．

**Quiz.** KSP (1)−(4)で，$\sum_{k=1}^{r} \min_{j\in N_k} w_j > c$ の時，最適値はいくらになるか?

一般に，分枝限定法を適用するにあたって，線形緩和問題（連続緩和問題ともいう）を解くことの重要性は，論をまたないだろう．KSP の線形緩和問題，即ち，0-1条件 (3) を $0\le x_j \le 1$ に緩めた問題は，如何にして解けばよいのであろうか．以下では，Yamada et al [8]に沿って，KSP の線形緩和問題への解法を概説する．以降，KSP の線形緩和問題を C(KSP)，その最適値を $\bar{z}$ で示す．

まず，各クラス $k$ について次の問題を定義する：

$$\begin{aligned} &\text{最大化} && \sum_{j\in N_k} p_j x_j \\ &\text{制約条件} && \sum_{j\in N_k} w_j x_j \le c^k \\ & && 0\le x_j \le 1,\ \ j\in N_k. \end{aligned} \tag{5}$$

問題 (5) の最適値を，重量制限 $c^k$ に依存するという意味で，$\bar{z}^k(c^k)$ と書く．以下，勝手なクラス $k$ に属する二つの項 $i, j\in N_k$ について，$i<j$ ならば $p_i/w_i \ge p_j/w_j$ が成立しているものと仮定する．この仮定の下では，問題 (5) の最適値，即ち，良く知られた Dantzig の上界（Dantzig [1]）は，即座に求まる事に注意されたい．

今，$\bar{p} := \min_{1\le k\le r} \sum_{j\in N_k} p_j$ と置く．容易に分かるように，KSP の最適値は，この $\bar{p}$ を越えない．ここで，各クラス $k$ について，$\bar{z}^k(\cdot)=\bar{p}$ を満たす重量 $\bar{c}^k(\bar{p})$ を考えた時，即ち，$\bar{z}^k(\bar{c}^k(\bar{p})) = \bar{p}\ (1\le k\le r)$ とした時：$\sum_{k=1}^{r} \bar{c}^k(\bar{p}) \le c$ であるなら，$\bar{p}$ が C(KSP) の最適値となる；$\sum_{k=1}^{r} \bar{c}^k(\bar{p}) > c$ の時，明らかに $\bar{z} < \bar{p}$ であり，また，最適値 $\bar{z}$ を与える各クラスへの重量配分 $(\bar{c}^1, \bar{c}^2, \dots, \bar{c}^r)$ について，次が成立する．

$$\bar{z}^k(\bar{c}^k) = \bar{z},\quad k=1, 2, \dots, r \tag{6}$$

$$\sum_{k=1}^{r} \bar{c}^k = c \tag{7}$$

直観的に言えば，(6): あるクラスで価値の総和が $\bar{z}$ より大きいならば，その差を与える重量分だけ，すべてのクラスに配分し直して底上げを行なうことで，また，(7): 各クラスに配分し終った後の重量に余りがあれば，その分だけ再配分し，結局どちらの場合に於いても，最適値を $\bar{z}$ より大きくできてしまう．この (6)-(7) を満たす $\bar{z}$ を線形時間で求める解法については，Kuno et al [4] にある．以上で，KSP の線形緩和問題の最適値が求まる．KSP へ分枝限定法を適用するにあたっての詳細，即ち，初期暫定値(解)の求め方，分枝変数の選択等は，[8]を参照されたい．

さて，従来の分枝限定法と線形緩和問題による枠組以外に，KSP の最適値を求める術はないものだろうか．本節の残りでは，[8]の後半で提案された効率的な KSP の最適値の求め方を，少し詳しく解説する．

以降, KSP の最適値を $z^*$ で示す. まずはじめに, 任意の整数 $z \geq 0$ について, $z^* \geq z$ と

$$\begin{aligned} &\sum_{j \in N_k} p_j x_j \geq z, \quad k = 1, 2, \ldots, r \\ &\sum_{j \in N} w_j x_j \leq c \qquad\qquad\qquad\qquad\qquad (8) \\ &x_j \in \{0,1\}, \ j \in N \end{aligned}$$

が実行可能解を持つ (feasible) ことは同値である事に注意する．何となれば: $z^* \geq z$ ならば，$z^*$の最適性から，(8) は少なくとも一つの実行可能解，即ち，$z^*$を与える解を持つ; 逆に，(8) が実行可能解を持ち且つ $z^* < z$ ならば，$z^*$の最適性に反する. 以降, (8) が実行可能解を持つ時の $z$ を, 到達可能 (attainable) と呼ぶ．明らかに，到達可能な $z$ の最大値が $z^*$を与える．以下，任意の整数 $z \geq 0$ の到達可能性を判定する方法を示す．

まず，各クラス$k$ごとに次の問題を定義する：

$$\begin{aligned}&\text{最 小 化} && \sum_{j\in N_k} w_j x_j \\ &\text{制約条件} && \sum_{j\in N_k} p_j x_j \geq z \\ & && x_j \in \{0,1\},\ j\in N_k.\end{aligned} \tag{9}$$

この問題(9)の最適値を$c^{*k}(z)$と書く．換言すれば，クラス$k$について，価値の和を$z$以上に保つ為の最小の重量和が$c^{*k}(z)$である．ここに，$z$の到達可能性は次と同値である．

$$\sum_{k=1}^{r} c^{*k}(z) \leq c$$

つまり，$c$を各クラスに，その価値の和を$z$以上に保つ為の最小の重量和分だけ配分可能であれば，$z$は到達可能である．また，(9)で$y_j := 1-x_j$とおけば，(9)は

$$\begin{aligned}&\text{最 大 化} && \sum_{j\in N_k} w_j y_j - \sum_{j\in N_k} w_j \\ &\text{制約条件} && \sum_{j\in N_k} p_j y_j \leq \sum_{j\in N_k} p_j - z \\ & && y_j \in \{0,1\},\ j\in N_k\end{aligned}$$

なるKPになるので，(9)の上/下界は苦労なく求まる（例えば，Martello and Toth [6]の第2節参照）．ここで，各クラス$k$それぞれについて求めた(9)の上界$\bar{c}^{k}(z)$，下界$\underline{c}^{k}(z)$について：$\sum_{k=1}^{r}\bar{c}^{k}(z)\leq c$であれば$z$は到達可能であり，$\sum_{k=1}^{r}\underline{c}^{k}(z) > c$であれば到達不可能であると言える；どちらでもない場合，残念ながら各クラスそれぞれについて，与えられた$z$を入れた(9)をexactに解かねばならない．尤も，$\sum_{k=1}^{r'} c^{*k}(z) > c\ (r' < r)$になった時点で打ち切って構わない．以上により，任意の整数$z\geq 0$の到達可能性が判定可能であることが示された．

さて，到達可能/到達不可能な二つの組$\{z_L,\ z_R\}$ $(z_L+1<z_R)$を考える．ま

ず，C(KSP)の最適値 $\bar{z}$ について，$\lceil \bar{z} \rceil$は到達不可能であるとし，$z_R$ の初期値とする[1]．次に，何らかの形で到達可能な $z_L$ を得たとする．この時，任意の整数 $z \geq 0$ の到達可能性は上記の事から判定可能であるので，**二分探索**によって KSP の最適値を求める事が出来る．$z_L$ の初期値[2] 等，詳しくは[8]を参照されたい．

## 3　Max-min 0-1ナップサック問題

本節では，max-min 0-1 ナップサック問題（max-min 0-1 knapsack problem，以下 MNK という）を紹介する．KP では項の価値 $p_j$ は恒久的であったが，未来に対して複数の局面（シナリオ）を設定し，項それぞれの各局面下で異なる価値を考慮するのが MNK である．MNK は，Yu [9]によって提案された．[9]では，分枝限定法＋最良優先探索による解法も，合わせて提案されている．MNK は，次の式で与えられる：

$$\text{(MNK)}\quad \text{最大化}\quad \min_{s \in S} \sum_{j \in N} v_j^s x_j \tag{10}$$

$$\text{制約条件}\quad \sum_{j \in N} w_j x_j \leq c \tag{11}$$

$$x_j \in \{0,1\},\ j \in N, \tag{12}$$

ここに，$S$ はシナリオの集合である．各項 $j \in N$ は重量 $w_j$ を持ち，シナリオ $s \in S$ の下で価値 $v_j^s$ を持つ．加えて，すべての $v_j^s$ 及び $w_j$ は正であるとして一般性を失わない．また，$|S| = 1$, i.e. シナリオが唯一つの時，MNK は KP に

[1] **注**：[8]では，$\bar{z}$ を $z_R$ の初期値としているが，$z_R - z_L = 1$ をループの終端条件としている事等から，$z_R$ の整数性を仮定しているはずである．よってここでは，$\bar{z}$ ではなく $\lceil \bar{z} \rceil$ とした．$\lceil \bar{z} \rceil = \bar{z}$ の時，$\lceil \bar{z} \rceil$は到達可能となる可能性のある事に注意されたい．また[8]で，手続き *Binary-Search* のループ処理の前に，$z_R - z_L \leq 1$ を見るべきであろう．

[2] [8]に従って $z_L$ の初期値を計算するのと，$z_L := 0$ として二分探索をいきなり始めるのと，実際にはどちらが速いだろうか？

一致する．目的函数(10)の意味する所は，すべてのシナリオでそれぞれ計算した中での最小の価値の総和を最大にする項の選択を求めよ，ということである．大雑把に言えば，KSP と同様，あるシナリオに於ける価値の総和が突出しても，最適解にはなり得ない．

**例 2**．項数 5，二つのシナリオからなる簡単な例を掲げておく．各項 $j \in \{1, 2, \ldots, 5\}$ につき，重量 $w_j$ 及びシナリオ $s \in \{1, 2\}$ の下での価値 $v_j^s$ は次のように与えられるとする：

| $j$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $w_j$ | 3 | 37 | 84 | 30 | 99 |
| $v_j^1$ | 16 | 22 | 12 | 73 | 106 |
| $v_j^2$ | 39 | 101 | 4 | 34 | 94 |

加えて，重量制限 $c = 126$ とする．例えば，選択 $\{1, 3, 4\}$ はナップサックに入り，$v_1^1 + v_3^1 + v_4^1 = 101 > 77 = v_1^2 + v_3^2 + v_4^2$ より，その最小の価値の総和は 77 である．選択 $\{1, 2, 3\}$ は，最小の価値の総和が 50 であるので，なお悪い．この例では最適値は 122 であり，それを与える最適解は $\{1, 5\}$ である．

先の例 1 で，一つの品目が唯一つの食品群（赤，緑，黄）にのみ属するというのは，現実にそぐわない．各品目について，赤，緑，黄三つの群それぞれに指数を持たせることがより好ましいが，これは MNK を用いれば定式化できる．つまり，赤，緑，黄それぞれの群を一つのシナリオとすれば良い．

さて，前節でも言及したように，一般にナップサック問題に分枝限定法を適用するにあたっては，元の問題及びその部分問題（分岐の過程で現れる，ある項の組の選択/非選択を決定する事により，対象とする項数を少なくした問題）の線形緩和問題を解いて上界を求めるのが常道であるが，MNK の線形緩和問

題の最適値は, KP に対する Dantzig の上界のように容易に求まりそうもない. 実は, [9]では, 線形緩和ではなくsurrogate 緩和が用いられている. MNK の surrogate 緩和問題は, [9]で示されたように, KP として書ける:

$$
\begin{aligned}
& z_U(\mu) = \max_x \sum_{j\in N} v_j(\mu)\, x_j\,, \quad v_j(\mu) = \sum_{s\in S} \mu_s v_j^s \\
& \text{制約条件} \qquad \sum_{j\in N} w_j x_j \le c \\
& \qquad\qquad\qquad \mu_s \ge 0\ (s\in S),\ \sum_{s\in S} \mu_s = 1 \qquad (13) \\
& \qquad\qquad\qquad x_j \in \{0,1\},\ j\in N.
\end{aligned}
$$

緩和問題(13)に関しては, 如何にして $z_U(\mu)$ を小さくするように, 乗数ベクトル $(\mu_s)_{s\in S}$ を決めるかが本質である. [9]では, 劣勾配 (subgradient) 法を用いて, その繰り返し処理の過程で乗数ベクトルを徐々に強化していく手続きが提案されている. その手続きから出力されるMNKの上/下界は非常に tight であり, 特筆に値する. また, 乗数ベクトル $(\mu_s)_{s\in S}$ について $\sum_{s\in S} \mu_s = 1$ を仮定すれば, ラグランジュ緩和からも(13)と同一の式を導けることを付言しておく (Iida[2], p.5 参照).

加えて, MNK への greedy 法の適用に関する記述が, [2]にある. KP に対しては, greedy 法はかなり良い下界を与えることが知られているが, MNK に対して良好な結果は得られていない([2]の Table 2 と[9]の Table III 中の $z_L$ を比較されたい). これは, KP の $p_j/w_j$ に相当するような, MNK に於ける項の重要度 − efficiency − を示す尺度を与え得ない事に起因すると思われる. さらに, (13)の乗数ベクトルの決め方に関する一考察が, Iida [3]にある. 残念ながら, [9]で提案されたそれら程 tight な上/下界を得るまでには至っていない.

話かわって, これは[9]にも少し言及されていることだが, 例 1, 2 からも分かるように, MNK は KSP の拡張と考える事が出来る. 以下では, KSPから MNK を具体的に構成してみよう. KSP (1)−(4) から MNK (10)−(12) を次のように構成する:

$$S = \{1, 2, \ldots, r\}$$

$$v_j^s = \begin{cases} p_j & j \in N_s \\ 0 & j \notin N_s. \end{cases}$$

即ち，KSP の一クラスを MNK の一シナリオに対応させるのである．あるクラス $s$ に属する項 $j$ ($\in N_s$) は，対応するシナリオ $s$ でのみ価値 $p_j$ を持つとする．この時，KSP の目的函数は

$$\min_{1 \leq s \leq r} \sum_{j \in N_S} p_j x_j = \min_{s \in S} \left\{ \sum_{j \in N_S} v_j^s x_j + \sum_{j \notin N_S} v_j^s x_j \right\} = \min_{s \in S} \sum_{j \in N} v_j^s x_j$$

であり，確かにMNKの目的函数として書ける．以上の事から，KSP を MNK の枠組で解く事が可能となる．しかし，KSP は既に高速に解ける問題であり，MNK として解くメリットは無いであろう．

以上，KSP から MNK への変換を与えたが，もしこの逆変換を与えることが出来れば，MNK を解くのに有望な枠組となるであろう事は想像に難くない．MNK から KSP への変換，即ち，情報量のより少ない方への変換など存在しないと決めつけるのは早計である．実際，KP の拡張である collapsing knapsack problem (CKP) から KP への変換を定義し，KP の枠組で CKP を解いた例もある (Pferschy et al [7])．

## 4 おわりに

本稿では，0-1ナップサック問題の数ある拡張の中でも特に，特徴的な maximin 型の目的函数を持つ二つのナップサック問題を概観した．一方の KSP は非常に良く研究されており，$n=10000$，$r=10$ 程度のかなり大規模な問題でも，実用時間以内に解ける事が示されている．他方，MNK は $n=60$，$|S|=30$ までの実績しかなく，より大規模な問題をより速く解ける可能性を秘めているものと思われる．とは言え，本文中で述べたように，この問題の提案者自身によって提案された上/下界は非常に強力であり，これを打ち破るのは

至難の技であろう．とまれ，MNK に関しては今後の発展が期待されるところである．無論，KSP の研究も終った訳ではない．

## 参考文献

[1] G. B. Dantzig, "Discrete-variable extremum problems," *Opns. Res.* **5**(2) 266–277, April 1957.

[2] H. Iida, *On solving the max-min 0–1 knapsack problem,* Research Report IS-RR-97-0025F, 北陸先端科学技術大学院大学, 923-1292 Japan, June 1997.

[3] H. Iida, "A note on the max-min 0–1 knapsack problem," *Journal of Combinatorial Optimization*, to appear.

[4] T. Kuno, H. Konno and E. Zemel, "A linear-time algorithm for solving continuous maximin knapsack problems," *Opns. Res. Lett.* **10**(1) 23–26, Feb. 1991.

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

[6] S. Martello and P. Toth, "Upper bounds and algorithms for hard 0-1 knapsack problems," *Opns. Res.* **45**(5) 768–778, Sept.–Oct. 1997.

[7] U. Pferschy, D. Pisinger and G. J. Woeginger, "Simple but efficient approaches for the collapsing knapsack problem," *Discrete Appl. Math.* **77**, 271–280, 1997.

[8] T. Yamada, M. Futakawa and S. Kataoka, "Some exact algorithms for the knapsack sharing problem," *Euro. J. Opnl. Res.* **106**, 177–183, 1998.

[9] G. Yu, "On the max-min 0-1 knapsack problem with robust optimization applications," *Opns. Res.* **44**(2) 407–415, March–April 1996.