Two Topics in Dominance Relations for the Unbounded Knapsack Problem

IIDA Hiroshi*

*Dept of Information and Management Science, Otaru University of Commerce, Otaru 047-8501, Japan; *E-mail: oggi@res.otaru-uc.ac.jp*

Abstract: On the unbounded knapsack problem, dominance relations play a crucial role to reduce items to be considered in a given instance. This article picks up two topics in dominance relations. One is a connection between dominance relations and polynomially solvable special cases, and the other is on unusual dominance relations.

Keywords: Combinatorial optimisation, Unbounded knapsack problem, Dominance relation, Polynomially solvable special case.

INTRODUCTION

This article deals with the unbounded knapsack problem (UKP), in which given a knapsack of capacity c and n types of items of profit and weight we pack the items into the knapsack so that the total profit of packed items is maximised without the total weight of those exceeding c. The UKP is formulated as follows:

$$z_{\max\leq} = \max_x \left\{ \sum_{j=1}^{n} p_j x_j \;\middle|\; \sum_{j=1}^{n} w_j x_j \leq c\,; \quad x_j \in \mathbb{N}_0,\ j \in N \right\}, \qquad (1)$$

where $N := \{1, 2, \ldots, n\}$, and each element $j \in N$ indicates an item of profit p_j and weight w_j. Differing from the conventional 0–1 knapsack problem (KP) of $x_j \in \{0, 1\}$, UKP (1) provides unbounded copies of every item as $x_j \in \mathbb{N}_0 := \{0, 1, 2, \ldots\}$. As seen in, e.g., Nemhauser and Wolsey [1, p.433] UKP is ordinarily formulated as a maximisation problem whilst there also exists a minimisation formulation:

$$z_{\min\geq} = \min_x \left\{ \sum_{j=1}^{n} p_j x_j \;\middle|\; \sum_{j=1}^{n} w_j x_j \geq c\,; \quad x_j \in \mathbb{N}_0,\ j \in N \right\}. \qquad (2)$$

We discuss both versions of UKP. To make it simple, throughout the article we assume that p_j, w_j and c are all positive integers regardless of the formulation of UKP. Moreover, only for (1), to exclude useless items we assume $w_j \leq c$ for all j. Also, let $x := (x_1, x_2, \ldots, x_n)$. Then such an n-vector is called a solution, and one satisfying the constraint is said to be feasible. For the sake of brevity we sometimes employ notation like $p > 0$, representing $p_j > 0$ for all j.

Besides the periodicity (see, e.g., [1, Chapter II.6] or Kellerer et al [2, Section 8.2]) UKP possesses one more property viz dominance relations studied by, e.g., Martello and Toth [3], Dudziński [4], Zhu and Broughan [5], Andonov et al [6] and [7]. Dominance relations allow to replace a given instance with an equivalent smaller-sized one. To take an example, if $p_j \geq p_k$ and $w_j \leq w_k$ hold in an instance of (1), then the jth item dominates the kth. In this case, the kth item is redundant because a solution of $x_k > 0$ does not degenerate by replacing all the kth items packed with the jth. As for (2) in the same case, conversely, the kth dominates the jth given the same argument. Furthermore, on UKP, polynomially solvable special cases have also been studied by, e.g., Magazine et al [8], Hu and Lenard [9] and Zukerman et al [10].

In the remainder of this article, Section 1 connects dominance relations with polynomially solvable special cases, and Subsection 1.1 discusses the extendability of the polynomially solvable special cases treated in Section 1. In Section 2 we present three dominance relations unusual in some sense.

1 Dominance Relations and Polynomially Solvable Special Cases

As stated in Introduction we may assume that there is no pair of items j, k fulfilling $p_j \geq p_k$ and $w_j \leq w_k$. Therefore in this section we assume $p_1 < p_2 < \cdots < p_n$ and $w_1 < w_2 < \cdots < w_n$.

It was shown in [10] that the condition

$$p_{j+1} \leq \lfloor w_{j+1}/w_j \rfloor p_j, \text{ for } j = 1, 2, \ldots, n-1 \qquad (3)$$

implies that the minimisation problem (2) can be solved by a polynomial algorithm. In fact, as in [3, p.18] (3) is (for each j fixed) a dominance relation for the maximisation problem (1), called *multiple dominance* in [6, p.396]. We are going to show that a condition

$$p_{j+1} \geq \lceil w_{j+1}/w_j \rceil p_j, \text{ for } j = 1, 2, \ldots, n-1 \qquad (4)$$

analogous to (3), which is (also for each j fixed) a dominance relation for (2), implies that (1) can be solved

1874-1142/08

by a greedy algorithm. The following corresponds to Lemma 3.2 in [10], and is proved in a similar way.

Lemma 1. If (4) holds then there exists an optimal solution $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ for the maximisation problem (1) with $x_n^* = \lfloor c/w_n \rfloor$.

proof. There is an optimal solution x^* of $x_j^* < \lceil w_{j+1}/w_j \rceil$ for all $j \in N \setminus \{n\}$ because by (4) the $(j+1)$-st item is at least not worse than $\lceil w_{j+1}/w_j \rceil$ copies of the jth. Suppose that Lemma 1 does not hold, which gives $x_n^* < \lfloor c/w_n \rfloor$ as well as a solution $(0, \ldots, 0, \lfloor c/w_n \rfloor)$ being not optimal, i.e., $\lfloor c/w_n \rfloor p_n < px^*$. Then, we have

$$\begin{aligned}\lfloor c/w_n \rfloor p_n &< \sum_{j=1}^{n} p_j x_j^* \\ &\le \sum_{j=1}^{n-1} \left(\lceil w_{j+1}/w_j \rceil p_j - p_j \right) \\ &\qquad + \lfloor c/w_n \rfloor p_n - p_n \\ &\le \lfloor c/w_n \rfloor p_n - p_1,\end{aligned}$$

reading a contradiction $p_1 < 0$.

Then, using (4) recursively, a greedy algorithm processing items from the nth to the 1st solves (1) with (4). The algorithm actually produces the x^* having appeared in the proof. Indeed, in a stage where the jth item is packed, residual capacity c' for the stage is less than w_{j+1} then $x_j^* = \lfloor c'/w_j \rfloor < \lceil w_{j+1}/w_j \rceil$. In fact, the result obtained here can be drawn from the preceding work by Magazine et al [8]. In what follows we will elaborate it.

Magazine et al [8] revealed necessary and sufficient conditions for a greedy algorithm solving the equality-constrained minimisation UKP formulated as:

$$z_{\min=} = \min_x \left\{ \sum_{j=1}^{n} p_j x_j \;\middle|\; \sum_{j=1}^{n} w_j x_j = c\,; \quad x_j \in \mathbb{N}_0,\ j \in N \right\}, \tag{5}$$

$$\begin{aligned}&p_1/w_1 \ge p_2/w_2 \ge \cdots \ge p_n/w_n, \\ &1 = w_1 < w_j,\ 2 \le j \le n.\end{aligned} \tag{6}$$

Before stating the conditions, we introduce two functions. One is $F_j(y)$ $(1 \le j \le n,\ 0 \le y \le c)$ traditionally called a *knapsack function*, which is restricted in (5) so that only the first j items are available and the capacity is y, where $z_{\min=} = F_n(c)$. Under the same restrictions as those on $F_j(y)$, the other $H_j(y)$ is a profit gained greedily, being formulated as $H_j(y) := \lfloor y/w_j \rfloor p_j + H_{j-1}(y - \lfloor y/w_j \rfloor w_j)$ for $j > 1$; otherwise, $H_1(y) := yp_1$.

Theorem 1. (Hu and Lenard, 1976) Suppose $H_j(y) = F_j(y)$ for all positive integers y and some fixed j. If $w_{j+1} > w_j$ and m and γ are the unique integers for which $w_{j+1} = mw_j - \gamma$ and $0 \le \gamma < w_j$, then the following are equivalent.

(a′) $H_{j+1}(y) \le H_j(y)$ for all positive integers y,
(a) $H_{j+1}(y) = F_{j+1}(y)$ for all positive integers y,
(b) $H_{j+1}(mw_j) = F_{j+1}(mw_j)$,
(c) $p_{j+1} + H_j(\gamma) \le mp_j$.

Hu and Lenard [9] added (a′) to the one in [8], which simplified the proof. Notice that (c) of $j = 1$ always holds because it is reduced to $p_2/w_2 \le p_1/w_1$, supported by (6). For the same reason, $w_j \mid w_{j+1}$, denoting that w_j divides w_{j+1} evenly, validates (c).

Owing to no restriction on the sign of p_js in [8, 9], (5) admits an operation such that we first add a minus sign to all p_js in (5) so as to make (5) a maximisation form, next replace $-p_j$ with p_j for all j, and last assign 0 to p_1. Hence, Theorem 1 still holds for the resulting (1) except that the operation causes both '$\le$' in (a′) and (c) to be '$\ge$', and (6) also to be reversed as

$$p_1/w_1 \le p_2/w_2 \le \cdots \le p_n/w_n. \tag{7}$$

Concerning the resultant (1), $w > 0$ and (7) of $p_1 = 0$ give $p \ge 0$; thus we have $H_j \ge 0$, which leads to (4)⇒(c). Consequently Theorem 1 proves that a greedy algorithm solves (1) with (4). Incidentally, this can also be drawn from applying the operation as to p_js to Corollary 1 in [9, p.195]. Specifically, the condition $p_{j+1} \le mp_j - \gamma p_1$ shown therein is directly reduced to (4).

1.1 On the Extendability of the Polynomially Solvable Special Cases

Regarding Theorem 1 a remark is that it is not applicable to all greedily-solvable instances. On the equality-constrained minimisation UKP (5), the following instance is greedily solvable:

j	1	2	3
p_j	2	3	6
w_j	1	4	9
c		13	

(8)

However (8) does not satisfy (c) as $p_3 + H_2(3) = 12 \not\le 9 = \lceil w_3/w_2 \rceil p_2$. Thus Theorem 1 argues that an instance of p_j, w_j fulfilling (c) for all $j \in N \setminus \{n\}$ is irrespective of c solved by a greedy algorithm. In another view, Theorem 1 excludes an instance whose greedily-solvability is on the value of c. This will also be shown by the physical evidence of (c) not including the capacity c.

In addition, on the maximisation problem (1), the following instance is a counterpart to (8):

j	1	2
p_j	2	3
w_j	2	3
c	5	

(9)

An optimal solution $(1, 1)$ is greedily obtained whereas (9) does not satisfy (c) as $p_2 + H_1(1) = 3 \not\geq 4 = \lceil w_2/w_1 \rceil p_1$. In this regard it may seem challenging to extend Theorem 1 as far as being applicable to all greedily-solvable instances of (5) (or (1)), yet it is known that determining whether a greedy algorithm solves the change-making problem (CMP, arising in adding $p = 1$ to (5)) of some specific c is $\mathcal{NP}$-hard as stated in Pearson [11, p.232]. Hence, on account of being an extension of CMP, unless $\mathcal{P} = \mathcal{NP}$ we could not expect a polynomial algorithm which determines whether a greedy algorithm solves (5) (most probably (1) either) of some specific c.

On the other hand how about (3)? In the rest of this section, on the minimisation problem (2) we assume (6), which is implied by (3). Here we will briefly mention the work in [10]. A point is that the condition (3) implies that there is an optimal solution x with $x_n \geq \lfloor c/w_n \rfloor$. Then, using (3) recursively, an algorithm proposed in [10] gives profit $G_n(c)$ computed as

$$G_j(y) := \begin{cases} \min\{\lceil y/w_j \rceil p_j, \lfloor y/w_j \rfloor p_j \\ \quad +G_{j-1}(y - \lfloor y/w_j \rfloor w_j)\}, & j > 1, \\ \lceil y/w_1 \rceil p_1, & j = 1. \end{cases}$$

The polynomially solvable special case for the maximisation problem (1) defined by (4) is that a greedy algorithm works whilst the case by (3) is not so, which will reflect the property of an inequality-constrained minimisation problem; that is, we can make an infeasible solution of (2) feasible by packing more items whereas in the other two versions of UKP, once the total weight of a solution has exceeded the capacity, the solution remains infeasible even by doing so.

In respect of this we note in passing why the proof of Theorem 1 is not suitable for (2). First of all, for (2), it will be natural to define $H_j(y) := \lceil y/w_j \rceil p_j$. Showing the contrapositive of (c) $\Rightarrow$ (a′), for example, we assume $\overline{y} > w_{j+1}$ where $\overline{y}$ denotes the smallest integer for which (a′) fails. The assumption is valid to UKPs of $= c$ or $\leq c$, one reason for which is that $H_{j+1}(y) = H_j(y)$ for any $y < w_{j+1}$; however, it is not valid to that of $\geq c$ because the $(j+1)$-st item can be packed against $y < w_{j+1}$ due to $H_{j+1}(\cdot)$ defined for (2).

In fact, (3) is not a necessary condition either, and therefore we can consider another sufficient condition for the case solvable by the algorithm in [10]. Although trivial, $w_n \mid c$ is one of what we want, where optimal value is $(c/w_n)p_n$ because by (6) for any feasible x we have $(c/w_n)p_n \leq (p_n/w_n)wx \leq px$. To take an example, the following instance satisfies $w_n \mid c$:

j	1	2
p_j	2	3
w_j	2	3
c	6	

(10)

Here we would like to add three points. First, the condition $w_n \mid c$ is indeed not a special case of (3) because in (10) $p_2 \not\leq \lfloor w_2/w_1 \rfloor p_1$. Second, (10) with $c = 4$ replaced is not solved by the algorithm in [10]; thus, it will be inevitable that a necessary and sufficient condition to define a class solvable by the algorithm in [10] includes the capacity, differing from (c) in Theorem 1. Last, when $w_j \mid c$ and $c < w_{j+1}$ for some j, the instance also is solved by the algorithm in [10].

To the best of our knowledge, a necessary and sufficient condition under which the algorithm in [10] delivers an optimal solution to (2) is not yet found out. To investigate (2), taking account of the transformation of (5) into (1), one might consider an equality-constrained maximisation UKP; however, it shall involve $w_1 = -1$ beyond the scope of this article having assumed $w > 0$ so far.

2 Unusual Dominance Relations

Any dominance relation proposed hitherto is to our knowledge concerned only with a relation amongst items, and does not involve the capacity. This section presents three dominance relations each of which involves the capacity. Following [2] $w_{\min}$ and $w_{\max}$ henceforth denote $\min_{j \in N} w_j$ and $\max_{j \in N} w_j$ respectively. In what follows we formulate three unusual dominance relations as Lemmas 2–4.

Lemma 2. For (1), if $w_1 \mid c$ (i.e., w_1 divides c evenly) and the 1st item is of maximum ratio of profit to weight (p_j/w_j) amongst all, then the 1st item dominates the others with optimal value $p_1 c/w_1$.

Whilst we can examine this relation in $O(n)$ time, it appears not to be worthwhile to apply this relation to a given instance because in that case, even crude upper $\lfloor p_1 c/w_1 \rfloor$ and lower $p_1 \lfloor c/w_1 \rfloor$ bounds coincide, and therefore an algorithm based on branch-and-bound could terminate forthwith, only visiting the root node. In addition, supposing that the 2nd item is of second maximum ratio of profit to weight, if $w_2 \mid (c - \lfloor c/w_1 \rfloor w_1)$ then $(\lfloor c/w_1 \rfloor, (c - \lfloor c/w_1 \rfloor w_1)/w_2, 0, \ldots, 0)$ is an optimal solution. In this case we could say that the pair of the 1st and 2nd items dominate the others, and so on. Needless to say an argument similar to that for the first one is suitable for (2).

Lemma 3. Only for (1) if, for some $k \in N$,

$$w_{\min} + w_k > c \text{ and}$$
$$\exists\, l \in N \setminus \{k\} \text{ such that } p_l \geq p_k, \qquad (11)$$

then the lth item dominates the kth.

Under (11), on a solution comprising one kth item, feasibility prohibits us from packing more items. Hence, if there is another item of profit not less than p_k, then the item shall construct a possibly improved (at least not worse) solution. Moreover, a feasible solution x of total profit px not less than p_k yielded in some way (e.g., greedily) will be of use for the lth item in (11). Note that (11) is also suitable for KP; however, it conversely implies that (11) does not utilise the nature of UKP.

In addition to (11), if an item of maximum profit $\max_j p_j$ is only one, then the item remains even if it fulfills the first half of (11); otherwise, at least one of those remains. In short, not all items of maximum profit are eliminated by (11). Further, the first half of (11) implies $2w_k > c$; thus, (11) holds only if $w_k > \lfloor c/2 \rfloor$. In the case of $w_{\min} > \lfloor c/2 \rfloor$, just one item of maximum profit remains. Conversely, not even an item is discarded by (11), provided $w_{\min} + w_{\max} \leq c$. Also, (11) is a little bit notable for a sense such that in the following instance, the 3rd item heavier dominates the 2nd lighter.

j	1	2	3
p_j	1	2	3
w_j	5	6	7
c		10	

Lemma 4. Only for (2), if $c \leq w_j < w_k$ and $p_j < p_k$, then the jth item dominates the kth.

We may thereby assume that an item of weight not less than c is at most one in (2). More precisely, if there are items of weight not less than c, then only one item of minimum profit amongst those remains. Recall that, for (2), $w_j \geq w_k$ and $p_j \leq p_k$ lead to the elimination of the kth item. Incidentally, the second condition of Lemma 4 is not $p_j \leq p_k$ but $p_j < p_k$ because if not so, in the case of $p_j = p_k$, it shall conflict with the claim of the usual dominance relation just mentioned, namely $w_j \geq w_k$ and $p_j \leq p_k$.

ACKNOWLEDGEMENTS

I would just like to thank anonymous referees for consuming her/his precious time and giving valuable comments for me.

REFERENCES

[1] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. (paperback reprinted) NY: Wiley 1999.

[2] Kellerer H, Pferschy U, Pisinger D. Knapsack problems. Berlin: Springer 2004.

[3] Martello S, Toth P. An exact algorithm for large unbounded knapsack problems. Opns Res Lett 1990; 9(1): 15-20.

[4] Dudziński K. A note on dominance relation in unbounded knapsack problems. Opns Res Lett 1991; 10(7): 417-9.

[5] Zhu N, Broughan K. On dominated terms in the general knapsack problem. Opns Res Lett 1997; 21(1): 31-7.

[6] Andonov R, Poirriez V, Rajopadhye S. Unbounded knapsack problem: Dynamic programming revisited. Euro J Opnl Res 2000; 123(2): 394-407.

[7] Iida H. On a condition under which the unbounded knapsack problem can polynomially be solved. Discussion paper series no. 101, Center for Business Creation, Otaru University of Commerce 2005 (in Japanese); available as ⟨http://www.res.otaru-uc.ac.jp/˜oggi/study/no101.pdf⟩

[8] Magazine MJ, Nemhauser GL, Trotter Jr. LE. When the greedy solution solves a class of knapsack problems. Opns Res 1975; 23(2): 207-17.

[9] Hu TC, Lenard ML. Optimality of a heuristic solution for a class of knapsack problems. Opns Res 1976; 24(1): 193-6.

[10] Zukerman M, Jia L, Neame T, Woeginger GJ. A polynomially solvable special case of the unbounded knapsack problem. Opns Res Lett 2001; 29(1): 13-6.

[11] Pearson D. A polynomial-time algorithm for the change-making problem. Opns Res Lett 2005; 33(3): 231-4.

Received: October 11, 2007 Revised: March 20, 2008 Accepted: April 3, 2008