1998年度日本オペレーションズ・リサーチ学会
秋季研究発表会

順序制約付きナップサック問題のDP解法*

02991720　防衛大学校　ナタウット　サムパイブーン†　SAMPHAIBOON Natthawut
01700900　防衛大学校　山田武夫　YAMADA Takeo

1　はじめに

n個の商品 $1,2,\cdots,n$ があり，それらの重量と利得をそれぞれ $w_i, p_i (i=1,2,\cdots,n)$ とする．これらを容量Bのナップサックに詰め込み，総利得を最大とする問題は，ナップサック問題(KP)と呼ばれ，多数の研究がなされている[1], [2]．本稿では，これに対して，商品間に先行順序関係があって，ある商品を採択するには，それに先行するすべての商品が採択されていなければならないような場合を考える．これを以下では**順序制約付きナップサック問題** (PCKP) と呼び，動的計画法による解法を考える．

2　問題の定式化

PCKP は有向グラフを用いて，次のように表現できる．まず，商品に対応して節点集合 $V=\{1,2,\cdots,n\}$ をとる．次に，商品iが商品jに先行するとき，節点iから節点jへ有向枝(i,j)を引き，このような枝全体の集合を$E \subseteq V \times V$ と記す．順序関係の定義から，有向グラフ $G=(V,E)$ にはサイクルは含まれないものとする．すなわち，G は**無閉路有向グラフ** (DAG) であり，その節点はトポロジカルに番号付けられていると仮定してよい．

図 1 はこのようなDAGの一例である．Gにおいて，節点iから節点jへの有向道があるとき，iはjの**先行節点**，jはiの**続行節点**であるという．

Gの部分グラフ $\acute{G}=(\acute{V},\acute{E})$ が**許容部分グラフ**であるとは$\acute{V}$の任意の節点に対し，そのすべての先行節点が$\acute{V}$に含まれることをいう．$\acute{G}$の重量と利得を

$$v(\acute{G}) := \sum_{i \in \acute{V}} p_i, \qquad w(\acute{G}) := \sum_{i \in \acute{V}} w_i$$

と記すと，問題は$w(\acute{G}) \leq B$を満たし，かつ，$v(\acute{G})$を最大とするような許容部分グラフ$\acute{G}$を求めることである．

*日本大学会館 (1998.10.15-16)
†E-mail: nata@cs.nda.ac.jp

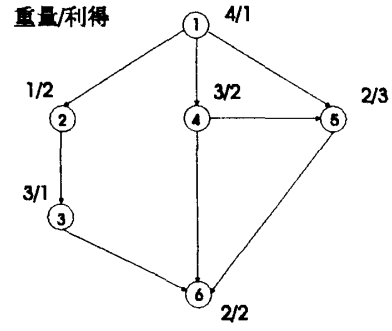

図 1: 無閉路有向グラフ上のPCKP

3　DPの基本関係式

任意の(許容とは限らない)部分グラフ $\acute{G}=(\acute{V},\acute{E})$ において，$\acute{V}$の番号最小の節点を$top(\acute{G}) := min\{v \in \acute{V}\}$とする．このとき，$\acute{G}$から$top(\acute{G})$を取り除いたものを$\acute{G}_L$とし，$\acute{G}$から$top(\acute{G})$と，その下流部分をすべて取り除いたものを$\acute{G}_R$とする(図 2)．$\acute{G}$に$b$だけのナップサック容量が与えられたときに，そこでの最大利得を求める問題を**部分問題**$\acute{G}$と呼び，そのときの最大利得を$v_b(\acute{G})$と記す．さらに$d_b(\acute{G})$を次の決定変数とする．

$$d_b(\acute{G}) = \begin{cases} 1, & \mathrm{top}(\acute{G})\text{が採択されるとき} \\ 0, & \text{そうでないとき} \end{cases}$$

ここで，ベクトル

$$\boldsymbol{v}\,(\acute{G}) := \{v_0(\acute{G}), v_1(\acute{G}), \cdots, v_B(\acute{G})\}$$
$$\boldsymbol{d}\,(\acute{G}) := \{d_0(\acute{G}), d_1(\acute{G}), \cdots, d_B(\acute{G})\}$$

を考え，これらを組み合わせた

$$\boldsymbol{s}\,(\acute{G}) := \{\, \boldsymbol{v}\,(\acute{G}),\, \boldsymbol{d}\,(\acute{G}) \,\}$$

を$\acute{G}$の**解ベクトル**と呼ぶ．これについては以下の関係式が成立する．

1. $\acute{G}$**が単一節点のみの場合** ($\acute{G}=(\{i\},\emptyset)$)

$$\boldsymbol{v}\,(\acute{G}) = \begin{cases} p_i, & b \geq w_i, \\ 0, & b < w_i, \end{cases} \qquad (1)$$

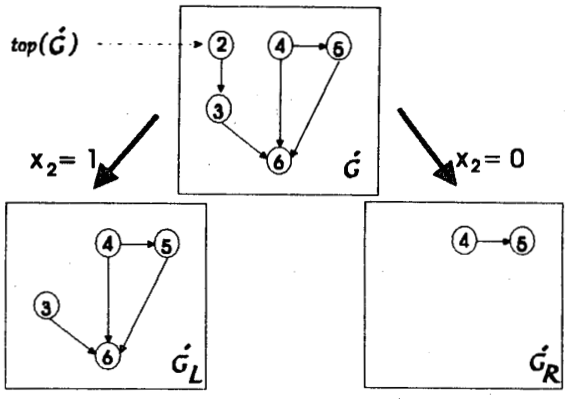

図 2: 部分問題とその子問題

$$d\,(\dot{G}) = \begin{cases} 1, & b \geq w_i, \\ 0, & b < w_i. \end{cases} \qquad (2)$$

2. 一般の場合

$$v_b(\dot{G}) = max\{v_b^L(\dot{G}), v_b^R(\dot{G})\} \qquad (3)$$

$$d_b(\dot{G}) = \begin{cases} 1, & \text{if } v_b^L(\dot{G}) \geq v_b^R(\dot{G}) \\ 0, & \text{if } v_b^L(\dot{G}) < v_b^R(\dot{G}) \end{cases} \qquad (4)$$

但し, $v_b^L(\dot{G}) = p_i + v_{b-w_i}(\dot{G_L})$, $v_b^R(\dot{G}) = v_b(\dot{G_R})$.

4 アルゴリズム

通常のDPと同様に, アルゴリズムは下位の部分問題から順次上向きに解ベクトルを求める部分と, その結果を利用して, 上から下へ商品の採否を決定して行く部分からなる. これらを手続き *Upward*, *Downward* として次に示す.

アルゴリズム *Upward*

step 1 : $\dot{G}$が単一節点のとき, 式(1)と式(2)から, $s\,(\dot{G})$を計算して, これをreturnする.

step 2 : 一般の場合は, $\dot{G_L}, \dot{G_R}$を求め, 再帰呼び出しにより次を計算する.
$s_L = Upward(\dot{G_L})$, $s_R = Upward(\dot{G_R})$;

step 3 : 式(3)と式(4)より, s_L, s_Rから$s\,(\dot{G})$を計算して, これをreturnする.

以下では, 商品iを採択するとき $x_i^* = 1$, そうでないとき $x_i^* = 0$ が出力される.

アルゴリズム *Downward*

step 1 : (初期化)
$x_i^* = 0, \forall i \in V; b = B; \dot{G} = G; v = v_b(\dot{G})$;

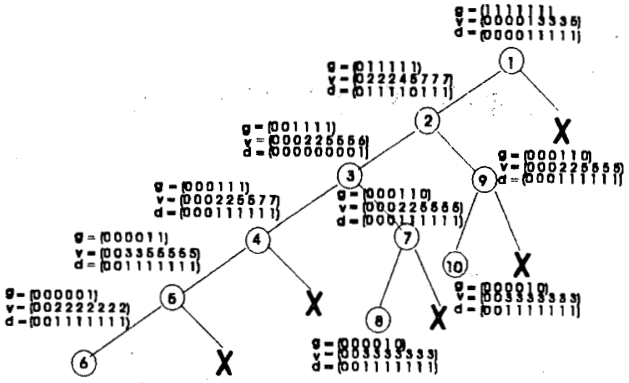

図 3: 実行例

step 2 : $v \leq 0$ なら終了. $d_b(\dot{G}) = 1$ならstep 3へ, そうでなければstep 4へ進む.

step 3 : $i = top(\dot{G}); x_i^* = 1; v = v - p_i; b = b - w_i;$ $\dot{G} = \dot{G_L}$ として step 2 へ.

step 4 : $\dot{G} = \dot{G_R}$ として step 2 へ.

5 実行例

図 1 のグラフに対し, $B = 8$として, *Upward*アルゴリズムを適用したときの実行結果を図 3 に示す. 各ノードの番号は, その子問題の呼び出し順で, ベクトルgは図 1 の節点がその子問題に含まれるか否か, を表している. 最大利得は, 図 3 のノード1のv[8]=5である. これに対し, *Downward*アルゴリズムを適用して, 図 1 の節点 {1, 2, 4} を最適解として得た.

6 むすび

順序制約付きナップサック問題に対するDP解法を示した. 今後, さらに計算量の削減法などを検討する予定である.

参考文献

[1] G. Cho and D. X. Shaw, A depth-first dynamic programming algorithm for the tree knapsack problem, *INFORMS Journal on Computing* 9(1997), 431-438.

[2] D. S. Johnson and K. A. Niemi, On knapsacks, partitions, and a new dynamic programming technique for trees, *Mathematics of Operations Research 8(1983)*, 1-14.