整数ナップサック問題が多項式時間で解ける特殊な場合を定める条件について

飯田浩志[*]

概 要

整数ナップサック問題は, よく知られた 0–1 ナップサック問題の数ある拡張の一つである. 0–1 ナップサック問題の拡張ゆえに, 整数ナップサック問題も容易には解けない問題であり, 分枝限定法・動的計画法等の一般的な枠組みを用いて解かざるを得ない. しかしその一方で, ある特殊な場合には多項式時間で解けるということも知られている. 本稿では, この特殊な場合に焦点を当て, これまでに行われた研究を概観するとともに, いくつかの話題を提供する.

キーワード: 組合せ最適化, ナップサック問題, 貪欲法, 多項式アルゴリズム

1 はじめに

数理計画の分野において, すべての変数が 0 もしくは 1 の値しか取り得ないとする整数計画問題を, 全 0–1 計画問題と呼ぶ. この全 0–1 計画問題の中でも, 制約条件が唯一のものは特に 0–1 ナップサック問題 (0–1 Knapsack Problem, 以下 KP) と呼ばれ, 古典的かつ代表的な組合せ最適化問題である. この KP は, "価値と重量なる二つの属性を持つさまざまな品物 (項) を, 与えられたナップサックの重量制限を越えない範囲で詰め, それら詰められた品物の価値の総和を最大にすることを目的とする問題" と解釈することができる. ナップサック問題という名前の由来は, ここにある.

KP は, それ自身が興味深くかつ有用なことに加え, その拡張 (一般化) となる問題が, これまでに数多く提案されている [7, 3]. 本稿では, それら数ある拡張の中から, 整数ナップサック問題 (Unbounded Knapsack Problem, 以下 UKP) を取り上げる. UKP は, 次のように定式化される:

$$
\begin{aligned}
&\text{最大化} && \sum_{j=1}^{n} c_j x_j \\
&\text{制約条件} && \sum_{j=1}^{n} a_j x_j \le b, \\
& && x_j \in \mathbb{N}_0,\ j = 1, 2, \ldots, n.
\end{aligned} \tag{1}
$$

ここに, $\mathbb{N}_0 := \{0, 1, 2, \ldots\}$ である. n 個の各添字 j がそれぞれ一つの項に対応し, 係数 a_j, c_j が項 j の重量と価値を, 変数 x_j が項 jの選択個数を表す (KP では $x_j \in \{0, 1\}$ に制限される). また, b がナップサックの重量制限を表す. 以後, n 次元ベクトル $x := (x_1, x_2, \ldots, x_n)$ を解と呼ぶ. 加えて $\sum_{j=1}^{n} a_j x_j$, $\sum_{j=1}^{n} c_j x_j$ を解 x の重量, 価値と呼ぶことにする. さらに今後は簡便さのため, 両者をそれぞれ ax, cx と略記する. 解 x は, 制約条件 $ax \le b$ を満たす時, (実行) 可能解と呼ばれる. 我々の目的である cx の最大化を実現する可能解を最適解, その最適解の価値を最適値と呼ぶ.

[*]小樽商科大学商学部 〒047-8501 北海道小樽市緑 3 丁目 5-21; E-mail: oggi@res.otaru-uc.ac.jp

Nemhauser と Wolsey [8, p.433] にも見られるように, 通常, UKP は上記のように最大化問題として定式化される. 他方, 次のように最小化問題として定式化されることもある:

$$\begin{array}{rl} \text{最小化} & cx \\ \text{制約条件} & ax \geq b, \\ & x_j \in \mathbb{N}_0,\ j = 1, 2, \ldots, n. \end{array} \tag{2}$$

本稿では, 両者ともに取り扱う.

以降では, 次の三つを仮定する: まず, すべての係数 a_j, c_j ならびに b は正の整数とする; 次に,

$$a_1 < a_2 < \cdots < a_n \tag{3}$$

かつ

$$c_1 < c_2 < \cdots < c_n \tag{4}$$

とする. この仮定は, いずれの項もいくつでも取れるとした UKP に特有の性質である支配関係 (dominance relation) から来る. 一般に支配関係は, 最適値を変えることなく, 与えられた UKP の項数を減らす上で重要な役割を担う. 端的にいえば, $a_j \leq a_k$ かつ $c_j \geq c_k$ ならば, 最大化問題 (1) では $x_k = 0$, 最小化問題 (2) では $x_j = 0$ なる最適解 x が存在する. 具体的には, 最大化問題 (1) の解 x で $x_k > 0$ の時, $x_j \leftarrow x_j + x_k$; $x_k \leftarrow 0$ という操作によって, 解 x より重くなくより価値ある (であろう) 解を構成できるので, 項 kは不要となる (同様の議論から, 最小化問題 (2) では逆に項 j が不要). よってどちらの問題にせよ, $a_j \leq a_k$ かつ $c_j \geq c_k$ なる関係を満足する二項は存在しないとして差し支えない. 支配関係はこれに限られるものではなく, 他にも種類がある. その詳細については, たとえば Andonov 他 [1] を参照されたい; 最後に, 最大化問題 (1) では

$$c_1/a_1 \leq c_2/a_2 \leq \cdots \leq c_n/a_n,$$

また, 最小化問題 (2) では

$$c_1/a_1 \geq c_2/a_2 \geq \cdots \geq c_n/a_n. \tag{5}$$

つまり, UKP の定式化に依らず, 常に項 nが最も好ましいとする. このことから, a_n が b を割り切る (この先, $a_n \mid b$ と書く) 場合には, 問題 (1) および (2) ともに $(0, \ldots, 0, b/a_n)$ が最適解となる.

UKP については, 前述した支配関係をはじめ, 最適解の周期性等, 効率よく解く上で重要となる様々な性質が明らかになってはいるものの, $\mathcal{NP}$ 困難な KP の拡張ゆえに UKP も $\mathcal{NP}$ 困難であり, 一般に容易には解けない. ところが, ある特殊な場合に限れば, UKP は多項式時間で解けることが知られている. これについての先行研究としては, Magazine 他 [5], Hu と Lenard [2], Zukerman 他 [10] がある. 本稿では, UKP が多項式時間で解けるという特殊な場合に焦点を当て, これまでに行われた研究を概観するとともに, まだ明確になっていない事柄にも言及する. 以下, 2 節では, [5, 2] で議論された, 貪欲法が最適解を与える場合について; 3 節では, [10] で議論された, 貪欲法とはまた別の多項式時間解法が適用可能な場合について, それぞれ見ていく.

2 最大化問題と貪欲法

貪欲 (greedy) 法とは, とりあえず目先のことだけを考える枠組みであり, 一般に最適解を与えないことは周知のとおりである. その一方で Magazine 他 [5] は, 等式制約下の最小化問題 (2), すな

わち

$$
\begin{array}{rl}
\text{最小化} & cx \\
\text{制約条件} & ax = b, \\
 & x_j \in \mathbb{N}_0,\ j = 1, 2, \dots, n,
\end{array}
\tag{6}
$$

に対して貪欲法が最適解を与えるための (再帰的な) 必要十分条件を示した. ここで, 問題 (6) では仮定 (5) に加えて, 可能解の存在を保証するために

$$
1 = a_1 < a_j,\ 2 \le j \le n, \tag{7}
$$

も仮定される. 問題 (6) への貪欲法は, 仮定 (5) を踏まえれば, 疑似コードで以下のように書ける:

$$
\begin{array}{l}
\textbf{for } (j = n;\ j > 1;\ j = j - 1)\ \{ \\
\qquad x_j = \lfloor b/a_j \rfloor;\ b = b - a_j x_j; \\
\} \\
x_1 = b;\ /*\ a_1 = 1 \text{ に注意 } */ \\
\text{return } x\,;
\end{array}
$$

特にすべての c_j を 1 とした (6) は, お釣り生成問題 (Change-Making Problem, 以下 CMP) と呼ばれる. この CMP は, 額面 a_j の硬貨 (あるいは紙幣) n 種類を用いてお釣り b を用意するにあたり, 硬貨の総使用枚数 $\sum_{j=1}^{n} x_j$ を最小にするのが目的である. この問題に直面すれば, 誰もが無意識に上記の貪欲法を用いるだろう. CMP の詳細については [7, 第 5 章] を参照されたい.

Magazine 他 [5] の示した条件について述べる前に, 函数を二つ定義する: 一つは $F_j(y)$ ($1 \le j \le n$, $0 \le y \le b$), 伝統的にナップサック函数と呼ばれるものである. 問題 (6) において, 最初の j 種類の項のみ選択可, 重量制限が y である場合の最適値を表す (つまり, $F_n(b)$ が問題 (6) の最適値); もう一つ $H_j(y)$ は, j および y について $F_j(y)$ と同じ制約の下, 前出の貪欲法が与える解の価値を表す. ここでは, Magazine 他 [5] のそれではなく, Hu と Lenard [2] による結果を示す.

定理 (Hu と Lenard, 1976). すべての正の整数 y と, ある固定された j について $H_j(y) = F_j(y)$ とする. もし $a_{j+1} > a_j$ で, p と δ が $a_{j+1} = pa_j - \delta$ および $0 \le \delta < a_j$ から決まる整数ならば, 以下は同値である.

- (a′) すべての正の整数 y について $H_{j+1}(y) \le H_j(y)$,
- (a) すべての正の整数 y について $H_{j+1}(y) = F_{j+1}(y)$,
- (b) $H_{j+1}(pa_j) = F_{j+1}(pa_j)$,
- (c) $c_{j+1} + H_j(\delta) \le pc_j$.

上の (a′) は, Magazine 他 [5] の得た結果へ Hu と Lenard [2] によって追加され, その証明をより簡素なものにした. つけ加えておくべきは, この (a′) と (c) の関係が, (a) と (b) の関係と同じということである. つまり, $\delta < a_{j+1}$ から $c_{j+1} + H_j(\delta) = c_{j+1} + H_{j+1}(\delta) = H_{j+1}(a_{j+1} + \delta) = H_{j+1}(pa_j)$ なので, (c) は $H_{j+1}(pa_j) \le H_j(pa_j)$ と書き直せる.

任意の正の整数 y について $H_1(y) = yc_1 = F_1(y)$, 加えて仮定 (3) から, 定理の (c) を $j = 1, 2, \dots, n-1$ まで積み上げた $n-1$ 本の条件式が $H_n(b) = F_n(b)$, すなわち, 貪欲法が問題 (6) を解く (最適解を与える) のと同値であることを, 定理は示している. 一つ, (c) は $j = 1$ の時には常に成立する. なぜなら $j = 1$ の時に (c) は, $a_1 = 1$ と $H_1(0) = 0$ により, $c_2/a_2 \le c_1/a_1$ に縮退するか

らである．同じ理由で, $a_j \mid a_{j+1}$ の場合には, (c) は成立する．CMP でいうと, 我が国の通貨体系では, 唯一の例外を除いて $a_j \mid a_{j+1}$ である．その唯一の例外にあっても, $5000 = 3 \cdot 2000 - 1000$ から $1 + H_8(1000) = 2 \leq 3 \cdot 1$ なので, 貪欲法が硬貨および紙幣の総使用枚数を最小にすることは保証されている．また, もし現在の通貨体系に, たとえば新たに四十円玉を加えてしまうと, その保証はなくなる．お釣りが 80 円 ($= pa_4$, i.e. 50 以上で最小の 40 の倍数) の場合を考えてみられたい．

さて,『すべての正の整数 y』なる記述から分かるように, 定理は, 重量制限 b の値いかんにかかわらず貪欲法で解ける場合を定めるための, 係数 a_j, c_j が満たすべき必要十分条件を示している．別の見方をすれば, b の値に依存して貪欲法で解けるか否かが左右される場合は定理にあてはまらない．つまり, 問題 (6) が貪欲法で解ける場合は, 定理によって尽くされてはいない．実際, 次の例

j	1	2	3
a_j	1	4	9
c_j	2	3	6
b		13	

は, $a_3 = 3a_2 - 3$ から $c_3 + H_2(3) = 12 \not\leq 9 = 3c_2$ なので, 定理の (c) を満足しないけれども, 貪欲法が最適解 $(0, 1, 1)$ を与える．定理の記述中に重量制限 b が現われないことも, その証左である．

また, [5, 2] では係数 c_j の符号に制約がないことから, 最大化問題 (1) に対しても定理が依然として成立することはすぐに出る．具体的には, 問題 (6) で目的函数 cx を異符号にして最大化問題とし, すべての $-c_j$ をあらためて c_j と書き直した後, 特に $c_1 = 0$ として x_1 をスラック (余裕) 変数にして除外した問題に対しても, 定理は成立する (ただし, 係数 c_j の符号を反転させたことで, 定理の (a$'$) および (c) 中の不等号が逆向きになる)．したがって上で述べたことは, 最大化問題 (1) でも事情は同じである．実際に次の例では, 定理の (c) が, $a_2 = 2a_1 - 1$ から $c_2 + H_1(1) = 3 \not\geq 4 = 2c_1$ (便宜的に項 $(a_0, c_0) = (1, 0)$ を付加して $H_1(1) = 1 \cdot c_0 = 0$) で満たされないものの, 貪欲法が最適解 $(1, 1)$ を与える．

j	1	2
a_j	2	3
c_j	2	3
b	5	

以上の点から, 問題 (6) または (1) が貪欲法で解ける場合すべてを拾い上げるべく定理を拡張することは, 一つ興味深い課題であろう．ただ残念なことに, 任意ではなく特定の b について CMP が貪欲法で解けるか否かを判定する問題は, Pearson [9, p.232] にも言及されているように, $\mathcal{NP}$ 困難であることが既に分かっている．よって, CMP の拡張である (6) について (おそらくは, (6) の特殊形である (1) についても), b を固定した上での効率の良い (多項式時間の) 判定法は望めないだろう．つけ加えると, 定理の (c) は, 各 $H_j(\delta)$ が $O(n)$ で求まるので, 全体としては $O(n^2)$ で判定できる．アルゴリズムの性能に関する基礎事項については, たとえば Korte と Vygen [4] を参照のこと．

3　最小化問題と Zukerman 他の解法

最小化問題 (2) にも, 多項式時間で解ける特殊な場合がある．Zukerman 他 [10] は, 次の条件が成立する時, (2) が彼らの提案する多項式アルゴリズム (後述) で解けることを示した．

$$c_{j+1} \leq \lfloor a_{j+1}/a_j \rfloor c_j, \text{ for } j = 1, 2, \ldots, n-1. \tag{8}$$

ここで, 条件 (8) は仮定 (5) を含意することに注意されたい. ポイントは, 条件 (8) が $x_n \geq \lfloor b/a_n \rfloor$ なる最適解 x の存在を指示することである. Zukerman 他 [10] の多項式アルゴリズムは, 条件 (8) を再帰的に用いるものであり, 疑似コードで以下のようになろう:

$$
\begin{array}{l}
\textbf{for } (j=1;\ j<n;\ j=j+1)\ x_j^* = x_j = 0;\ /*\ \text{clear } x^*, x\ */ \\
x_n^* = +\infty; \\
\textbf{for } (j=n;\ j>=1;\ j=j-1)\ \{ \\
\qquad x_j = \lceil b/a_j \rceil;\ /*\ \text{候補 (可能解)}\ */ \\
\qquad \textbf{if } (cx < cx^*)\ x^* = x; \\
\qquad x_j = \lfloor b/a_j \rfloor;\ b = b - a_j x_j; \\
\qquad \textbf{if } (b == 0)\ \textbf{break};\ /*\ \text{exit loop} */ \\
\} \\
\text{return } x^*;
\end{array}
\tag{9}
$$

ようするに, 高々n 個の候補の中から, その価値が最も小さい解を選択している.

じつは条件 (8) は, $n-1$ 本の不等式一つ一つを単独で見れば, 別の意味で以前から文献に登場していた. それは, 最大化問題 (1) における支配関係として, である (Martello と Toth [6, p.18]). つまり, j を一つ固定するごとに, 最大化問題 (1) で成立すれば, 項 $j+1$が不要となる. 詳しくいうと, 床函数の性質から得られる $a_{j+1} \geq \lfloor a_{j+1}/a_j \rfloor a_j$ と併せれば, 項 $j+1$を $\lfloor a_{j+1}/a_j \rfloor$ 個の項 jで置き換え可能と分かる. では逆に, 条件 (8) に対応する, 最小化問題 (2) における $n-1$ 本の支配関係

$$
c_{j+1} \geq \lceil a_{j+1}/a_j \rceil c_j,\ \text{for } j = 1, 2, \ldots, n-1 \tag{10}
$$

を考えた時, これは, 最大化問題 (1) が多項式時間で解ける場合を定めるだろうか?

結論からいうと, 条件 (10) は, 前節で紹介した定理の (c) に包含される. 何となれば, 問題 (6) から (1) への先の変換: $-cx \leftarrow cx$; $c_j \leftarrow -c_j$; $c_1 \leftarrow 0$ で仮定 (5) は $0 \leq c_2/a_2 \leq \cdots \leq c_n/a_n$ となり, もう一つの仮定 (7) と併せれば, 係数 c_j がすべて非負, すなわち $H_j \geq 0$ が即座に得られるからである (定理の記述中, $p = \lceil a_{j+1}/a_j \rceil$ に留意されたい). 詰まるところ (10) は, 最大化問題 (1) が貪欲法で解けるための十分条件である. このことは, [2, p.195] の系 1 に同じ変換を施しても出る.

では最小化問題 (2) に立ち返って, 条件 (10) と同様に (8) も, 最小化問題 (2) が (9) を用いて解けるための十分条件にすぎない, すなわち必要十分条件ではないのかといえば確かにそのとおりで, 条件 (8) を満足しないにもかかわらず (9) が最適解を与える例として

j	1	2	3
a_j	1	2	3
c_j	1	2	3

あるいは

j	1	2	3
a_j	3	6	9
c_j	3	5	6

等が挙げられる. 前者では, (9) が与える解の重量は常に b に等しい. また, 後者の最適解 x では, 支配関係を考慮すれば容易に分かるように必ず $x_1 + x_2 \leq 1$ だから, $x_3 \geq \lfloor b/a_3 \rfloor$ がいえる (なぜなら最適解 x で $x_3 \leq \lfloor b/a_3 \rfloor - 1$ とした時, $x_1 + x_2 \leq 1$ を用いれば, $3x_1 + 6x_2 + 9x_3 < b$ なる矛盾が導かれるからである). あとは, $\lfloor b/9 \rfloor$ 個の項 3をナップサックに詰めた後の残重量 $b - \lfloor b/9 \rfloor 9$ が $0, \ldots, 8$ の 9 通りすべてについて, (9) が最適解を与えることを確認すれば良い.*

*一般化すると, 最適解 x で $x_1 + x_2 + \cdots + x_{n-1} \leq 1$ として良い時, x が可能解であるためには $x_n \geq \lfloor b/a_n \rfloor$ が必要である. この場合も最適解の候補は高々n 個であり, (9) を用いて解ける. 換言すれば, 項 nを $\lceil b/a_n \rceil - 1$ 個と $k := \min\{j \mid a_j + (\lceil b/a_n \rceil - 1)a_n \geq b\}$ から決まる項 kを一つ, の選択が最適であり, それは (9) によって考慮される.

したがって, 最小化問題 (2) が (9) を用いて多項式時間で解けるという非常に好ましい特殊ケースは, 条件 (8) の定める場合を含んだ上で, まだ広げ得る余地がある. (8) とはまた別の十分条件としては (trivial ではあるものの), $a_n \mid b$ がすぐに思い付く. $a_n \mid b$ なる場合に (9) が与える解は, 唯一の候補 $(0, \ldots, 0, b/a_n)$ である. また, これを汎化すれば, ある j について $a_j \mid b$ かつ $b < a_{j+1}$ ならば最小化問題 (2) は (9) で解ける (この場合の最適値は $\min\{(b/a_j)c_j, c_{j+1}\}$) ということも出る. その他, 前掲した例の前者にもあるように, c_j/a_j の値がすべての項で一致しかつ $a_1 = 1$ の場合にも, (2) は (9) で解ける.†

さらに, 条件 (8) と (10) の対応から, 定理の (c) に対応するものを最小化問題 (2) で類推すれば, (9) を用いて解けるための必要十分条件が得られるのでは, とも思える. これについては, まず, 前出の $H_j(y)$ になぞらえて, 解法 (9) が与える解の価値を表す函数 $H'_j(y)$ $(1 \leq j \leq n,\ 0 \leq y \leq b)$ を

$$H'_j(y) := \begin{cases} \min\left\{\lceil y/a_j \rceil c_j, \lfloor y/a_j \rfloor c_j + H'_{j-1}(y - \lfloor y/a_j \rfloor a_j)\right\}, & j > 1 \\ \lceil y/a_1 \rceil c_1, & j = 1, \end{cases}$$

とする. $H'_j \geq 0$ であること, および $H'_j(y)$ が $y \geq 0$ でのみ定義されることを勘案すれば, (8) を含んでより広い条件としては, 次が妥当であろう‡

$$c_{j+1} - H'_j(a_{j+1} - \lfloor a_{j+1}/a_j \rfloor a_j) \leq \lfloor a_{j+1}/a_j \rfloor c_j, \text{ for } j = 1, 2, \ldots, n-1. \tag{11}$$

だが残念なことに, 次の反例が示すように, この条件は我々の求めるものではない.

j	1	2	3
a_j	3	7	10
c_j	3	6	8
b		14	

この例は (11) を満足するけれども, (9) が最適解 $(0, 2, 0)$ を与えない. さらに, $n = 2$ の時に (11) は $c_2 \leq \lceil a_2/a_1 \rceil c_1$ となるが, 仮定 (5) を含意しないことからも, これは明らかに広すぎる. 実際,

j	1	2
a_j	2	3
c_j	2	3
b	4	

を拾ってしまう (最適解は $(2, 0)$ であって, (9) が与える解 $(1, 1)$ は最適ではない). しかしここで注意したいのは, この例に新たに三つ目の項 $(a_3, c_3) = (1, 1)$ を追加して仮定 (3)–(4) に準じて項を並べ替えれば, 先に示したように (9) で解ける, ということである. このことは, 最小化問題 (2) が (9) を用いて解けるための必要十分条件が, 少なくとも, (8) のような隣り合う二項の係数のみから決まる漸化的な条件式の組では表現し得ない旨を暗示する.

いずれにせよ, 最小化問題 (2) が (9) を用いて解けるための必要十分条件とは如何なるものか? こちらもまた, 興味深い課題であろう.

---

†突きつめれば, $c_j/a_j = \text{const.}$ かつ a_1 が b および他のすべての a_j を割り切る時, (2) は (9) で解ける.

‡前述したように, 定理の (c) が $H_{j+1}(pa_j) \leq H_j(pa_j)$ と書けることから, $a_{j+1} = p'a_j + \delta$, $0 \leq \delta < a_j$ として (ここに $p' = \lfloor a_{j+1}/a_j \rfloor$ である) $H'_{j+1}(p'a_j) \leq H'_j(p'a_j)$ を考えるのが良いと思われるかもしれないが, これを変形すると, $H'_{j+1}(p'a_j) = H'_{j+1}(a_{j+1} - \delta) = \min\{c_{j+1}, H'_j(a_{j+1} - \delta)\} = \min\{c_{j+1}, H'_j(p'a_j)\} = \min\{c_{j+1}, p'c_j\}$ から $\min\{c_{j+1}, p'c_j\} \leq p'c_j$ となる (特に $a_j \mid a_{j+1}$ で $\delta = 0$ の時には (8) に一致する) ので, 何ら有用な知見は得られない.

4　おわりに

計算機の性能が格段に向上しても, 組合せ的爆発を押さえ込むことは不可能である. この意味で, 与えられた問題が多項式時間で解けることが即座に判定できれば, 実用上これに勝るものはない. 解法を実装する上での技法云々にも増して, アルゴリズムを正しく選ぶことが, 何より肝要である. ここでは, 整数ナップサック問題が多項式時間で解ける特殊な場合を定める条件について, これまでに行われた研究を概観した上で, さらなる課題を二つ掲げた. 本稿が, 今後の議論の端緒となれば幸いである.

参考文献

[1] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," *European Journal of Operational Research* **123**(2) 394–407 (2000).

[2] T. C. Hu and M. L. Lenard, "Optimality of a heuristic solution for a class of knapsack problems," *Operations Research* **24**(1) 193–196 (1976).

[3] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer-Verlag 2004.

[4] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics 21)*, Springer-Verlag Berlin Heidelberg 2002, 2nd ed. (2000, 1st ed.).

[5] M. J. Magazine, G. L. Nemhauser, and L. E. Trotter, Jr., "When the greedy solution solves a class of knapsack problems," *Operations Research* **23**(2) 207–217 (1975).

[6] S. Martello and P. Toth, "An exact algorithm for large unbounded knapsack problems," *Operations Research Letters* **9**(1) 15–20 (1990).

[7] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, Wiley-Interscience, Chichester, England, 1990.

[8] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, 1999, paperback reprinted (1988, hardcover original).

[9] D. Pearson, "A polynomial-time algorithm for the change-making problem," *Operations Research Letters* **33**(3) 231–234 (2005).

[10] M. Zukerman, L. Jia, T. Neame, and G. J. Woeginger, "A polynomially solvable special case of the unbounded knapsack problem," *Operations Research Letters* **29**(1) 13–16 (2001).

追記: [3, 4] の書評が *Operations Research Letters* **33**(2) 216–219 (2005) に掲載されている, 参考までに.