Mathematical Programming 15 (1978) 155–161.
North-Holland Publishing Company

THE COLLAPSING 0–1 KNAPSACK PROBLEM

Marc E. POSNER*

RCA/David Sarnoff Research Center, Princeton, NJ, U.S.A.

Monique GUIGNARD

University of Pennsylvania, Philadelphia, PA, U.S.A.

Received 6 December 1976
Revised manuscript received 19 April 1978

The "Collapsing 0–1 Knapsack Problem" is a type of non-linear knapsack problem in which the knapsack size is a non-increasing function of the number of items included.

An algorithm is developed and computational results included.

Key words: Nonlinear Knapsack, 0–1 Programming.

1. Introduction

Recently, several approaches to the non-linear 0–1 knapsack problem have been introduced in the OR literature. See, for example, Lawler and Bell [14], Hammer and Rudeanu [8], Hansen [9], and Huard [10].

One important class of non-linear knapsack problems may be referred to as partitioning processes. Given a collection of items (usually sequentially ordered) a partition must be created between items, separating each from every other. That is, one item requires no partition, two items one partition, three items two partitions, etc. This problem can be formally stated as:

$$\text{(P1)}\quad \begin{aligned} &\text{maximize} && z_1(x) = \sum_{i=1}^{n} f_i x_i, \\ &\text{subject to} && \sum_{i=1}^{n} k_i x_i \leqq h\Big(\sum_{i=1}^{n} x_i\Big), \\ &&& x_i = 0 \text{ or } 1, \quad i = 1, \dots, n, \\ &&& f_i, k_i \text{ known integers} > 0, \quad i = 1, \dots, n, \\ &&& h : R \to R \text{ a monotone non-increasing functional.} \end{aligned}$$

The non-linearity of the problem lies in the form of the right-hand side of the constraint. The physical interpretation of this is that the partition itself uses some of the available resource.

* Currently affiliated with the University of Wisconsin-Milwaukee, Milwaukee, WI, 53201.

This research was motivated by an application in satellite communications. When a particular communications band has only one user (sent Frequency Division Multiple Access) the entire band can be used. However, multiple transmissions on the band require gaps—partitions—between the portions of the band assigned to each user. These partitions prevent cross talk problems, but also reduce the effective communications capacity of the band. Therefore, given a capacity constrained environment the problem of which customers to allow on your communications band can be handled by a formulation such as the form (P1). f would take the value of each customer (generally the revenue generated), k the size of the band width each customer requires and h the capacity of the band as a function of the number of users on the system.

Although the problem was motivated by a specific application it can be generalized. Other applications in this class are e.g., design of a shopping center where both the type and size of shop must be selected, and the loading of fragile items on a truck. A second but similar class of problems which can be modeled by (P1) are "multiplicity" problems. A trivial example will serve to illustrate this idea. It is relatively easy to carry one basketball. However, although much smaller in total volume, it would require great dexterity to carry fifty marbles. Thus capacity is affected not only by size but also by number of items chosen. A more relevant example of such a problem is computer operations in a time sharing environment. With only one user on the system, the system can be used continuously. However, suppose two or more users are on simultaneously. Then time must be spent not only to throw one user off and bring another on but also to keep track of everyone that is on the system. As the number of users increase the effective processing capabilities of the computer decreases.

Which customers should be allowed on the computer to optimize profit? The solution to (P1) with the proper assignment of values to its parameters will tell us.

This paper will propose an algorithm to provide an efficient solution to this problem.

2. Notation

Several notational conventions and abbreviations will be used for the sake of simplicity:

$\#T \equiv$ cardinality of set T,

$x_T \equiv \{x_i \mid i \in T\}$,

$(0, \dots, 0, (b_r), 0, \dots, 0) \equiv$ vector that has a value of b in the r^{th} component,

FS(P1) $\equiv$ Feasible set of problem (P1),

OS(P1) $\equiv$ Optimal set of problem (P1).

3. Development of the method

The algorithm used to solve P1 relies heavily upon implicit enumeration, following E. Balas [1]. However, the peculiarities of the problem have made many improvements possible. In this section we will detail some of these improvements.

Lemma 1 will be familiar to those interested in the Classical 0–1 Knapsack problem. It states that if the knapsack is full with only the most valuable per unit items included, then the solution is optimal.

Lemma 1. *Let $h(a) = c$ be constant for integer $a \geqq 0$. Assume that the variables are reordered such that if $r \leqq n$, then for each $i \in \{1, \dots, r\}$ and $j \in \{r+1, \dots, n\}$, $f_i/k_i \geqq f_j/k_j$. If $\sum_{j=1}^{r} k_i = c$, then $x^* = (1, \dots, (1)_r, 0, \dots, 0) \in \mathrm{OS(P1)}$.*

This lemma will be needed in Theorem 1, a theorem central to the approach. Theorem 1 provides a method for bypassing many feasible solutions. Under certain conditions, it also allows us to place an efficient upper bound on the number of objects in the knapsack, which in turn allows the skipping of additional feasible solutions.

Theorem 1. *Assume $f_1/k_1 \geqq, \dots, \geqq f_n/k_n$. Let $r \in \{1, 2, \dots, n\}$ such that $\sum_{i=1}^{r-1} k_i + k_r\alpha = h(r)$ for $1 \geqq \alpha > 0$. Then $\sum_{i=1}^{r-1} f_i + f_r\alpha \geqq z_1(x)$ for all $x \in \mathrm{FS(P1)}$ such that $\sum_{i=1}^{n} x_i \geqq r$.*

Proof. Consider the problem

$$\text{(P2)}\qquad \begin{aligned} &\text{maximize} && z_2(x_1, \dots, x_{n+1}) = \sum_{i=1}^{n} f_i x_i + \alpha f_r x_{n+1}, \\ &\text{subject to} && \sum_{i=1}^{n} k_i x_i + \alpha k_r x_{n+1} \leqq h(r), \\ &&& x_i = 0 \text{ or } 1, \quad i = 1, \dots, n+1. \end{aligned}$$

By Lemma 1, $(1, \dots, (1)_{r-1}, 0, \dots, 0, 1) \in \mathrm{OS(P2)}$. Since h is monotonically decreasing, if $\bar{x} \in \mathrm{FS(P1)}$ such that $\sum_{i=1}^{n} \bar{x}_i \geqq r$, then $(\bar{x}, 0) \in \mathrm{FS(P2)}$. Therefore, $z_1(\bar{x}) \leqq$ the optimal value of (P2).

When the results of Theorem 1 are inapplicable, we can still establish an upper bound by making use of Theorem 2.

Theorem 2. *Assume the indices are reordered such that $k_1 \leqq k_2 \leqq, \dots, \leqq k_n$. Let*

$$N = \max\left\{a \mid \sum_{i=1}^{a} k_i \leqq h(a)\right\}.$$

If $x \in \mathrm{FS(P1)}$, then $\sum_{i=1}^{n} x_i \leqq N$.

Proof. Suppose there exists $x \in \mathrm{FS(P1)}$ such that $\sum_{i=1}^{n} x_i = m > N$. Then $\sum_{i=1}^{n} k_i x_i \geqq \sum_{i=1}^{m} k_i > h(N+1) \geqq h(m)$. This contradicts the fact that $x \in \mathrm{FS(P1)}$.

A partial solution, using implicit enumeration, is an assignment of binary values to a subset of the x_i's. We define F to be the set of indices of x in a partial solution; $\bar{F}$, then, is the set of indices which have yet to be assigned a value. x_j is "fixed" if $j \in F$.

According to the following theorem, if the best possible completion (which may not be feasible) of the partial solution x_F has less value than the current best solution x', we may skip to the next partial solution.

Theorem 3. *Let $T \subseteq \bar{F}$ such that $f_j \leqq f_i$ for every $i \in T$ and $j \in (\bar{F} - T)$ and such that $\#T = N - \sum_{i \in F} x_i$. If y is a feasible completion of x_F and $z_1(x') \geqq \sum_{i \in F} f_i x_i + \sum_{i \in T} f_i$, then $z_1(x') \geqq z_1(y)$.*

Proof. Let $L = \{i \mid y_i = 1\}$. By Theorem 2, $\sum_{i=1}^{n} y_i \leqq N$. Therefore $\#T = N - \sum_{i \in F} y_i \geqq \#(L \cap \bar{F})$. By the definition of T, if $i \in (L \cap \bar{F})$, then either $i \in T$ or for each $j \in T$, $f_i \leqq f_j$. Hence $\sum_{i \in L \cap \bar{F}} f_i \leqq \sum_{i \in T} f_i$ and

$$z_1(x') \geqq \sum_{i \in F} f_i y_i + \sum_{i \in T} f_i \geqq \sum_{i \in F \cap L} f_i x_i + \sum_{i \in \bar{F} \cap L} f_i x_i = \sum_{i \in L} f_i x_i = z_1(y).$$

Finally, the last two theorems allow us to fix the values of x_i for certain $i \in \bar{F}$. Theorem 4 delineates conditions under which x_i may be set at 1, while Theorem 5 is concerned with setting x_i at 0.

By finding the values of x_i when $i \in \bar{F}$, the numger of completions to be considered is reduced. Using these two theorems in a repetitive fashion, we may eliminate the partial solution entirely.

Theorem 4. *Let $T \subseteq \bar{F}$ such that $f_j \leqq f_i$ for each $i \in T$, and each $j \in (\bar{F} - T)$ and such that $\#T = N - \sum_{i \in F} x_i$. Let $f_r = \max\{f_j \mid j \in (\bar{F} - T)\}$. If $\sum_{i \in F} f_i x_i + \sum_{i \in T} f_i + f_r - f_j \leqq z_1(x')$ for some $j \in T$, then $z_1(x') \geqq z_1(y)$ for all $y \in \mathrm{FS(P1)}$ such that $y_F = x_F$ and $y_j = 0$.*

The proof is similar to the proof of Theorem 3.

Theorem 5. *If there exists $j \in \bar{F}$ such that $\sum_{i \in F} k_i x_i + k_j \geqq h(\sum_{i \in F} x_i + 1)$, then every feasible completion y has $y_j = 0$.*

Proof. $\sum_{i \in F} k_i x_i + \sum_{i \in \bar{F}} k_i x_i \geqq \sum_{i \in F} k_i x_i + k_j \geqq h(\sum_{i \in F} x_i + 1)$.

4. The algorithm

(1) $I = \{1, 2, \dots, n\}$, $z^0 = 0$. For all indices $i \in I$:
 (a) if $k_i > h(1)$, then $I = I - \{i\}$,

(b) if $h(1) \geqq k_i \geqq h(2)$, then $I = I - \{i\}$ and if $f_i > z^0$, then $z^0 = f_i$ and

$$x_j^0 = \begin{Bmatrix} 1 & j = 1 \\ 0 & j \neq i \end{Bmatrix},$$

(c) if $h(2) > k_i \geqq h(3) - 1$, then $I = I - \{i\}$ and let

$$f_r = \max\{f_j \mid j \neq i,\ j \in I,\ k_i + k_j \leq h(2)\}.$$

if $f_i + f_r > z^0$, then $z^0 = f_i + f_r$ and

$$x_j^0 = \begin{cases} 1 & \text{if } j = i \text{ or } j = r, \\ 0 & \text{otherwise.} \end{cases}$$

(2) Reorder indices such that $f_1/k_1 \geqq f_2/k_2 \geqq, \dots, \geqq f_{\#I}/k_{\#I}$. Let

$$r = \max\left\{a \mid \sum_{i=1}^{a} k_i < h(a)\right\} + 1,\ F = S = \{1, 2, \dots, r-1\}$$

If $h(r) > \sum_{i=1}^{r-1} k_i$ and

$$\sum_{i=1}^{r-1} f_i + f_r \left(\frac{h(r) - \sum_{i=1}^{r-1} k_i}{k_r} \right) \leqq z^0,$$

then $N = r - 1$ (Theorem 1).

Otherwise let $k_{\sigma_1} \leqq, \dots, \leqq k_{\sigma_{\#I}}$, $N = \max\{a \mid \sum_{i=1}^{a} k_{\sigma_i} \leqq h(a)\}$ (Theorem 2).

Go to (3).

(3) If $\sum_{i\in F} k_i x_i + \sum_{i\in \bar{F}} k_i < h(\sum_{i\in F} x_i + \sum_{i\in \bar{F}} 1)$, then set $x_{\bar{F}} = 1$, $F = F \cup \bar{F}$, go to (9). Otherwise go to (4).

(4) Let $T \subseteq \bar{F}$ such that $\#T = \min\{N - \sum_{i\in F} x_i, \#\bar{F}\}$ and $f_j \leqq f_i$ for each $i \in T$ and $j \in (\bar{F} - T)$.

If $\sum_{i\in F} f_i x_i + \sum_{i\in T} f_i \leq z^0$, then go to (10). Otherwise go to (5).

(5) Let $f_r = \max\{f_i \mid i \in (\bar{F} - T)\}$ and let

$$V = \left\{ j \in T \mid \sum_{i\in F} f_i x_i + \sum_{i\in T} f_i + f_r - f_j \leqq z^0 \right\}.$$

Set $S_V = 1$, $F = F \cup V$, (Theorem 4).

If $\sum_{i\in F} k_i x_i > h(\sum_{i\in F} x_i)$, then go to (10). Otherwise go to (6).

(6) Let $W = \{j \in \bar{F} \mid \sum_{i\in F} k_i x_i + k_j > h(\sum_{i\in F} x_i + 1)\}$.

If $W \neq \emptyset$, then set $x_w = 0$, $F = F \cup W$ and go to (3) (Theorem 5). Otherwise go to (7).

(7) Let $R \subseteq \bar{F}$ and let $r = \min a \in (\bar{F} - R)$ such that $i \leqq j$ for every $i \in R$ and $j \in (\bar{F} - R)$, and such that $\sum_{i\in F} k_i x_i + k_R \leqq h(\sum_{i\in F} x_i + \#R)$, and $\sum_{i\in F} k_i x_i + k_R + k_r > h(\sum_{i\in F} x_i + \#R + 1)$.

Set $x_r = 1$, $F = F \cup R$, $S = S \cup R$.

If $\sum_{i\in F} k_i x_i < h(\sum_{i\in F} x_i + 1)$, then go to (8). Otherwise go to (9).

(8) If

$$\sum_{i\in F} f_i x_i + f_r \frac{h\left(\sum_{j\in F} x_i + 1\right) - \sum_{i\in F} k_i x_i}{k_r} \leqq z^0,$$

then go to (10) (Theorem 1). Otherwise set $x_r = 0$, $F = F \cup \{r\}$ and go to (3).

(9) If $\sum_{i\in F} f_i x_i > z^0$, then $z^0 = \sum_{i\in F} f_i x_i$, $x_F^0 = 0$. Go to (10).

(10) If $S = \emptyset$, then $z^* = z^0$, $x^* = x^0$ and stop. Otherwise

$$j = \max\{i \in S \mid x_i = 1\}.$$

$$x_j = 0, \quad S = S - \{i \in S \mid i > j\}, \quad F = S.$$

Go to (3).

5. Computational results

This algorithm was programmed in APL and run on an IBM 370-158.

The values of f, k and h were generated using the APL random number generator. In the case of h the values were then sorted in a non-increasing sequence. Example 1 (see Table 1) will be used to illustrate this procedure. 30 random numbers between 1 and 100 were generated and assigned to f_1 thru f_{30}. Another 30 numbers were generated and assigned to k_1 thru k_{30}. Finally 10 random numbers between 1 and 300 were generated, sorted in non-increasing order and assigned to $h(1)$ thru $h(10)$. $h(11)$ thru $h(30)$ were given the value of 0. The only change to this procedure is in example 3 where $h(1) = \cdots = h(30) = 300$.

Table 1

Example	Number of variables	Range of f	Range of k	Number of i where $h(i) > 0$	Range of h	CPU (sec)
1	30	1–100	1–100	10	1–300	33.68
2	30	1–100	1–100	30	1–300	117.17
3	30	1–100	1–100	30	300–300	143.32
4	50	1–300	50–1000	10	1–1000	21.75
5	100	1–300	100–1000	10	1–1000	68.60

Although no set of examples can give conclusive evidence, the examples serve to illustrate some of the relative efficiencies involved in using our algorithm.

The algorithm makes efficient use of the knowledge that the values of h are shrinking. The faster h shrinks, the greater the number of possible solutions we can eliminate by the various tests and checks; this, in turn, speeds up solution time. Examples 1 thru 3 illustrate this idea. In Example 1, h decreases very rapidly to 0, in 2 less so and in 3, h does not decrease at all. The run times correspondingly increase.

Examples 4 and 5 illustrate another factor that contributes to reduced run times. When the size of the objects are large relative to the knapsack size, only a few items can be placed in the knapsack. This drastically reduces the number of nodes that must be checked.

References

[1] E. Balas, "An additive algorithm for solving linear programs with zero–one variables", *Operations Research* 13 (1965) 517–546.

[2] G.B. Dantzig, "Discrete variable extremum problems", *Operations Research* 5 (1957) 266–277.

[3] R. Garfinkel and G. Nemhauser, *Integer programming* (J. Wiley, New York, 1972).

[4] A. Geoffrion, "An improved implicit enumeration approach for integer programming", *Operations Research* 17 (1969) 437–454.

[5] L. Gilman and A. Rose, *APL/360 an interactive approach* (Wiley, New York, 1970).

[6] H. Greenberg and R.L. Hegerish, "A branch search algorithm for the knapsack problem", *Management Science* 16 (1970) 327–332.

[7] M. Guignard and K. Spielberg, "Mixed-integer algorithms for the (0, 1) knapsack problem", *IBM Journal of Research & Development* 16 (1972) 424–430.

[8] P. Hammer and S. Rudeanu, "Pseudo-Boolean programming", *Operations Research* 17 (1969) 233–261.

[9] P. Hansen, "Algorithme pour les programmes non lineaires en variables zero–un", Comptes Rendus 270 (1970) 1700–1702.

[10] T.C. Hu, *Integer programming and network flows* (Addison Wesley, Reading, MA, 1969).

[11] P. Huard, "Programmes mathematiques non lineaires a variables bivalentes", in: H.W. Kuhn, ed., *Proceedings of the Princeton symposium on mathematical programming* (Princeton University Press, 1970) 313–322.

[12] G.P. Ingargiola and J.F. Korsh, "Reduction algorithm for zero–one single knapsack problems", *Management Science* 20 (1973) 460–463.

[13] P.J. Kolesar, "A branch and bound algorithm for the knapsack problem", *Management Science* 13 (1967) 723–735.

[14] E. Lawler and M. Bell, "A method for solving discrete optimization problems", *Operations Research* 14 (1966) 1098–1112.

[15] T.L. Morin and R.E. Marsten, "An algorithm for nonlinear knapsack problem", *Management Science* 22 (1976) 1147–1158.

[16] R.M. Nauss, "An efficient algorithm for the 0–1 knapsack problem", *Management Science* 23 (1976) 27–31.

[17] G. Plateau and D. Fayard, "Contribution a la resolution de probleme du knapsack: methodes d'exploration", these presentee a l'universite des Sciences et Techniques de Lille obtenir le titre de Docteur de Specialite, 1971.

[18] M. Posner, "The generalized knapsack problem", Ph.D. Thesis, University of Pennsylvania (1974) 112–127.

[19] H.M. Weingartner and D.N. Ness, "Methods for the solution of the multi-dimensional 0/1 knapsack problem", *Operations Research* 15 (1967) 83–103.