Some Properties of 0-1 Knapsack Problems

by

Yao-Nan Lien

Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277.
(614) 292-5236
lien@ohio-state.arpa
cbosgd!osu-eddie!lien

February 28, 1987

OSU-CISRC-2/87-TR6

1. INTRODUCTION

This article explores the properties of some *0-1 knapsack problems.* The standard 0-1 knapsack problem is as follows: Given n pairs of positive integers, (p_i, w_i) and a positive integer c, find $x_1, x_2, \ldots, x_n$ so as to

$$\text{maximize} \quad \sum_{i=1}^{n} p_i x_i, \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq c, \; x_i \in \{0,1\}.$$

We may consider of i as an indexed *item*, with associated *profit* p_i and *weight* w_i. The problem is to allocate a subset of these items into a knapsack with a *capacity* c such that the total profit is maximized. The *profit-density* of an item i is the ratio of its profit to its weight $(p_i \,/\, w_i)$. The *maximum-profit* (P^*) of a knapsack problem instance is defined as the total profit of an optimal 0-1 knapsack allocation. There is no polynomial-time optimal algorithm to solve the 0-1 knapsack problem. However, it does have pseudo polynomial-time optimal algorithms and fully polynomial-time heuristic algorithms [3]. In the *0-1 multiple-choice-knapsack problem* [1, 8], some groups of items are to be allocated to a knapsack, and at most one item from each group can be allocated to the knapsack. In the *0-1 multiple-knapsack problem,* some categories of items are to be allocated to a set of knapsacks. These problems have been studied extensively for more than 30 years [5, 2, 4, 6, 3, 7].

In this article, the correlation between the maximum-profit and some parameters is investigated for the standard 0-1 knapsack problem. The results are extended to other versions of the knapsack problem. There is no indication having been noticed that the maximum-profit of a particular instance absolutely depends on any particular parameter. Instead, the difference in the maximum-profits of two or more instances, which is referred to as the *profit-difference,* is easier to estimate in terms of the difference in some parameters. The profit-difference is estimated for the following conditions:

P1. When the same set of items is to be allocated to two knapsacks with different capacities. Three variations are studied:

(a) Either the capacities, the total occupied weight of the optimal solutions, or the inclusive relationship between the sets of items is known.

simple heuristic algorithm to obtain approximate solutions for the file allocation problem is as follows.

Step 1. To allocate the first copy of all files arbitrarily to get an initial feasible solution.

Step 2. To improve the solution by allocating more files to each site independently as a standard knapsack problem.

The performance estimation of such a heuristic algorithm, i.e., how close to the optimal solution an approximate solution is, relies on the knowledge of the largest possible profit-difference between the knapsack in the original capacity and the knapsack in the reduced capacity. The property shown in Theorem 2 is useful in this case.

A very interesting situation arises when the profits and/or the weights of items are not constants. For instance, in the file allocation problem, the "profit" of allocating a file to a site may be dependent on the number of copies being allocated to the system. One simple way to cope with the problem in a heuristic algorithm is to estimate the profit before the allocation, and then to solve the problem. In some cases, the problem can be modeled as a *0-1 multiple-choice-knapsack problem,* in which more than one choice is available when allocating an item to the knapsack. The problem is "harder" than the standard 0-1 knapsack problem. The properties shown in Theorem 3 is useful to evaluate the heuristic algorithms of this problem.

This article is organized as follows: the notations used in this article are introduced in the remaining part of this section; the properties of the standard 0-1 knapsack problem and the multiple-choice-knapsack problem are discussed in Section 2 and Section 3; as an example, heuristic algorithms with guaranteed performance for some special cases of the 0-1 multiple-knapsack problem are shown in Section 4; and a concluding remark is given in Section 5.

The following notations are defined for the standard 0-1 knapsack problem.

w_i the weight of an item i;

p_i the profit of an item i if it is allocated to a knapsack;

x_i the allocation of item i in a knapsack,

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is allocated to the knapsack} \\ 0 & \text{if item } i \text{ is not allocated to the knapsack;} \end{cases}$$

(a) $c_1 \geq c_2$,

(b) $W(\text{Å}(c_1,I)) \geq c_2$,

(c) $c_1 \geq W(\text{Å}(c_2,I))$,

(d) $W(\text{Å}(c_1,I)) \geq W(\text{Å}(c_2,I))$;

and $P^*_{c_1} = P^*_{c_2}$, when the equality in any condition holds.

Proof:

(a) Prove by contradiction. Suppose there exists such two instances that $P^*_{c_1} < P^*_{c_2}$ and $c_1 \geq c_2$, we can always allocate all items that are allocated to the smaller knapsack to the larger knapsack. The total profit is $P^*_{c_2}$, which is in turn greater than $P^*_{c_1}$. This contradicts to the assumption that $P^*_{c_1}$ is the maximum-profit of the instance (c_1,I). Similarly, the theorem can be proved for conditions (b), (c), or (d).

If any equality condition in condistions (a), (b), (c), and (d) holds, both $P^*_{c_1} \geq P^*_{c_2}$ and $P^*_{c_1} \leq P^*_{c_2}$ must hold. Therefore, $P^*_{c_1} = P^*_{c_2}$.

□

Corollary 1.1:
Given a problem instance (c,I) and a set A which is a subset of I,

$P^*_c \geq P(A)$, if $c \geq W(A)$.

Proof: If this is false, we can hypothetically create an instance $(W(A),A)$ whose maximum-profit, $P(\text{Å}(W(A),A))$, is $P(A)$. From Lemma 1.1, we can get $P^*_c \geq P(\text{Å}(W(A),A)) = P(A)$. This contradicts to the assumption that P^*_c is smaller than $P(A)$.

□

Lemma 1.2 concerns the profit-difference between two problem instances when the items to be allocated in one instance is a proper subset of that in the other instance.

Proof: If there exists such an optimal allocation $A(c,I)$ such that $W(A(c,I)) < c - k$, we can always allocate into the knapsack another arbitrary item which is not in the knapsack to increase the total profit since the weight of any item is no greater than k. This contradicts to the assumption that $W(A(c,I))$ is the total weight of any optimal allocation.

□

The profit-density is a very important parameter in the standard 0-1 knapsack problem. Without the integer constraints, the problem can be solved in polynomial time by simply allocating the items in the descending order of profit-density and allocating the portion of the last item that fulfills the knapsack. This algorithm is also a good heuristic if the problem has integer constraints [3]. It is interesting to know the correlation between the profit-density and the maximum-profit. Lemma 1.4 states some properties relevant to the profit-density and is used in the proofs of some theorems in this section.

Lemma 1.4:

For any set of arbitrary positive numbers $p_1, p_2, \ldots, p_n, w_1, w_2, \ldots, w_n$, if

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq ,\ldots, \geq \frac{p_j}{w_j} \geq \frac{p_{j+1}}{w_{j+1}} \geq , \ldots , \geq \frac{p_n}{w_n},$$

then

(a) $\sum_{i=1}^{j} p_i \geq \sum_{i=j+1}^{n} p_i$, if $\sum_{i=1}^{j} w_i \geq \sum_{i=j+1}^{n} w_i$, and

(b) $$\sum_{i=1}^{j} p_i \geq \left(\frac{\sum_{i=1}^{j} w_i}{\sum_{i=j+1}^{n} w_i} \right) \sum_{i=j+1}^{n} p_i .$$

Proof:

(a) Since $p_i \geq \frac{p_{j+1}}{w_{j+1}} w_i$ and $\frac{p_{j+1}}{w_{j+1}} w_k \geq p_k$, $(i = 1, \ldots, j)$ and $(k = j+1, \ldots, n)$,

$$\sum_{i=1}^{j} p_i \geq \frac{p_{j+1}}{w_{j+1}} \sum_{i=1}^{j} w_i \geq \frac{p_{j+1}}{w_{j+1}} \sum_{k=j+1}^{n} w_k = \sum_{k=j+1}^{n} \frac{p_{j+1}}{w_{j+1}} w_k \geq \sum_{k=j+1}^{n} p_k .$$

Proof:

(a) Refer to the illustration in Figure 1.

(a.1) Assume $\AA(c_1,I)=\{1,2,..,\ell\}$ is an optimal 0-1 knapsack allocation for the instance (c_1,I), we can find j such that

$(c_1-\Delta c-k) \le \sum_{i=1}^{j} w_i \le (c_1-\Delta c)$ since the weight of each item is no greater than k. We denote I_j as the set of items $\{1,2, \ldots, j\}$. and $\overline{I_j}$ as $(\AA(c_1,I)-I_j) = \{j+1, \ldots, \ell\}$.

(a.2) Since $W(I_j) \le (c_1-\Delta c)$ and $c_2 = (c_1-\Delta c)$, $c_2 \ge W(I_j)$.
From Lemma 1.1, we know $P^*_{c_2} = P(\AA(c_2,I)) \ge P(I_j)$.

(a.3) Since $W(I_j) \ge (c_1 - \Delta c - k)$,
$W(\overline{I_j}) = W(\AA(c_1,I)) - W(I_j) \le (\Delta c + k)$.

(a.4) From (a.3) and Lemma 1.2, we can get
$P(\AA(c_1,I)-I_j) = P(\overline{I_j}) = P(\AA(\Delta c+k\ ,\ \overline{I_j})) \le P(\AA(\Delta c+k,I))$.

(a.5) Finally, from (a.2) and (a.4), we get
$P^*_{c_2} + P(\AA(\Delta c+k,I) \ge P(\AA(c_1,I)-I_j)+P(I_j)=P(\AA(c_1,I))=P^*_{c_1}$.

(b) Similar to part (a), we assume $\AA(c_1,I)=\{1,2,..,\ell\}$ is an optimal 0-1 knapsack allocation for the instance (c_1,I), and the items are in the descending order of profit-density, i.e., $\frac{p_1}{w_1} \ge \frac{p_2}{w_2} \ge, \ldots, \frac{p_\ell}{w_\ell}$. We denote I_j and $\overline{I_j}$ as in part (a).

(b.1) We first prove that $P^*_{c_2} \ge \left\lfloor \frac{c_1}{\Delta c+k} - 1 \right\rfloor P(\overline{I_j})$. Denote z as

$\left\lfloor \frac{c_1}{\Delta c+k} - 1 \right\rfloor$. Since $W(\overline{I_j}) \le \Delta c+k$, from Lemma 1.4(b), we know

$$P^*_{c_2} \ge P(I_j) \ge \frac{c_1-\Delta c-k}{W(\overline{I_j})} P(\overline{I_j}) \ge \frac{c_1-\Delta c-k}{\Delta c+k} P(\overline{I_j}) = zP(\overline{I_j})$$

Therefore, we get $P^*_{c_2} \ge zP(\overline{I_j})$.

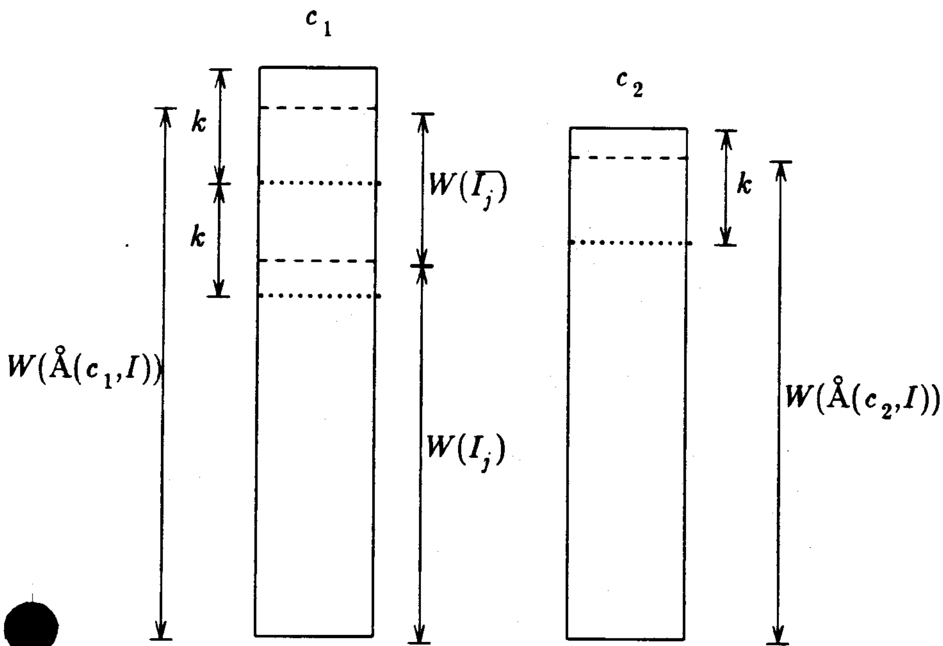

Figure 1 Illustration of the proof of Theorem 1, ($\Delta c = k$).

reduction. In Theorem 2, $\mathring{A}(c - W(Q),\ I - Q)$ is an optimal solution for allocating the remaining items to the remaining capacity when the set of items Q are already allocated in the knapsack, while $\mathring{A}((W(Q)+k,\ I)$ is an optimal solution of allocating the set of items I to a knapsack with a capacity $W(Q)+k$.

Theorem 2:

Given a knapsack instance (c, I), a set of items Q which is a subset of I, and a constant k, then (a.2.4) From (a.2.2) and (a.2.3), we can get

$$P^*_{c_2} = P(\mathring{A}(c_2, I-Q)) + P(Q) \geq P(I_j - Q) + P(Q \cap I_j) = P(I_j).$$

(a.3) Similar to Steps (a.3), (a.4), and (a.5) in Theorem 1, we get

$$P^*_{c_2} + P(\mathring{A}(W(Q)+k, I) \geq P^*$$

(b) Since $W(I_j) \geq (c - W(Q) - k)$ and $W(\bar{I_j}) \leq W(Q)$, we get

$$P^*_{c_2} \geq P(I_j) \geq \frac{W(I_j)}{W(\bar{I_j})} P(\bar{I_j}) \geq \frac{c - W(Q) - k}{W(Q)+k} P(\bar{I_j}) = \left(\frac{c}{W(Q)+k} - 1 \right) P(\bar{I_j})$$

from Lemma 1.4(b). Similar to Steps (b.2), (b.3), and (b.4) in Theorem 1, we can get $P^*_{c_2} \geq \left(1 - \frac{W(Q)+k}{c} \right) P^*$.

(c) Similar to the proof in Theorem 1(c), we can get the approximation.

□

Again, Theorem 2(a) provides smaller bound than 2(b) does, but the deviation shown in 2(b) is more convenient. The approximation in 2(c) is even more convenient.

Summary

In this section, the maximum-profits of two knapsack problem instances are compared. In most cases, the same set of items is to be allocated to the knapsacks with different capacities. Some upper bounds of the profit-difference are presented. The results are extended to the case when some items are already allocated in the knapsack. These properties are very useful to improve some optimal algorithms and to design good heuristic solutions. It can also be used in the evaluation of heuristic algorithms. An interesting example is shown in Section 4.

3. TO ALLOCATE DIFFERENT ITEM SETS TO THE SAME KNAPSACK

In this section, the properties of allocating different sets of items to the same knapsack are to be investigated. As mentioned in Section 1, these properties are useful to evaluate the heuristic algorithms for the multiple-choice-knapsack problem. We assume there is a one-to-one correspondence between all sets of items, i.e., all sets have the same number of items.

Proof: Denote $\mathring{A}_{I_1} = \mathring{A}(c, I_1)$ and $\mathring{A}_{I_2} = \mathring{A}(c, I_2)$. Suppose $\mathring{A}_{I_1}$ is an optimal allocation of instance (c, I_1), we can allocate $f_{12}(\mathring{A}_{I_1})$ to the knapsack such that the profit is $P^*_{I_1} - \sum_{i \in \mathring{A}_{I_1}} (p_{i_1} - p_{i_2}) \geq P^*_{I_1} - \sum_{i \in \mathring{A}_{I_1}} \mathcal{D}\,(p_{i_1} - p_{i_2}) \geq P^*_{I_1} - \sum_{i=1}^{n} \mathcal{D}\,(p_{i_1} - p_{i_2})$.

Therefore, $P^*_{I_2} + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_1} - p_{i_2}) \geq P^*_{I_1}$.

$\square$

Theorem 3 is useful to answer the following question: When allocating the items from one set according to the optimal solution of allocating another set of items to the knapsack, what would be the possible profit reduction as compared to the real optimal solution?

Theorem 3:

Given two knapsack problem instances (c, I_1) and (c, I_2),

(a) $P(f_{21}(\mathring{A}(c, I_2))) + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_2} - p_{i_1}) \geq P^*_{I_2}$,

(b) $P(f_{21}(\mathring{A}(c, I_2))) + \sum_{i=1}^{n} |p_{i_1} - p_{i_2}| \geq P^*_{I_1}$,

if $|I_1| = |I_2| = n$, and $w_{i_1} = w_{i_2}$, $i = 1, 2, \ldots, n$.

Proof: Denote $\mathring{A}_{I_2} = \mathring{A}(c, I_2)$.

(a) Since $w_{i_1} = w_{i_2}$, $i = 1, 2, \ldots, n$, $f_{21}(\mathring{A}_{I_2})$ is a feasible solution of (c, I_1). It is easy to see that

$$P(f_{21}(\mathring{A}_{I_2})) + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_2} - p_{i_1}) \geq P(f_{21}(\mathring{A}_{I_2})) + \sum_{i_2 \in \mathring{A}_{I_2}} \mathcal{D}\,(p_{i_2} - p_{i_1}) \geq P^*_{I_2}.$$

(b) From Lemma 3.2, we know $P^*_{I_2} + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_1} - p_{i_2}) \geq P^*_{I_1}$

Therefore, deriving from (a),

$$P(f_{21}(\mathring{A}_{I_2})) + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_1} - p_{i_2}) + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_2} - p_{i_1}) = P(f_{21}(\mathring{A}_{I_2})) + \sum_{i=1}^{n} |p_{i_2} - p_{i_1}| \geq P^*_{I_1}.$$

$\square$

$$\text{maximize} \quad \sum_{\ell=1}^{m} \sum_{i=1}^{n} p_{\ell i} x_{\ell i}, \tag{2}$$

$$\text{subject to} \quad \sum_{i=1}^{n} w_i x_{\ell i} \le c_\ell, \quad \forall \; \ell,$$

$$x_{\ell i} \in \{0,1\}, \text{ and}$$

$$\sum_{\ell=1}^{m} x_{\ell i} \ge 1, \forall \; i.$$

We may consider that some categories of items are to be allocated to m knapsacks with capacities $c_1, c_2, \ldots, c_m$. All items in a category are identical. A profit of $p_{\ell i}$ is produced if one item of category i is allocated to knapsack ℓ. For convenience, an item of category i in knapsack ℓ is referred to as item ℓi. A knapsack can only be allocated at most one item from a category, and at least one item from each category must be allocated to any knapsack. The problem can be easily proved NP-hard, so fast heuristic solutions as well as performance evaluations are needed. A heuristic algorithm whose performance can be estimated using theorems developed in Section 2 and 3 is as follows.

Heuristic Algorithm MK-HEUR

1. An item from each category is allocated to an arbitrary knapsack, such that the total weight of all items in any knapsack is no greater than $\lceil \frac{n}{m} \rceil k$, where k is the maximum weight of all items.

2. For each knapsack, allocate more items using standard 0-1 knapsack algorithms for the remaining capacity.

□

For simplicity, we assume that the capacity of every knapsack is no less than $\lceil \frac{n}{m} \rceil k$. Otherwise, Step 1 as well as the error estimations must be modified. Under this assumption, Step 1 is always feasible since each knapsack can hold at least $\lceil \frac{n}{m} \rceil$ items and at least ($\lceil \frac{n}{m} \rceil m \ge n$) items can be allocated. In step 2, either optimal or heuristic 0-1 knapsack algorithms can be used. Since there exist

$MA(C,I)$ a 0-1 multiple-knapsack allocation for an instance (C,I);

$M\mathring{A}(C,I)$ an optimal 0-1 multiple-knapsack allocation for an instance (C,I);

The following notations are special for MCMK-HEUR:

T the number of choices of each item;

$MCMA(C,I)$ a 0-1 multiple-choice-multiple-knapsack allocation for an instance (C,I);

$MCM\mathring{A}(C,I)$ an optimal 0-1 multiple-choice-multiple-knapsack allocation for an instance (C,I);

$\wp i_0$ The most profitable choice of item $\wp i$, $(p_{\wp i_0} = \max_j p_{\wp i_j})$;

$I_\wp^0$ the set of items composed of the most profitable choices of items in $I_\wp$, $I_\wp^0 = \{\wp i_0 \mid i = 1,2, \ldots, n\}$.

$\overline{p_{\wp i}}$ the difference between the most profitable choice and the least profitable choice of an item $\wp i$, $\{\overline{p_{\wp i}} = p_{\wp i_0} - \min_j p_{\wp i_j}\}$;

To evaluate algorithm MK-HEUR, we have to know the profit reduction due to the preallocation in Step 1. It is difficult to compare the heuristic solutions to the optimal solutions directly. Rather, we compare the profit generated from each site in algorithm MK-HEUR to the maximum profit that the site can generate if it is solved as a standard 0-1 knapsack problem. The total difference is larger than the difference between the heuristic solutions and the optimal solutions. Thus, it serves as an upper bound, which is shown in Lemma 4.1. For convenience, we refer to the 0-1 knapsack allocation for all knapsacks as a *many-knapsack allocation*. In this case, the maximum-profit of a many-knapsack allocation is $\sum_{\wp=1}^{m} P(\mathring{A}(c_\wp, I_\wp))$.

Lemma 4.1:

The maximum-profit of a 0-1 multiple-knapsack problem is less than or equal to the maximum-profits of a many-knapsack allocation, i.e.,

$$P^* = P(M\mathring{A}(C,I)) \leq \sum_{\wp=1}^{m} P(\mathring{A}(c_\wp, I_\wp)).$$

(a.2) Denote $P(\mathring{A}(c_\ell, I_\ell))$ as P_ℓ^*. Since $W(Q_\ell) \leq |Q_\ell| k$, from Theorem 2(a), we get

$$P(\mathring{A}(c_\ell - W(Q_\ell)\,,\, I_\ell - Q_\ell)) + P(Q_\ell) + P(\mathring{A}((|Q_\ell|+1)k, I_\ell)) \geq P_\ell^*, \ \forall\, \ell.$$

Further, $P(\mathring{A}((1+\lceil \frac{n}{m} \rceil)k\,,\, I_\ell)) \geq P(\mathring{A}((|Q_\ell|+1)k, I_\ell))$ since

$|Q_\ell| \leq \lceil \frac{n}{m} \rceil$,

so, $P(\mathring{A}(c_\ell - W(Q_\ell)\,,\, I_\ell - Q_\ell)) + P(Q_\ell) + P(\mathring{A}((1+\lceil \frac{n}{m} \rceil)k\,,\, I_\ell)) \geq P_\ell^*$.

(a.3) From (a.2) and Lemma 4.1, we get $P(H) + \sum_{\ell=1}^{m} P(\mathring{A}((1+\lceil \frac{n}{m} \rceil)k\,,\, I_\ell))$

$$= \sum_{\ell=1}^{m} (P(\mathring{A}(c_\ell - W(Q_\ell)\,,\, I_\ell - Q_\ell)) + P(Q_\ell)) + \sum_{\ell=1}^{m} P(\mathring{A}((1+\lceil \tfrac{n}{m} \rceil)k\,,\, I_\ell))$$

$\geq \sum_{\ell=1}^{m} P_\ell^* \geq P^*$. Therefore,

$$P(H) + \sum_{\ell=1}^{m} P(\mathring{A}((1+\lceil \tfrac{n}{m} \rceil)k\,,\, I_\ell)) \geq P^*.$$

(b)

From Theorem 2, we know $P(H) \geq \sum_{\ell=1}^{m} \left(1 - \frac{W(Q_\ell)+k}{C_\ell}\right) P_\ell^* \geq \sum_{\ell=1}^{m} (1-\epsilon_\ell) P_\ell^*$.

Then, it is easy to prove that $P(H) \geq P^* \left(1 - \sum_{\ell=1}^{m} \left(\epsilon_\ell \frac{P_\ell^*}{P^*} \right) \right)$.

□

Unlike Theorem 1(b) and 2(b), the deviation shown in Theorem 4(b) depends not only on q, k, and c_ℓ, but also on the maximum-profit of some knapsack problem instances (P_ℓ^*). Intuitively, we can imagine that the total deviation is the sum of the deviation of each knapsack weighed by the maximum-profit of the knapsack. Unless we know the maximum-profit of all knapsacks, it is impossible to know the overall deviation. Therefore, Theorem 4 is almost useless in this case. However, it is still useful if the maximum profit of each knapsack can be estimated. For example, if the maximum-profits of all knapsacks are approximately the same, the deviation is no greater than $\left(\sum_{\ell=1}^{m} \epsilon_\ell \right) / \ell$.

3.

These results are very useful to design good optimal and heuristic algorithms as well as to estimate the performances of the heuristic algorithms of many NP-hard problems. In Section 4, the performances of heuristic solutions for some special 0-1 multiple-knapsack problems are estimated based on the results in Section 2 and 3. Since many resource allocation problems can be modeled as some forms of the knapsack problem, the results in this article will be very valuable for them.

ACKNOWLEDGMENT

The author wishes to thank Professor Benjamin Wah of University of Illinois, Professor Sartaj Sahni of University of Minnesota, Professor Steve Lai, Professor Dik Lee, and Professor Yih-Long Chang of the Ohio State University for their valuable comments and advice.

REFERENCE

1. Armstrong, Ronald D., Prabhakant Sinha, and Andris A. Zoltners, "The Multiple-Choice Nested Knapsack Model," *Management Science*, vol. 28, pp. 34-43, Jan. 1982.
2. Chandra, Ashok K., D.S. Hirschberg, and C. K. Wong, "Approximate Algorithms For Some Generalized Knapsack Problems," *Theoretical Computer Science*, vol. 3, pp. 293-304, 1976.
3. Garey, Michael R. and David S. Johnson, in *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, Reading, England, 1979.
4. Ibarra, O. H. and C. E. Kim, "Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems.," *JACM*, vol. 22, pp. 463-468, 1975.
5. Lawler, Eugene L., "Fast Approximation Algorithms For Knapsack Problems," *Math. of Operations Research*, vol. 4, no. 4, pp. 339-356, Nov. 1979.
6. Sahni, S., "Approximate Algorithms for the 0/1 Knapsack Problem," *JACM*, vol. 22, pp. 115-124, 1975.
7. Salkin, Harvey M. and Cornelis A. de Kluyver, "The Knapsack Problem: A Survey," *Naval Research Logistics Quartly*, vol. 22, pp. 127-144, 1975.

Some Properties of 0-1 Knapsack Problems

by

Yao-Nan Lien

Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277.

ABSTRACT

The properties of some 0-1 knapsack problems are presented. The maximum-profits, which are the total profits of the optimal solutions, of different problem instances are compared. When the same set of items is to be allocated to two knapsacks with capacities c_1 and c_2 $(c_1 \geq c_2)$ in the standard 0-1 knapsack problem, the ratio of the difference in their maximum-profit to the maximum-profit of the larger knapsack will never be greater than $(c_1 - c_2 + k) / c_1$, where k is the maximum weight of the items. Approximately, the ratio of the maximum-profit of the smaller knapsack to that of the larger knapsack will be greater than the ratio of their capacities if $c_1, c_2 >> k$. Similar results are obtained when a subset of the items are already allocated into a knapsack before the allocation. Finally, the performances of some heuristic algorithms of the 0-1 multiple-choice-knapsack problem and the 0-1 multiple-knapsack problem are estimated based on these results.

INDEX TERMS: Knapsack problem, heuristic, integer programming, multiple knapsack problem, multiple-choice-knapsack problem.

(b) The weights of items are not bounded.

(c) The weights of items are bounded by a constant value smaller than the capacities of the knapsacks.

P2. One set of items to be allocated into an empty knapsack and into the same knapsack except that a subset of the items is preallocated in the knapsack. The weight of items are bounded too.

P3. Two different sets of items are to be allocated to a knapsack (two knapsacks with the same capacity).

P4. The profits of items are non-deterministic before the allocation.

The knapsack problems are important because many resource allocation problems can be modeled, in one way or another, as a knapsack problem. One such example is the *file allocation problem* [9], which is to distribute possibly replicated files to a distributed computer system so that the overall operating cost is minimized. The underlining model of this problem is very close to a special case of 0-1 multiple-knapsack problem under some special constraints.

As a result, knowing the properties of these knapsack problems is important for the search and evaluation of solutions. Searching for the optimal solutions of these problems is time consuming since most of the problems are NP-hard. The solutions are very likely obtained by some form of exhaustive search such as the branch-and-bound search. The speed of these optimal algorithms is heavily dependent on the ability to identify those partial solutions that may approach an optimal solution quickly and those that do not lead to an optimal solution. Unfortunately, this ability is very much dependent on the problems to be solved so that very few general rules can be adopted. Thus, the knowledge on the properties of these problems is very useful to increase the speed of these searching algorithms. By knowing the properties of a problem, a searching strategy may be able to choose shorter paths toward an optimal solution. For example, to solve the standard 0-1 knapsack problem using branch-and-bound search, the property of condition P2 can be used to estimate the largest possible profit for a partial solution and to decide whether to further expand it or not. Moreover, it may be possible to convert the knowledge of a problem into good heuristic algorithms. For example, Theorem 2 in this article shows that to allocate items in the descending order of profit-density may lead to good approximate solutions for the standard 0-1 knapsack problem (this is a well known fact.) Finally, the properties of these problems may be very helpful in the evaluation of heuristic algorithms. For example, a

(c,I) a knapsack problem instance that the set of items I is to be allocated to a knapsack with a capacity c;

$A(c,I)$ a knapsack allocation for an instance (c,I), i.e. $A=\{i \mid x_i=1, i \in I\}$;

$\mathring{A}(c,I)$ an optimal 0-1 knapsack allocation for an instance (c,I);

$P(A)$ the total profit of the set of items A, $P(A)=\sum_{i \in A} p_i$;

$W(A)$ the total weight of the set of items A, $W(A)=\sum_{i \in A} w_i$.

The knapsacks concerned in this article are indistinguishable unless their capacities are different. Although the capacity of a knapsack may be greater than or equal to the total weight of the items in an instance, we generally assume that the capacity of the knapsack is smaller than the total weight of all items in most instances.

2. TO ALLOCATE ONE ITEM SET TO DIFFERENT KNAPSACKS

In this section, the properties of allocating the same set of items to two knapsacks with different capacities are to be studied. Two instances are compared based on the knapsack capacities or the total weights of optimal solutions in Lemma 1.1 and 1.2. The correlation between the profit-difference and the capacity-difference when the weights of items are not bounded is discussed and is shown in Theorem 1. Similar results are obtained in Theorem 2 when some items are preallocated in the knapsack.

Basic Properties

Lemma 1.1 compares the maximum-profits of two problem instances when either the capacity or the total weight of an optimal allocation of each instance to be compared is known. Recall that $\mathring{A}(c,I)$ is an optimal allocation of instance (c,I) and $W(\mathring{A}(c,I))$ is the total weight of $\mathring{A}(c,I)$. We denote

$$P_c^* = P(\mathring{A}(c,I)),\ P_{c_1}^* = P(\mathring{A}(c_1,I)), \text{ and } P_{c_2}^* = P(\mathring{A}(c_2,I)).$$

Lemma 1.1:

Given two problem instances, (c_1,I) and (c_2,I), and arbitrary optimum allocations $\mathring{A}(c_1,I)$, $\mathring{A}(c_2,I)$,

$P_{c_1}^* \geq P_{c_2}^*$, if any of the following conditions holds:

Lemma 1.2:

Given two problem instances (c_1, I_1) and (c_2, I_2),

$P^*_{c_1} \geq P^*_{c_2}$, if $c_1 \geq c_2$ and $I_2 \subset I_1$.

Proof: We can easily prove it by contradiction in a way similar to the proof in Lemma 1.1.

□

Knapsacks with Weight-unbounded Items

We are comparing the maximum-profits when nothing is known except the capacities of the knapsacks being compared. Unfortunately, the profit-difference can be as high as the maximum-profit of the larger knapsack no matter how small the difference of their capacities is. This can be easily seen by hypothetically creating a set of items whose weights are all greater than the capacity of the smaller knapsack and smaller than that of the larger knapsack.

This property provides little help to our intended objectives since there is only a small correlation between the profit-difference and the knapsack capacities. However, these extreme cases may not be very common in the real world and are very likely manually solvable. In many cases, this correlation is much larger when the weights of items are bounded. The remaining part of this section is to examine the problem when this condition holds. Smaller upper bounds can be found in some cases.

Knapsacks with Weight-bounded Items

The main reason that the profit-difference between two instances may be very large is because the weights of items are not bounded. Since the knapsack capacities in many real world problems are much larger than the weights of items, it is reasonable to examine the cases when the weights of items are bounded by some constants. In the following theorems, we assume that the weights of all items in an instance are bounded by a constant value smaller than the knapsack capacity.

Lemma 1.3:

Given a knapsack with a capacity c and a positive integer k $(k \leq c)$, if there is no item whose weight is greater than k, then the total weight in an optimal 0-1 knapsack allocation is greater than or equal to $(c-k)$, i.e.,

$W(\text{Å}(c, I)) \geq c - k$, if $(w_i \leq k, \forall\ i \in I)$.

(b) As in (a), since $\sum_{i=1}^{j} p_i \geq \frac{p_{j+1}}{w_{j+1}} \sum_{i=1}^{j} w_i$ and $\frac{p_{j+1}}{w_{j+1}} \sum_{k=j+1}^{n} w_k \geq \sum_{k=j+1}^{n} p_k$, we get

$$\frac{\sum_{i=1}^{j} p_i}{\sum_{i=1}^{j} w_i} \geq \frac{p_{j+1}}{w_{j+1}} \geq \frac{\sum_{k=j+1}^{n} p_k}{\sum_{k=j+1}^{n} w_k}. \text{ Therefore, } \sum_{i=1}^{j} p_i \geq \left(\frac{\sum_{i=1}^{j} w_i}{\sum_{k=j+1}^{n} w_k} \right) \sum_{k=j+1}^{n} p_k .$$

□

Now, we are ready to examine the correlation between the profit-difference and the capacity-difference under the condition that the weights of all items are bounded to a constant k. If the capacity difference is Δc, Theorem 1 tells us that (a) the profit-difference between two knapsacks will never be greater than the maximum-profit of a knapsack with $\Delta c + k$ capacity; (b) the ratio of this difference to the maximum-profit of the larger knapsack is less than or equal to $(\Delta c + k) / c_1$, where c_1 is the capacity of the larger knapsack; and (c) the ratio of the maximum-profit of the smaller knapsack to that of the larger knapsack is greater than or equal to the ratio of their capacities approximately. Notice that the same set of items is to be allocated to both knapsacks.

Theorem 1:
Given two knapsack problem instances (c_1,I), (c_2,I), and a positive integer k $(w_i \leq k, \forall\ i \in I,)$

(a) $P^*_{c_2} + P(\text{Å}(\Delta c + k, I)) \geq P^*_{c_1}$,

(b) $P^*_{c_2} \geq \left(1 - \frac{\Delta c + k}{c_1} \right) \cdot P^*_{c_1}$, and

(c) $\frac{P^*_{c_2}}{P^*_{c_1}} \geq \frac{c_2}{c_1}$, if c_1, $c_2 >> k$,

where $\Delta c = c_1 - c_2$.

(b.2)

$$\frac{P^*_{c_2}}{P^*_{c_2} + P(\bar{I}_j)} \geq \frac{zP(\bar{I}_j)}{zP(\bar{I}_j) + P(\bar{I}_j)},$$

since for any positive numbers x, y, and c, $\dfrac{y}{y+c} \geq \dfrac{x}{x+c}$, if $y \geq x$.

(b.3)

$$\frac{P^*_{c_2}}{P^*_{c_1}} = \frac{P^*_{c_2}}{P(I_j) + P(\bar{I}_j)} \geq \frac{P^*_{c_2}}{P^*_{c_2} + P(\bar{I}_j)} \geq \frac{zP(\bar{I}_j)}{zP(\bar{I}_j) + P(\bar{I}_j)}$$

$$= \frac{z}{(z+1)} = \frac{c_1 - \Delta c - k}{c_1} = 1 - \frac{\Delta c + k}{c_1}$$

(c) When $c_1, c_2 >> k$, $c_1 \simeq c_2 + \Delta c + k$. After some simple transformation, we can get $\dfrac{P^*_{c_2}}{P^*_{c_1}} \geq \dfrac{c_2}{c_1}$.

□

The upper bound found in part (a) is smaller than that of (b), but it depends on another 0-1 knapsack allocation. It may not be convenient in some applications. In many cases, the information in the form of normalized deviation as follows is more useful:

$$P^*_{c_2} \geq (1-\epsilon)P^*_{c_1}, \qquad (0 \leq \epsilon \leq 1), \quad \text{where } \epsilon \text{ is a constant.}$$

In this case, part (b) and (c) are more useful since ϵ only depends on k and the knapsack capacities. By simple calculation, we can derive that the deviation will never be greater than 50% of the maximum-profit of the larger knapsack as long as $(c_1 - c_2 + k) \leq 2c_1$. When the capacities of the knapsacks are much larger than k, the ratio of their maximum-profits $(P^*_{c_2} / P^*_{c_1})$ is approximately greater than the ratio of their capacities (c_2 / c_1).

To estimate the performance of the heuristic algorithm of the multiple knapsack problem mentioned in Section 1, it is important to know how much the maximum-profit may be reduced when some items are preallocated in the knapsack. If we follow the logic in Theorem 1, we can easily derive the similar results as shown in Theorem 2. Theorem 2 should find an upper bound for this

(a) $P(\mathring{A}(c-W(Q), I-Q)) + P(Q) + P(\mathring{A}(W(Q)+k, I)) \geq P(\mathring{A}(c,I)) = P^*$,

(b) $P(\mathring{A}(c-W(Q), I-Q)) + P(Q) \geq \left(1 - \dfrac{W(Q)+k}{c}\right)P^*$, and

(c) $\dfrac{P(\mathring{A}(c-W(Q), I-Q)) + P(Q)}{P^*} \geq \dfrac{c-W(Q)}{c}$, if $c, c-W(Q) >> k$,

where $(w_i \leq k, \forall\ i \in I)$.

Proof:

Denote $c_2 = c - W(Q)$, and $P^*_{c_2} = P(\mathring{A}(c-W(Q), I-Q)) + P(Q)$.

(a)

(a.1) Similar to Theorem 1, we assume $\mathring{A}(c,I) = \{1,2,..,l\}$ to be an optimal allocation, then we can find j such that

$(c - W(Q) - k) \leq \sum_{i=1}^{j} w_i \leq (c - W(Q))$. We denote I_j as the set of items $\{1,2, \ldots, j\}$ and $\bar{I_j}$ as $(\mathring{A}(c,I) - I_j) = \{j+1, \ldots, l\}$ as usual.

(a.2) Since $W(I_j) \leq c - W(Q)$ and $c_2 = c - W(Q)$, $c_2 \geq W(I_j)$. It can be proved that $P^*_{c_2} = P(\mathring{A}(c_2, I-Q)) + P(Q) \geq P(I_j)$ as follows:

case 1: $Q \cap I_j = \varnothing$.

We know $P(\mathring{A}(c_2, I_j - Q)) = P(I_j - Q) = P(I_j)$ since
$c_2 \geq W(I_j - Q)$ and
$P(\mathring{A}(c_2, I-Q)) \geq P(\mathring{A}(c_2, I_j - Q)) = P(\mathring{A}(c_2, I_j))$ from Lemma 1.2. Therefore,
$P^*_{c_2} = P(\mathring{A}(c_2, I-Q)) + P(Q) \geq P(\mathring{A}(c_2, I_j)) = P(I_j)$.

case 2: $Q \cap I_j \neq \varnothing$.

(a.2.1) Since $I_j \subset I$, so $(I_j - Q) \subset (I - Q)$.

(a.2.2) Since $c_2 = c - W(Q) \geq W(I_j) \geq W(I_j - Q)$ and (a.2.1), we get $P(\mathring{A}(c_2, I-Q)) \geq P(\mathring{A}(c_2, I_j - Q)) = P(I_j - Q)$.

(a.2.3) $P(Q) \geq P(Q \cap I_j)$.

Further notations are introduced as follows.

I_t the t-th set of items. $I_t = \{i_t | i = 1,2, \ldots, n\}$

w_{i_t} the weight of item i_t ;

p_{i_t} the profit of item i_t;

$f_{ts}(A)$ the mapping from the set of items $A \subseteq I_t$ to the corresponding items in set I_s, i.e., $f_{ts}(A) = \{i_s \mid i_s \in I_s, \forall\ i_t \in A,\ A \subseteq I_t\}$.

$P^*_{I_t}$ the maximum-profit of (c, I_t), i.e. $P^*_{I_t} = P(\text{Å}(c, I_t))$.

Lemma 3.1:

Given two knapsack problem instances (c, I_1) and (c, I_2),

$P^*_{I_1} \geq P^*_{I_2}$, if $|I_1| = |I_2| = n$ and any of the following conditions holds,

(a) $w_{i_1} = w_{i_2}$ and $p_{i_1} \geq p_{i_2}$, $i = 1,2, \ldots, n$;

(b) $w_{i_1} \leq w_{i_2}$ and $p_{i_1} = p_{i_2}$, $i = 1,2, \ldots, n$;

(c) $w_{i_1} \leq w_{i_2}$ and $p_{i_1} \geq p_{i_2}$, $i = 1,2, \ldots, n$.

Proof: Proved by contradiction for all cases.
For (a), if there exists such a pair of instances that $P^*_{I_1} < P^*_{I_2}$, we can allocate $f_{21}(\text{Å}(c, I_2))$ to the knapsack such that $P(f_{21}(\text{Å}(c, I_2))) \geq P^*_{I_2} > P^*_{I_1}$. This is a contradiction.

Similarly, it is easy to prove for condition (b) and (c).

□

Lemma 3.2:

Given two knapsack problem instances (c, I_1) and (c, I_2),

$$P^*_{I_2} + \sum_{i=1}^{n} \mathcal{D}\,(p_{i_1} - p_{i_2}) \geq P^*_{I_1},$$

if $|I_1| = |I_2| = n$ and
$w_{i_1} \geq w_{i_2}$, $i = 1,2, \ldots, n$,

where

$$\mathcal{D}\,(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0. \end{cases}$$

In the multiple-choice-knapsack problem, the allocation may be done by arbitrarily fixing one choice for each item first, then allocating them to the knapsack as a standard 0-1 knapsack problem. Such a heuristic is also useful in a standard 0-1 knapsack problem when the actual profits of items to be allocated are nondeterministic before the allocation. Lemma 3.2 and Theorem 3 are useful to evaluate such a heuristic algorithm. In Corollary 3.1, the result is extended to the case that an arbitrary number of sets of items are to be allocated.

Corollary 3.1:
Given a set of knapsack problem instances (c,I_1), (c,I_2), $\ldots$, (c,I_m), then

$$P(f_{\alpha\beta}(\mathring{A}(c,I_\alpha))) + \sum_{i=1}^{n}(\max_j \{p_{i_j}\} - \min_j \{p_{i_j}\}) \geq P^*_{I_\beta},\ \alpha,\beta = 1,2,\ldots,m$$

if $|I_1| = |I_2| = ,\ldots, = |I_m| = n$,

$w_{i_0} = w_{i_1} = w_{i_2} = ,\ldots, = w_{i_m}$, $i = 1,2,\ldots,n$.

Proof: It is easy to prove from Theorem 3.

□

To use Corollary 3.1, we can imagine that $P^*_{I_\beta}$ is the optimal solution and the item set I_α is the choices chosen before the allocation.

4. A PERFORMANCE GUARANTEED HEURISTIC SOLUTION FOR 0-1 MULTIPLE-KNAPSACK PROBLEMS

A special case of *0-1 multiple-knapsack problem* is as follows: Given a set of positive integers $C=\{c_\ell \mid \ell=1,2,\ldots,m\}$, and n sets of positive integers, $(p_{1i},\ldots,p_{\ell i},\ldots,p_{mi},w_i)$, for $i=1,2,\ldots,n$, find $x_{11},x_{12},\ldots,x_{mn}$ so as to pseudo polynomial optimal algorithms, optimal solutions may not be difficult to obtain in practical cases. Here, we assume an optimal algorithm is used.

Once the profit of each item is nondeterministic among several choices before the allocation, the algorithm MK-HEUR can be modified to algorithm MCMK-HEUR as follows.

Heuristic Algorithm MCMK-HEUR

1. Select one choice for each item either randomly or according to any estimation method.
2. Use algorithm MK-HEUR to solve it.

□

In the rest of this section, we try to estimate the largest possible errors that the algorithm MK-HEUR and MCMK-HEUR may produce. Further notations are defined as follows. For clarity, the notations for MCMK-HEUR are not explicitly shown here. Subscripts are added to the item indices in MCMK-HEUR to indicate various choices.

I — the set of categories, $I=\{1,2,\ldots,n\}$;

C — the set of knapsacks, $C=\{c_1,c_2,\ldots,c_m\}$;

(C,I) — the multiple-knapsack problem instance that I is to be allocated to C;

ℓi — the item of category i that is to be allocated to knapsack ℓ, (called item ℓi);

I_ℓ — the set of items to be allocated to knapsack ℓ, $I_\ell=\{\ell 1,\ell 2,\ldots,\ell n\}$;

c_ℓ — the capacity of knapsack ℓ;

w_i — the weight of an item i;

$p_{\ell i}$ — the profit of an item of category i if it is allocated to knapsack ℓ, (the profit of item ℓi);

$x_{\ell i}$ — the allocation of an item of category i in knapsack ℓ, (the allocation of item ℓi),

$$x_{\ell i}=\begin{cases}1 & \text{if an item of category } i \text{ is allocated to knapsack } \ell,\\ 0 & \text{if an item of category } i \text{ is not allocated to knapsack } \ell;\end{cases}$$

(c_ℓ,I_ℓ) — a 0-1 knapsack problem instance that the set of items I_ℓ is to be allocated to knapsack ℓ with a capacity c_ℓ. It may also be written as (c_ℓ,I);

Proof: Since a solution of a 0-1 multiple-knapsack problem instance is also a feasible solution of a many-knapsack allocation, the maximum-profit of a many-knapsack allocation will never be smaller than that of the 0-1 multiple-knapsack problem.

□

Similar result for MCMK-HEUR is shown in Corollary 4.1.

Corollary 4.1:

$$P^* = P(MCM\mathring{A}(C,I)) \le \sum_{\ell=1}^{m} P(\mathring{A}(c_\ell, I_\ell^0)).$$

Proof: It can be proven in a way similar to Lemma 4.1 based on the result in Corollary 3.1.

□

Theorem 4:
The largest possible error that algorithm MK-HEUR can produce is no greater than

(a) $\sum_{\ell=1}^{m} P(\mathring{A}((1+\lceil \frac{n}{m} \rceil)k \,,\, I_\ell)),$ if $(w_i \le k, \forall\ i \in I).$

(b) $P^* \left(\sum_{\ell=1}^{m} \left(\epsilon_\ell \frac{P_\ell^*}{P^*} \right) \right),$ if $(w_i \le k, \forall\ i \in I),$

where $\epsilon_\ell = \dfrac{(q+1)k}{c_\ell}$, $q = \lceil \frac{n}{m} \rceil$, and $P_\ell^* = P(\mathring{A}(c_\ell, I_\ell)).$

Proof: Denote Q_ℓ as the set of items being allocated to knapsack ℓ in step 1 of algorithm MK-HEUR and H is the set of items being allocated by algorithm MK-HEUR .

(a)

(a.1) $$P(H) = \sum_{\ell=1}^{m} (P(\mathring{A}(c_\ell - W(Q_\ell) \,,\, I_\ell - Q_\ell)) + P(Q_\ell))$$

Corollary 4.2:
The largest possible error that algorithm MCMK-HEUR can produce is no greater than

(a) $\sum_{\varrho=1}^{m}\left(P(\text{Å}((1+\lceil\frac{n}{m}\rceil)k\ ,\ I_\varrho^0))\ +\ \sum_{i=1}^{n}\overline{p_{\varrho i}}\right)$, if $(w_i \le k, \forall\ k\in I)$;

(b) $P^*\left(\sum_{\varrho=1}^{m}\left(\epsilon_\varrho \frac{P_{\varrho_0}^*}{P^*}\right)\right)$, if $(w_i \le k, \forall\ i\in I)$ and $(\sum_{i=1}^{n}\overline{p_{\varrho i}} << P(\text{Å}((1+\lceil\frac{n}{m}\rceil)k\ ,\ I_\varrho^0)))$, where $\epsilon_\varrho = \dfrac{(q+1)k}{c_\varrho}$ $q = \lceil\frac{n}{m}\rceil$, and $P_{\varrho_0}^* = P(\text{Å}(c_\varrho, I_\varrho^0))$.

Proof: Similar to Theorem 4, it can be easily proven.

□

Although the heuristic algorithms MK-HEUR and MCMK-HEUR are not very elegant, it is not difficult to obtain better solutions by modifying them. In this way, the performance of these improved solutions may be predictable.

5. CONCLUDING REMARKS

In this article, the properties of some 0-1 knapsack problems are explored. In Section 2, the same set of items are to be allocated to two knapsacks with different capacities. When the weights of all items are no greater than a constant k, the smallest profit-difference we found in this paper is the maximum-profit of a knapsack with $(c_1 - c_2 + k)$ capacity or $(c_1 - c_2 + k) / c_1$ of the maximum-profit of the larger knapsack, where $c_1(c_2)$ is the capacity of the larger (smaller) knapsack. When the capacities of both knapsacks are much larger than k, the ratio of their maximum-profit $(P_{c_2}^* / P_{c_1}^*)$ is greater than the ratio of their capacities (c_2 / c_1) approximately. The result is extended to the case when a subset of the items (Q) is already allocated into the knapsack before the allocation; the maximum-profit reduction will never be greater than the maximum-profit of a knapsack with $W(Q)+k$ capacity, where $W(Q)$ is the total weight of Q. The deviations depending on the capacity, Q, and k are also obtained. The difference of maximum-profits of allocating two sets of items into a knapsack is also investigated in Section

8. Sinha, P. and A. A. Zoltners, "The Multiple Choice Knapsack Problem," *Operatings Research*, vol. 27, no. 3, pp. 503-515, May-June 1979.

9. Wah, B. W. and Y. N. Lien, "Design of Distributed Databases On Local Computer Systems With A Multiaccess Network," *IEEE Trans. On Software Eng.*, vol. SE-11, no. 7, pp. 606-619, July 1985.