3M-3

排他制約付きナップサック問題に対する解法

山崎 洋祐 [1†]　Riccardo Schiavoni[2]　Manuel Iori[3]　柳浦 睦憲 [4‡]　Silvano Martello[5]
名古屋大学大学院情報科学研究科 [1†]　University of Bologna[2]　University of Modena and Reggio Emilia[3]
名古屋大学大学院情報科学研究科 [4‡]　University of Bologna[5]

1　前書き

ナップサック問題 (knapsack problem) は情報科学の分野における基礎的な問題で，これまでに多くの研究がある．また，ナップサック問題に制約を加えた問題も多数存在する．本論文ではその中のひとつである排他制的付きナップサック問題 (disjunctively constrained knapsack problem) [1, 2, 3] を対象とする．この問題は，同時に選択できないアイテムに関する制約 (排他制約) と容量制約を満たすようにいくつかのアイテムを選択するとき，選択したアイテムの価値の合計を最大化する問題である．排他制約付きナップサック問題は，ナップサック問題や最大独立集合問題 (maximum independent set problem) を特殊ケースとして含む．よって排他制約付きナップサック問題は NP 困難である．この問題は，トラックによる配送計画などを考える際，混載できない荷物の組合せがあるときにどのように荷物を積載すれば良いかを考える場合などに応用することができる．他にも従業員の能力に基づいた人事管理，予算内の物資の購入など多くの応用例の基礎となっている．

2　問題の定義と定式化

入力として，容器の容量 c と，アイテム集合 $V = \{1, \ldots, n\}$ の各アイテム $i \in V$ に対する重み w_i と価値 p_i，および同時に選択できないアイテム対の集合 E が与えられる．出力はアイテムの部分集合 $S \subseteq V$ であり，その部分集合 S に含まれるアイテムのどの対もその両方が E に含まれてはならない．また，部分集合 S に含まれる各アイテムの重みの和は，容量を超えてはならない．この条件下で部分集合 S に含まれるアイテムの価値の和を最大にするのがこの問題の目的である．アイテムが解 S に含まれるならば 1，含まれないならば 0 の値をとる変数 x_i を用いて，問題を以下のように定式化することができる:

$$
\begin{aligned}
(P) \quad & \text{maximize} && \sum_{i=1}^{n} p_i x_i, && (1)\\
& \text{subject to} && \sum_{i=1}^{n} w_i x_i \le c, && (2)\\
& && x_i + x_j \le 1, \quad \forall (i, j) \in E, && (3)\\
& && x_i \in \{0, 1\}, \quad \forall i \in V. && (4)
\end{aligned}
$$

3　上界計算法

アイテム集合 V を頂点集合，排他制約を表す集合 E を辺集合とするグラフを考える．頂点の部分集合で，任意の頂点間に辺が存在するものをクリークと呼ぶ．問題 (P) の任意の実行可能解 x は，任意のクリーク $C \subseteq V$ に対して $\sum_{i \in C} x_i \le 1$ を満たすが，上界値計算において線形計画緩和問題を考える際に，式 (3) の一部をこのようなクリークに基づく制約に置き換えることにより上界の改善が期待できる．しかし，クリークの個数は指数オーダーになり得るため，全てを列挙するのは問題例が大規模な場合などに現実的でない．また，既存研究において問題 (P) の上界を計算する際に一般的に用いられてきた線形計画緩和はその計算に時間がかかる．提案する手法は，特定の条件を満たすクリーク集合を生成することで，上界を計算する際に線形計画問題に対する汎用解法を必要としない計算法である．

提案手法では，以下の条件を満たすようにクリークを生成する．

1. どのアイテムもいずれかのクリークに含まれる．
2. クリークの接続関係がサイクルを持たない．

以上の条件を満たすクリーク集合に対しては，線形計画緩和問題が持つ特別な構造をうまく利用することで双対問題を効率良く解くことができ，高速に上界を与えることができる．

4　発見的解法

反復局所探索法に基づく発見的解法を提案する．探索には 2 種類の近傍を用いた．最初に ADA (add and subsequent drops and addtions) 近傍について述べる．ADA

†Yousuke YAMASAKI　‡Mutsunori YAGIURA
†‡Graduate School of Information Science, Nagoya University

近傍は次の3ステップからなる操作によって生成されうる解集合である：1. 解に含まれないアイテムを1つ解に加える．2. 1で加えたアイテムと排他制約をもつアイテムを解から外す．3. もし容量制約を満たしていないならばある基準の下で優先度の低いアイテムから順に容量制約を満たすまで解からアイテムを外す．そうでないなら解に加えることのできるアイテムを優先度の高い順に解に加える．次にDA (drop and subsequent addtions) 近傍について述べる．DA近傍は次の2ステップからなる操作によって生成されうる解集合である：1. アイテムを1つ解から外す．2. 解に加えることのできるアイテムを優先度の高い順に解に加える．

1回の局所探索では，この2つの近傍を用いて，解の改善がなくなるまで繰り返し近傍を探索する．解の改善がないときには，解に摂動を加えたのち再び局所探索を行う．摂動には，解に含まれず互いに排他制約を持たないアイテムをランダムにいくつか選択し，それらの各アイテムに対してそれを最初に解に加える対象とするADA近傍の操作を適用する，というものを用いた．提案手法では，局所探索と摂動をあらかじめ定めた回数繰り返すものとした．

また，解に含まれないアイテムを2-3木の構造を用いて保持することによって追加すべきアイテムを特定するのに要する時間を削減し，近傍探索の高速化を行った．

5　計算結果

上界の計算においては，線形計画緩和によって上界を計算するのが難しいような非常に大規模な問題例についても短時間で上界を計算することが可能となった．また，問題例によっては問題 (P) の線形計画緩和問題の最適値よりも良い上界が得られた．

発見的解法に関しては，上界の計算において得られた情報に基づいてアイテムの優先度を定めて局所探索法を実行することで，問題例によっては既存研究より良い解を得ることができた．

提案した上界の計算法と発見的解法を分枝限定法に取り入れた結果，辺密度の高い問題例においては，既存のアルゴリズムよりも高速に最適解を求めることができた．その計算結果の一部を表1に示す．これらの問題例はYamadaら[3]の問題例生成プログラムを用いて生成したタイプweakと呼ばれるもので，アイテム数 $n = 1000$, 容量 $c = 1000$ のものである．表中，μ は辺密度を，ISBB-CFは提案する上界値計算法を分枝限定法に取り入れた手法が最適解を得るのに要した計算時間を，CPLEXは汎用ソルバーCPLEX(ver. 10.0)が最適解を得るのに要した計算時間を示す．なお，表中,「T.O.」は30分計算を行っても計算が終了しなかったことを意味する．

表1: 最適解を得るのに要した計算時間 (秒) の比較

		time[sec]	
μ	n	ISBB-CF	CPLEX
0.1	1000	66.8	21.1
0.2	1000	245.7	139.2
0.3	1000	393.0	207.1
0.4	1000	459.9	723.9
0.5	1000	620.4	T.O.
0.6	1000	163.6	1601.5
0.7	1000	163.8	T.O.
0.8	1000	287.3	T.O.
0.9	1000	143.9	T.O.

6　まとめ

排他制約付きナップサック問題に対して線形計画緩和問題に対する汎用解法を必要としない上界の計算法を提案した．計算実験の結果，提案手法が非常に大規模な問題例や，辺密度の高い問題例に対して有効であることを確認できた．また，提案する上界値計算法の情報を用いて局所探索を行うことで局所探索の性能が向上することを確認した．その結果，いくつかの問題例に対しては既存研究よりも良い解を求めることができた．さらに，提案した上界値計算法と発見的解法を組み込んだ分枝限定法が辺密度の高い問題例に対して有効であることを示すことができた．

参考文献

[1] M. Hifi and M. Michrafy, “Reduction strategies and exact algorithms for the disjunctively constrained knapsack problem,” Computers & Operation Research, vol.34, pp.2657–2673, 2005

[2] M. Hifi and M. Michrafy, “A reactive localsearch-based algorithm for the disjunctively constrained knapsack problem,” Journal of the Operational Research Society, vol.57, pp.718–726, 2006

[3] T. Yamada, S. Kataoka and K. Watanabe, “Heuristic and exact algorithms for disjunctively constrained knapsack problem,” IPSJ Journal, vol.43, pp.2864–2870, 2002