1996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

ナップサック共有問題の近似・厳密解法*

02501760 防衛大学校情報工学科 二川真由美 FUTAKAWA Mayumi
01700900 同 上 *山田武夫 YAMADA Takeo
01107880 同 上 片岡靖詞 KATAOKA Seiji

1 はじめに

n 個の商品の集合 $N := \{1, 2, \cdots, n\}$ があり，商品 j には重量 w_j と利得 p_j が与えられているとする．さらに，商品は s 個の群に分かれていて，k 番目の商品群を $N^k \subseteq N$ と書くと，$\cup_{k=1}^{s} N^k = N, N^k \cap N^h = \emptyset \ (k \neq h), |N^k| = n_k$ であるとする．ナップサック共有問題 (Knapsack Sharing Problem: KSP) とは，ナップサックに収容可能な商品の集合で，各商品群ごとの総利得の最小値を最大化する問題であり，次のように定式化される [1].

$$\text{KSP}: \left| \begin{array}{ll} \text{Maximize} & \min_{1 \leq k \leq s} \sum_{j \in N^k} p_j x_j \\ \text{subject to} & \sum_{j \in N} w_j x_j \leq c, \\ & x_j \in \{0, 1\}, \quad j \in N \end{array} \right.$$

この問題は $\mathcal{NP}$ 困難であることが証明できる．

本研究では，KSP に対する近似解法と厳密解法を提案し，数値実験によりそれらの性能と特性を評価する．

2 KSP の連続緩和と上界値

KSP の連続緩和は，

$$\text{C(KSP)}: \left| \begin{array}{ll} \text{Maximize} & \min_{1 \leq k \leq s} \sum_{j \in N^k} p_j x_j \\ \text{subject to} & \sum_{j \in N} w_j x_j \leq c, \\ & 0 \leq x_j \leq 1, \quad j \in N \end{array} \right.$$

と書ける．これを以下のように分割して解くことを考える．まず，容量 c を各商品群ごとの容量 $c^1, \cdots, c^s$ に分割し，c^k をパラメータとする相互に独立な s 個の補助問題

$$\text{Aux}^k(c^k): \left| \begin{array}{ll} \text{Maximize} & \sum_{j \in N^k} p_j x_j \\ \text{subject to} & \sum_{j \in N^k} w_j x_j \leq c^k, \\ & 0 \leq x_j \leq 1, \quad j \in N^k \end{array} \right.$$

を考える．$\text{Aux}^k(c^k)$ は連続型ナップサック問題で容易に解くことができ，この最適目的関数値を $\bar{z}^k(c^k)$ と記すと，問題 C(KSP) は次のように書き換えられる．

$$\text{C}^\dagger\text{(KSP)}: \left| \begin{array}{ll} \text{Maximize} & \min_{1 \leq k \leq s} \bar{z}^k(c^k) \\ \text{subject to} & \sum_{k=1}^{s} c^k \leq c, \\ & c^k \geq 0, \ k = 1, \cdots, s \end{array} \right.$$

ここで，商品は商品群ごとに効率順に番号づけられていて，$p_1^k/w_1^k \geq p_2^k/w_2^k \geq \cdots \geq p_{n_k}^k/w_{n_k}^k$ であるとする．$\bar{z}^k(\cdot)$ は単調非減少で区分的に線形な凹関数 [2] なので，関数 $\bar{z}^k(\cdot)$ を横軸方向に加え合わせた関数を $\bar{z}(\cdot)$ とすると，明らかに $\bar{z} := \bar{z}(c)$ は $\text{C}^\dagger$(KSP) の最適目的関数値，すなわち KSP の上界値を与える．$\bar{z}$ は，Kuno 等 [3] のアルゴリズムにより線形時間で計算することができる．このとき，$\bar{z}^k(\cdot)$ は直線 $z = \bar{z}$ と第 u_k 区間で交わるとする．

3 KSP の近似解法と問題の縮小

3.1 近似解法

緩和問題 C(KSP) を解いて u_k を得た．そこで，$\mathring{x}^k = (\mathring{x}_j^k | j \in N^k)$ を，$j < u_k$ のときに 1，それ以外は 0 のベクトルと定義し，$\mathring{x} := (\mathring{x}^1, \cdots, \mathring{x}^s)$ とすると，$\mathring{x}$ は明らかに KSP の 1 つの実行可能解である．これを KSP の自明解と呼ぶ．

グリーディ解は，自明解から出発し，“累積利得が最小の商品群から，残っている商品でナップサックに収容でき，かつ最も効率のよいものを次に採用する”という方針で，逐次商品をナップサックに入れていく方法である．これにより得られる解を $(\underline{x}, \underline{z})$ と表記する．このとき，$(\bar{z} - \underline{z})/\underline{z} \times 100$ はグリーディ解の相対誤差の上限を与える．

3.2 問題の縮小

上で得た上下界値を利用すると，問題中の一部の変数を 0 または 1 に固定でき，問題の規模が縮小される．この釘付けテスト [4] を KSP に導入するために，$k = 1, \cdots, s$, $j = 1, \cdots, n_k$ に対して，

$$\bar{z}(k, j) := \bar{z} - (p_j^k \gamma^k - w_j^k) / \textstyle\sum_{\ell=1}^{s} \gamma^\ell \tag{1}$$

を定義する．ここに，$\gamma^k := w_{u_k}^k / p_{u_k}^k$ である．すると次の定理が成立する．

定理 1 KSP の最適解 x^* においては

(i) $j < u_k$, $\bar{z}(k, j) < \underline{z}$ ならば $x_j^{*k} = 1$,

(ii) $j > u_k$, $\bar{z}(k, j) < \underline{z}$ ならば $x_j^{*k} = 0$,

である $(k = 1, \cdots, s)$.

3.3 数値実験

近似解法と問題縮小アルゴリズムを C 言語を用いて DEC3100 上に実現し，数値実験を行った．実験は，商品データの統計的性質，ナップサック容量，商品数比率の様々な組合せについて，商品数 n が 50 から 250000 までの範囲で計算時間と近似精度，問題縮小率を計測した．これにより，以下の知見を得た．

- データの整列以外の計算時間は，問題のサイズとともにほぼ線形に増加し，相対誤差は減少する．

*OR 学会春季大会 (小樽商大, 1996.5.15-16)

- データの整列以外の計算時間は, 例題のタイプやナップサック容量, 商品数比率の影響をあまり受けない.
- 商品の重量と利得間の相関が強まるにつれて問題の縮小効果は減少するが, 計算時間や相対誤差はあまり変化しない.

4 KSP の厳密解法

4.1 動的計画法

動的計画法により KSP を解く擬多項式時間のアルゴリズムを作成できるが, メモリーの制約から実際には小規模な問題しか解くことができない.

4.2 分枝限定法

KSP を分枝限定法で解くために, 部分問題 $Q = (Q_0, Q_1, Q_X)$ を次のように定義する. $Q_0 \subseteq N$ $(Q_1 \subseteq N)$ は 0 (1) に固定された変数の集合で, Q_X はその他の変数の集合とする. このとき, 商品群 k のうち 1 に固定された商品の総重量と総利得をそれぞれ W_Q^k, P_Q^k とすると, $c_Q = c - (\sum_{k=1}^{s} W_Q^k)$ はナップサックの残容量を表す. 部分問題 Q はもとの問題を次のように修正した KSP として定義される.

1. ナップサック容量を c_Q とする.
2. Q_0, Q_1 に属する商品をすべて除く.
3. $P_Q^k \neq 0$ の場合, 商品群 k に新たに重量 0, 利得 P_Q^k の商品を付加する $(k = 1, \cdots, s)$.

部分問題 Q で $Q_X \neq \emptyset$ のとき, Q_X に含まれる任意の商品 j により, x_j を 0 または 1 に固定した Q の 2 つの子問題 $Q[x_j = 0]$, $Q[x_j = 1]$ が生成でき, このような分枝操作を再帰的に行うことにより KSP の実行可能領域をもれなく, かつ重複なく分割していくことができる. 部分問題 Q の上界値は第 2 節の方法で高速に計算できる. 以上より KSP を解く分枝限定アルゴリズムを構成できる.

4.3 2 分探索法

KSP の最適値を z^* とすると, 任意の正数 z に対し, $z^* \geq z$ か否かの判定は, 次を満たす $\{x_j\}$ が存在するか否かの判定と同値である.

$$\begin{cases} \sum_{j \in N^k} p_j x_j \geq z, \ k = 1, \cdots, s \\ \sum_{k=1}^{s} \sum_{j \in N^k} w_j x_j \leq c, \\ x_j \in \{0, 1\} \end{cases} \tag{2}$$

このような $\{x_j\}$ が存在するとき, z は達成可能といい, そうでないとき達成不可能という. z の達成可能性を判定するために, 次の逆ナップサック問題を考える.

$$\mathrm{IKP}^k(z): \left| \begin{array}{ll} \text{Minimize} & \sum_{j \in N^k} w_j x_j \\ \text{subject to} & \sum_{j \in N^k} p_j x_j \geq z, \\ & x_j \in \{0, 1\} \end{array} \right.$$

この問題の最適値を $c^{*k}(z)$ と記すと, z の達成可能性は $\sum_{k=1}^{s} c^{*k}(z) \leq c$ と同値である. $\mathrm{IKP}^k(z)$ は, $y_j := 1 - x_j$ とおけば, 通常のナップサック問題に変換されるので, その上下界値 $\bar{c}^k(z), \underline{c}^k(z)$ は容易に導け, 厳密値 $c^{*k}(z)$ も比較的容易に計算できる. そこで, z^* の上下界値 $z_r := \bar{z}$, $z_l := \underline{z}$ から出発し, $z_m = \lfloor (z_l + z_r)/2 \rfloor$ として z_m の達成可能性を判定して, z_m が達成可能なら $z_l := z_m$, そうでなければ $z_r := z_m$ とする. これを $z_r - z_l = 1$ となるまで反復すると最後に得られた z_l が z^* で, これが KSP の解を与える.

4.4 数値実験

分枝限定法と 2 分探索法のアルゴリズムを **3.3** と同様に DEC3000/400 上に実装し, 同様の例題について数値実験を行った. その結果の一部を表 1 に示す. 実験全体から得られた知見をまとめると以下のとおりである.

- 計算時間の面から, 2 分探索法が有利と判定される.
- 特に, 商品群の数が多い場合, 前項の傾向が強い.
- いずれの方法でも, 商品の重量と利得間の相関が強まるにつれ, 急激に計算が困難となる.

表 1: 厳密解法の計算時間 (秒)

n	分枝限定法	2 分探索法
200	2.15	0.03
400	4.58	0.06
600	7.42	0.13
800	15.29	0.20
1000	35.20	0.20
1500	122.95	0.57
2000	122.67	0.79

注. 商品の重量と利得は無相関で, ナップサック容量は $c = 250n$, $n_1 = n_2 = n/2$, 分枝戦略は商品番号順である (10 回の平均).

5 まとめ

ナップサック共有問題を新たに定式化し, その近似解法と厳密解法を構築するとともに, 数値実験によりそれらの性能と特性を評価した.

参考文献

[1] 二川, 山田, 片岡, 1995 年 OR 学会秋季大会 1-F-3.

[2] S. Martello, P. Toth, *Knapsack Problems*, John Wiley & Sons (1990).

[3] T. Kuno, H. Konno, *Opns. Res. Letts.*, **10**, (1991).

[4] 今野, 鈴木, 「整数計画法と組合せ最適化」, 日科技連 (1982).