Towards an alternative relaxation for the multiple-choice knapsack problem

IIDA Hiroshi

Abstract

The linear programming relaxation is a typical prescription for solving the multiple-choice knapsack problem. The efficiency of the relaxation applied to the problem has been validated in the literature whereas it is also well known that the relaxation does not always work well for the problem. In this paper we have devised another relaxation for the problem. With profound regret, the result is negative.

Keywords: knapsack; multiple-choice constraint; linear programming relaxation; subset-sum problem

1 Introduction

This paper focuses on the Multiple-Choice Knapsack Problem (MCK), which has been intensively studied in the last two decades. In addition, the problem has many applications as summarized in Dyer et al [7]. Recently, an application of MCK was presented by Pisinger [17]. Also, an early review of MCK is seen in Dudziński and Walukiewicz [4].

The MCK is an extension of the classical 0–1 knapsack problem (KP). The KP is, given n items of profit and weight, to pack the items into a knapsack of capacity c so that the total profit of the packed items is maximized without the total weight of those exceeding the capacity. The KP is formally stated as follows:

$$\text{(KP)} \quad \text{maximize} \quad \sum_{j=1}^{n} p_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} w_j x_j \leq c$$

$$x_j \in \{0,1\}, \; j=1,2,\ldots,n,$$

where each index j indicates an item. On coefficients p_j, w_j and 0–1 variable x_j corresponding to an item associated with j: the first two represent the profit and weight of the item respectively; the last does the choice of it as $x_j = 1$ (packed)/0 (otherwise). For a comprehensive overview of recent studies on KP, see Martello et al [13].

Furthermore in MCK more complicated, all items are split into several classes so that any pair of the classes is mutually disjoint, and we must select just one item in each class. To formulate MCK, we introduce several notation: To begin with a class N_i of cardinality n_i, i.e. $N_i = \{1, 2, \ldots, n_i\}$. In addition, $m (\geq 2)$ classes are given and $\Sigma_{i \in M} n_i = n$, where $M = \{1, 2, \ldots, m\}$. In what follows we assume that an element in a class is one-to-one correspondence to an item, and we call an item associated with $j \in N_i$ the j-th item in N_i. On the three corresponding to the j-th item in N_i: the profit and weight are denoted by p_{ij}, w_{ij} respectively; 0–1 variable is by x_{ij}. Also, we call an n-vector of $x = (x_{11}, x_{12}, \ldots, x_{m,n_m})$ solution. Now, the MCK is formulated as follows:

$$\text{(MCK)} \quad \text{maximize} \quad \sum_{i \in M} \sum_{j \in N_i} p_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{i \in M} \sum_{j \in N_i} w_{ij} x_{ij} \leq c \tag{2}$$

$$\sum_{j \in N_i} x_{ij} = 1, \; i \in M \tag{3}$$

$$x_{ij} \in \{0,1\}, \; i \in M, \; j \in N_i. \tag{4}$$

Throughout this paper without loss of generality we assume that: profit p_{ij}

and weight w_{ij} for any i, j and the capacity c are all positive integers; $n_i \geq 2$ for any $i \in M$; in order to exclude an infeasible or trivial problem,

$$\sum_{i \in M} \min_{j \in N_i} w_{ij} \leq c < \sum_{i \in M} \max_{j \in N_i} w_{ij}. \tag{5}$$

A typical relaxation for MCK is the linear programming (LP) relaxation in which the constraint (4) is relaxed as $x_{ij} \geq 0$ for any i, j. We call the resultant problem LMCK, for short. The LMCK has been utilized to solve MCK so far, e.g. Sinha and Zoltners [18], Armstrong et al [2], Dyer et al [6], and Pisinger [15]. Moreover, LMCK itself has also been studied by, e.g. Glover and Klingman [9], Zemel [20, 21], Dyer [5], and Dudziński and Walukiewicz [3]. Both [21] and [5] especially include a linear time algorithm for LMCK.

On the other hand, two types of Lagrangian relaxation (see, e.g. Fisher [8]) were examined by Nauss [14], in each of which the constraint (2) or (3) is absorbed into the objective function (1). On a comparison between the two, computational experiments in [14] explains that an algorithm which incorporates the first relaxation, i.e. based on (2), dominates another which does the second (plus one more). The first relaxation was also employed in Aggarwal et al [1] and the aforementioned [7]. In fact, as pointed out in [14], the dual of the first relaxation is equivalent to LMCK, which is also implied in [1] (p.221), and can also be seen in an algorithm proposed in [7] to solve the dual.

The remainder of this paper is organized as follows: In Section 2 we briefly review the competitor LMCK. In Section 3 we construct a relaxation problem which will give a bound tighter than that of LMCK.

2 Linear programming relaxation

A point for solving a given problem is to reduce the problem, in other words, to eliminate items sans which the objective function value can be maximized. The following was proved in [14], which is efficient to reduce MCK (see [18], TABLE II in p.511).

Theorem (**Nauss, 1978**) If $p_{ij} \geq p_{ik}$ and $w_{ij} \leq w_{ik}$, then adding the constraint $x_{ik}=0$ to MCK, has no effect on the optimal solution value for the MCK.

Now we may assume that in each class there exist no two items so that one has the same profit or weight as the other's. Moreover for LMCK, the following was proved in [18].

Proposition (**Sinha and Zoltners, 1979**) Let $j,\ k,\ l \in N_i$ with $w_{ij} < w_{ik} < w_{il}$ and $p_{ij} \leq p_{ik} \leq p_{il}$ and $(p_{ik}-p_{ij})/(w_{ik}-w_{ij}) \leq (p_{il}-p_{ik})/(w_{il}-w_{ik})$; then an optimal solution to LMCK exists with $x_{ik}=0$.

By Theorem and Proposition, it follows that promising items for LMCK form an upper convex boundary in each class, see Figure 1. In the following we review an upper bound given by LMCK, that is, its optimal (solution) value. For simplicity we assume that LMCK has already been reduced, and in each class all remaining items are ordered so that $w_{i1} < w_{i2} < \cdots < w_{in_i}$, which also induces $p_{i1} < p_{i2} < \cdots < p_{in_i}$. Note that since either Theorem or Proposition does not rule out an item of minimum weight in each class, the first half of (5) is still valid while we assume the latter half again if necessary.

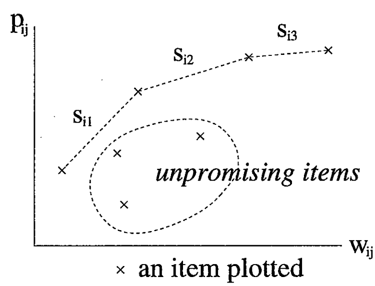

Figure 1: promising items in N_i for LMCK

First of all we define a *slope* $s_{ij}=(p_{i,j+1}-p_{ij})/(w_{i,j+1}-w_{ij})$ for $i \in M$,

$1 \le j < n_i$, which is as in Figure 1 a gradient of the line joining the j-th and $(j+1)$-st items in N_i. As seen in, e.g. [15], it is known that obtaining the optimal value is finding a slope s_{kl} satisfying $W \le c < W - w_{kl} + w_{k,l+1}$, where

$$W = \sum_{i \in M} \left\{ w_{i1} + \sum_{\{j \mid s_{ij} > s_{kl}\}} (w_{i,j+1} - w_{ij}) \right\} \tag{6}$$

(We assumed for simplicity that there exists no slope equal to s_{kl}). The W reduces to a weight sum of m items in which no two items are in the same class, since in each class slopes have already been sorted in descending order by the assumption. To find such s_{kl}, first, we sort all slopes in nonascending order, and initialize W with $\Sigma_{i \in M} w_{i1}$. Next, on the head of a sequence of the slopes sorted, say s_{41}, we replace w_{41} included in W with w_{42}, and exclude the s_{41} from the sequence. Namely we augment W along the slope s_{41}, since the greater the s_{ij}, the greater the profit gained proportional to the increased part of W by a replacement with s_{ij}. Similarly, while $W \le c$ do the replacement as the head of the sequence says. Consequently, with W (6) of $W \le c < W - w_{kl} + w_{k,l+1}$, we shall find s_{kl} as a slope concerned with the last replacement. Once the s_{kl} is found, the optimal value is obtained as $P^W + s_{kl}(c - W)$, where P^W is a profit sum of m items each of which contributes to W. It should be noted that in the solution corresponding to the optimal value just obtained, at most two x_{ij}'s violate the integrality as

$$\begin{cases} x_{kl} = (W - w_{kl} + w_{k,l+1} - c)/(w_{k,l+1} - w_{kl}) \\ x_{k,l+1} = 1 - x_{kl} = (c - W)/(w_{k,l+1} - w_{kl}) \end{cases}$$

even though the integrality for all x_{ij}'s was relaxed, which distinctly accounts for the efficiency of LMCK.

On the other hand, LMCK includes an issue. Specifically, the optimal value is sensitive to the distribution of items. When profits and weights are randomly distributed in a fixed range, Proposition plays a central role and LMCK works efficiently. In Example 1 below, however, Proposition results in neglecting characteristics of the distribution of items.

Example 1. Consider an instance of MCK with two classes; Items in each class are $(p_{i1}, w_{i1}) = (1, 1)$, $(p_{i2}, w_{i2}) = (2, 9)$, $(p_{i3}, w_{i3}) = (10, 10)$ for $i = 1, 2$ (see Figure 2). In this example, the optimal value of LMCK is always equal to capacity c so long as $2 \leq c < 20$. However, that of MCK with $c = 10$ is 3, and with $c = 19$ it is 12. In both cases, the gap between the two reaches up to 7. Moreover, if many additional items were in the lower-right quarter of Figure 2, nothing influenced LMCK.

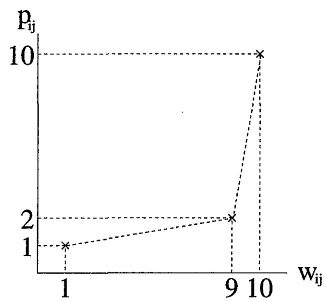

Figure 2: Example 1: the distribution of items in N_1 or N_2

Especially in the case where slopes are constant, LMCK makes no sense. It arises for instance when the profit is equal to the weight for any item, we call an MCK in which case (i.e. $p_{ij} = w_{ij}$ for any i, j) Multiple-Choice Subset-Sum Problem (MCSSP). As easily observed, the LP relaxation applied to MCSSP brings the capacity as the optimal value in the exactly same way as that to a KP in which case. Therefore, the hitherto proposed methods which employ LMCK will show poor performance to MCSSP. Computational experiments of applying a state-of-the-art method to MCSSP are presented in [15].

3 An alternative to LMCK

Since as mentioned in Introduction there exists a linear time algorithm for LMCK, it will not be easy to develop a relaxation problem solvable more efficiently than LMCK. Hence, to attain the title of this section, there will be no alternative but to develop one which gives a bound tighter than LMCK. In this section we construct such a problem.

Throughout this section, in each class, we assume that there exist no two items so that one has the same profit or weight as the other's by Theorem, and all items are sorted in ascending order of its weight, i.e. $w_{i1} < w_{i2} < \cdots < w_{in_i}$ for any $i \in M$.

To begin with we apply a preprocessing to MCK: Suppose that an item of minimum weight (also of minimum profit) in each class has already been packed, and coefficients of remaining $n - m$ items are transformed as follows:

$$\begin{cases} p'_{ij} = p_{ij} - p_{i1} \\ w'_{ij} = w_{ij} - w_{i1}. \end{cases}$$

Accordingly, the capacity diminishes by $\Sigma_{i \in M} w_{i1}$. As a result, the constraints of MCK (3)-(4) are replaced with

$$\sum_{j=2}^{n_i} x_{ij} \leq 1, \quad i \in M$$
$$x_{ij} \in \{0, 1\}, \quad i \in M, \quad j = 2, 3, \ldots, n_i,$$

where $\Sigma_{j=2}^{n_i} x_{ij} = 0$ implies that the 1st item is selected in N_i; otherwise the j-th is, provided $x_{ij} = 1$. Moreover, those of LMCK corresponding to the above are as follows:

$$\begin{aligned} &\sum_{j=2}^{n_i} x_{ij} \leq 1, \quad i \in M \\ &x_{ij} \geq 0, \quad i \in M, \quad j = 2, 3, \ldots, n_i. \end{aligned} \tag{7}$$

This represents LMCK just before augmenting initialized W as described in Section 2.

From a theoretical viewpoint, our aim of a bound tighter than LMCK will be achieved by the optimal value of a problem involving constraints of intermediate strength between the two above, in other words, constraints defining a feasible region which completely includes that of MCK and is strictly included in that of LMCK. In view of LMCK permitting of up to two variables of positive value within some class, ones for our aim are uniquely determined. We hereby formulate a relaxation problem involving the ones:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in M} \sum_{j=2}^{n_i} p'_{ij} x_{ij} + \sum_{i \in M} p_{i1} \\
\text{subject to} \quad & \sum_{i \in M} \sum_{j=2}^{n_i} w'_{ij} x_{ij} \leq c - \sum_{i \in M} w_{i1} \\
& \sum_{j=2}^{n_i} x_{ij} \leq 1, \ \ i \in M \\
& x_{ij} \geq 0, \ \ i \in M, \ \ j = 2, 3, \ldots, n_i \\
& x_{ij} x_{ik} = 0, \ \ i \in M, \ \ 2 \leq j < k \leq n_i.
\end{aligned}$$

We call this one-fractional relaxation problem (1-frp, for short). The name comes from its property, viz., we may assume that its optimal solution includes at most one x_{ij} of value neither 0 nor 1 (Such x_{ij} is a short while said to be fractional). Indeed, when an optimal solution of 1-frp includes two fractional x_{ij}'s, say $\{x_{rs}, x_{tu}\}$, profit-to-weight ratio p'_{rs}/w'_{rs} of an item corresponding to x_{rs} must coincide with that of an item to x_{tu}. Therefore we can modify the solution so that it includes at most one fractional x_{ij}.

Although a solution of 1-frp is an $n-m$ vector, it can readily be transformed back into an original n-vector: For any $i \in M$, $x_{ij} = p/q$ $(j > 1, 0 < p \leq q)$ implies $x_{i1} = (q-p)/q$; otherwise $x_{i1} = 1$, which is interpreted that the value of the x_{ij} indicates the location of a point which divides a line segment joining the 1st and j-th items in N_i in the ratio of $p/q : 1-p/q$ as A depicted in Figure 3. Further, raising the point A vertically up to the A', we will obtain a feasible solution of LMCK due to the convexity of items promising for LMCK, which illustrates 1-frp giving an upper bound not greater than LMCK. In particular, on an MCK with $n_i = 2$ for any $i \in M$, the optimal value of 1-frp coincides with that of LMCK (Also in this case, a problem resulting from the preprocessing being applied to the MCK is a KP).

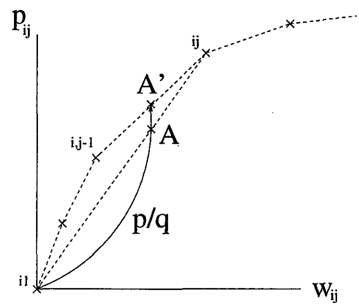

Figure 3: the gap between 1-frp and LMCK

In the following we state three points on 1-frp. In representing a solution appeared in the statement, we omit x_{ij} of value 0 for convenience.

1. The total weight of an optimal solution of LMCK is always equal to the capacity while that of 1-frp is not always so.

   Example 2. Consider an MCK of $m = 2$; Items in each class are the same as $(p_{i1}, w_{i1}) = (2, 1)$, $(p_{i2}, w_{i2}) = (6, 3)$, $(p_{i3}, w_{i3}) = (10, 11)$ for $i = 1, 2$. After the preprocessing, the 2nd and 3rd items in each class remain as in Figure 4. With $c = 16$, the optimal value of LMCK is 17 by a solution, e.g. $x_{13} = 1$, $x_{22} = 3/4$, $x_{23} = 1/4$ while that of 1-frp is 16, since after the choice of the 3rd item in one class, to select the 2nd item of profit 4 is better than two fifths of the 3rd of profit 3.2 in the other despite remaining capacity 2.

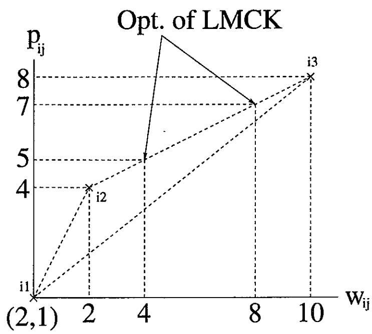

Figure 4: Example 2, $c=14$ or 18 after the preprocessing

Also in the same MCK with $c=20$, that of LMCK is 19 by, e.g. $x_{13}=1$, $x_{22}=1/4$, $x_{23}=3/4$ while 1-frp gives $\lfloor 18.4 \rfloor=18$. On the 1-frp in this case, four fifths of the 3rd item is better than the whole of the 2nd.

2. The 1-frp will be less sensitive to the distribution of items than LMCK. More precisely, there exists an item which is redundant for LMCK but is not for 1-frp.

Example 3. Consider an MCK of $m=2$; Items in each class are the same as $(p_{i1}, w_{i1})=(1, 1)$, $(p_{i2}, w_{i2})=(10, 2)$, $(p_{i3}, w_{i3})=(11, 4)$, $(p_{i4}, w_{i4})=(13, 5)$ for $i=1, 2$ and $c=9$. Items in each class after the preprocessing are in Figure 5; Also the capacity c diminishes by 2. Then, the optimal value of LMCK is 25 by a solution, e.g. $x_{12}=1/3$, $x_{14}=2/3$, $x_{24}=1$ while that of 1-frp is 24 by, e.g. $x_{13}=x_{24}=1$. Thus, the 3rd item validates the point.

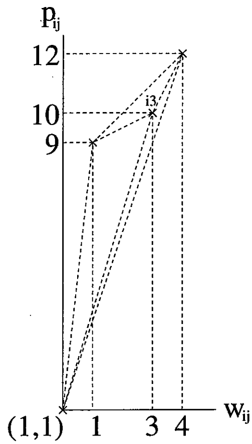

Figure 5: Example 3, $c=7$ after the preprocessing

Here a question arises that which items are screened out for 1-frp besides those by Theorem. It is obvious that the following holds:

Observation. If the j-th and k-th items in N_i satisfy $w_{i1} < w_{ij} < w_{ik}$ and

$$\frac{p_{ij}-p_{i1}}{w_{ij}-w_{i1}} \leq \frac{p_{ik}-p_{i1}}{w_{ik}-w_{i1}}$$

then there exists an optimal solution of 1-frp with $x_{ij}=0$.

Clearly this is obtained by in each class fixing the j-th item to the 1st in the assumption of Proposition.

3. The 1-frp is completely useless for MCSSP with the same behavior as LMCK. On this subject, we suggest that it will be more promising to devise a tailored method by an argument similar to that for KP. The

KP with the condition, that is, the profit is equal to the weight for any item is particularly called Subset-Sum Problem (SSP), and several methods tailored have been proposed so far, e.g. Iida and Vlach [12], Pisinger [16], and Soma and Toth [19]. The term of MCSSP is according to [16], in which an algorithm tailored for MCSSP has actually been proposed.

Then, how should we solve 1-frp? It might seem that after sorting all profit-to-weight ratio of p'_{ij}/w'_{ij} in nonascending order, the 1-frp could be solved with an ordinary greedy approach. However, the one applied to Example 4 below ends in failure.

Example 4. Consider an MCK of $m=2$; Items are $(p_{11}, w_{11})=(1, 1)$, $(p_{12}, w_{12})=(8, 5)$, $(p_{13}, w_{13})=(9, 9)$, $(p_{21}, w_{21})=(1, 1)$, $(p_{22}, w_{22})=(4, 5)$, and $c=10$. After the preprocessing, the items are as seen in Figure 6. A greedy approach applied to the 1-frp gives objective function value 10 by a solution (0, 1, 0), i.e. (0, 0, 1, 1, 0) in original n-vector. However, the greater 12 is gained by an optimal solution (1, 0, 1), i.e. (0, 1, 0, 0, 1), which is the same as that of LMCK.

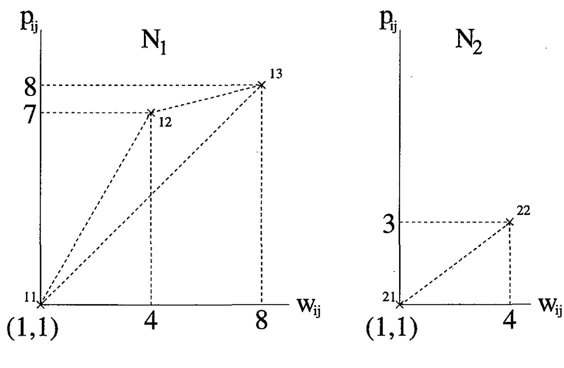

Figure 6: Example 4, $c=8$ after the preprocessing

In fact, except for its indices starting at $j=1$, the same problem as 1-frp has already been examined by Ibaraki et al [10], in which the problem examined is called P. In addition, Section 2 in [10] has already exposed several of characteristics of 1-frp (i.e. P) described above. According to [10], the P is NP-complete and there will exist no linear time algorithm for it, which is also suspected from that in the first half of Example 2 the total weight of the optimal solution of 1-frp does not reach the capacity.

Notes. On $\bar{P}$ a relaxation problem introduced in [10] for P: In the formulation of P, constraints not involving the capacity are the two below:

$$0 \leq x_{ij} \leq 1, \quad i \in M, \quad j \in N_i, \tag{8}$$

$$\text{At most one of } x_{i1}, x_{i2}, \ldots, x_{in_i} \text{ is positive, for } i \in M. \tag{9}$$

The latter half of (8), i.e. $x_{ij} \leq 1$, can be replaced with $\Sigma_{j \in N_i} x_{ij} \leq 1$, since under the first half of (8) and (9) it follows that $x_{ij} \leq \Sigma_{j \in N_i} x_{ij} = \max_{j \in N_i} x_{ij}$. Then, excluding (9) from P results in $\bar{P}$ involving $x_{ij} \geq 0$ and $\Sigma_{j \in N_i} x_{ij} \leq 1$ remained, which are almost the same as (7), i.e. those of LMCK transformed. Being different from LMCK, an optimal solution of $\bar{P}$ may include just only one variable of not integer value due to not $\Sigma_{j \in N_i} x_{ij} = 1$ but ≤ 1, also due to which $\bar{P}$ (and P too) does not require the first half of (5).

In general, a slightly loose but quickly obtainable bound is more practical than that of tight but expensive, viz. of much cost to compute it (see, e.g. Iida [11]). A branch-and-bound algorithm proposed in [10] for P solves $\bar{P}$ on each subproblem spawned, however, $\bar{P}$ is almost the same as LMCK. In light of this although it is unknown in comparison with LMCK how much tight a bound by 1-frp is, yet we should conclude that 1-frp is expensive and quite unsuitable for MCK as a relaxation problem.

At the beginning of this section we stated that a relaxation problem for MCK being an alternative to LMCK will be the problem giving a bound tighter than LMCK. Nevertheless as we have seen in this section it seems that there exists no practical one between MCK and LMCK, or more specifically, between the former with no variable of not integer value in an optimal solution and the latter with at most two such variables in that.

References

[1] V. Aggarwal, N. Deo, and D. Sarkar, "The knapsack problem with disjoint multiple-choice constraints," *Naval Res. Logistics* **39** (2) 213–227 (1992).

[2] R. D. Armstrong, D. S. Kung, P. Sinha, and A. A. Zoltners, "A computational study of a multiple-choice knapsack algorithm," *ACM Transactions on Math. Software* **9** (2) 184–198 (1983).

[3] K. Dudziński and S. Walukiewicz, "A fast algorithm for the linear multiple-choice knapsack problem," *Operations Res. Letters* **3** (4) 205–209 (Oct. 1984).

[4] K. Dudziński and S. Walukiewicz, "Exact methods for the knapsack problem and its generalizations," *European J. Operational Res.* **28** (1) 3–21 (1987).

[5] M. E. Dyer, "An O(n) algorithm for the multiple-choice knapsack linear program," *Math. Programming* **29** (1) 57–63 (May 1984).

[6] M. E. Dyer, N. Kayal, and J. Walker, "A branch and bound algorithm for solving the multiple-choice knapsack problem," *J. Computational and Applied Math.* **11**, 231–249 (1984).

[7] M. E. Dyer, W. O. Riha, and J. Walker, "A hybrid dynamic programming/branch-and-bound algorithm for the multiple-choice knapsack problem," *J. Computational and Applied Math.* **58** (1) 43–54 (March 1995).

[8] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management Science* **27**(1) 1–18 (1981).

[9] F. Glover and D. Klingman, "A $o(n \log n)$ algorithm for LP knapsacks with GUB constraints," *Math. Programming* **17**(3) 345–361 (1979).

[10] T. Ibaraki, T. Hasegawa, K. Teranaka, and J. Iwase, "The multiple-choice knapsack problem," *J. Operations Res. Soc. Japan* **21**(1) 59–95 (1978).

[11] H. Iida, "A note on the max-min 0-1 knapsack problem," *J. Combinatorial Optimization* **3**(1) 89–94 (1999).

[12] H. Iida and M. Vlach, "An exact algorithm for the subset-sum problem," in: *Cooperative Research Report 92 Optimization — Modeling and Algorithms —*, **9**, 11–29 (The Institute of Statistical Mathematics, Tokyo 106-8569, Japan, 1996).

[13] S. Martello, D. Pisinger, and P. Toth, "New trends in exact algorithms for the 0–1 knapsack problem," *European J. Operational Res.* **123**(2) 325–332 (2000).

[14] R. M. Nauss, "The 0-1 knapsack problem with multiple choice constraints," *European J. Operational Res.* **2**(2) 125–131 (1978).

[15] D. Pisinger, "A minimal algorithm for the multiple-choice knapsack problem," *European J. Operational Res.* **83**(2) 394–410 (June 1995).

[16] D. Pisinger, "Linear time algorithms for knapsack problems with bounded weights," *J. Algorithms* **33**(1) 1–14 (1999).

[17] D. Pisinger, "Budgeting with bounded multiple-choice constraints," *European J. Operational Res.* **129**(3) 471–480 (2001).

[18] P. Sinha and A. A. Zoltners, "The multiple-choice knapsack problem," *Operations Res.* **27**(3) 503–515 (1979).

[19] N. Y. Soma and P. Toth, "An exact algorithm for the subset sum problem," *European J. Operational Res.* **136**(1) 57–66 (2002).

[20] E. Zemel, "The linear multiple choice knapsack problem," *Operations Res.* **28**(6) 1412–1423 (1980).

[21] E. Zemel, "An O(n) algorithm for the linear multiple choice knapsack problem and related problems," *Information Processing Letters* **18**(3) 123-128 (March 1984).