Acta Cybernetica, Tom. 10, Fasc. 1-2, Szeged, 1991

Heuristics for the 0-1 Min-Knapsack Problem

J. Csirik * J.B.G. Frenk M. Labbé † S. Zhang

Econometrisch Instituut, Erasmus Universiteit Rotterdam, P.O. Box 1738, 3000DR Rotterdam, the Netherlands

Abstract

The 0-1 min-knapsack problem consists in finding a subset of items such that the sum of their sizes is larger than or equal to a given constant and the sum of their costs is minimized. We first study a greedy-type heuristic having a worst-case bound of 2. This heuristic is then refined to obtain a new one with a worst-case bound of 3/2.

1 Introduction

The classical 0-1 knapsack problem (*max-knapsack*) has been extensively studied in the literature. Some greedy-type heuristics have been analysed and ε-approximation schemes are known for this problem ([2,5,6,7]). On the contrary, the *min-knapsack* problem has found until now only few interest in the English literature. Most of the results and algorithms are translated from Russian ([1,3,4]), and are given without proof.

The min-knapsack problem is formulated as follows:
given n pairs of positive integers (c_j, a_j) and a positive integer M, find $x_1, x_2, \ldots, x_n$ so as to

$$\begin{array}{rl} \text{minimize} & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} a_j x_j \geq M \\ & x_j \in \{0,1\}, j = 1, \ldots, n. \end{array}$$

The problem is clearly NP-hard, and so finding a good heuristic solution is of interest. Obviously, the problem is feasible if and only if $\sum_{j=1}^{n} a_j \geq M$. Next we assume that this condition is satisfied for the considered problems.

In this paper, we analyse a greedy heuristic proposed by Gens and Levner [4]. A similar heuristic also exists for the max-knapsack problem. However, for the min-knapsack, we need a slight modification of the main idea. Then, the different behaviour of this heuristic for the max and min problems is shown, when the item sizes a_j are bounded by $M/k, k \geq 2$. The heuristic we consider has a worst-case bound of 2. We then provide a refinement with worst-case bound of 3/2, with a possible ε-approximation scheme extension. We finally propose some practical improvement.

*On leave from Department of Computer Science, University of Szeged, Árpád tér 2, 6720 Szeged, Hungary

†Fellow of the European Institute for Advanced Studies in Management, Brussels

2 The Heuristic

We will use throughout a_j to denote an *item* as well as its *size*, while c_j represents its *cost*. Furthermore, c_j/a_j is defined as the relative cost of item a_j.

Algorithm *GR*

Step 1. Sort the items in nondecreasing order of their relative costs. From now on, we assume that

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

Step 2. (a) Let k_1 be the index for which

$$\sum_{i=1}^{k_1} a_i < M \leq \sum_{i=1}^{k_1+1} a_i.$$

Then the sublist $(a_1, a_2, \ldots, a_{k_1+1})$ is a candidate for the solution given by the heuristic GR. Let

$$S_1 = (a_1, a_2, \ldots, a_{k_1}),$$

then this candidate can be written as

$$S_1 \cup \{a_{k_1+1}\}.$$

(b) Let $k_1+2, k_1+3, \ldots, k_2-1$ be a (possibly empty) series of indices so that for all of the corresponding items (i.e. $j \in \{k_1+2, \ldots, k_2-1\}$) the following holds:

$$\sum_{i=1}^{k_1} a_i + a_j \geq M.$$

Let

$$B_1 = (a_{k_1+1}, \ldots, a_{k_2-1}),$$

then all $S_1 \cup \{a_j\}, j \in \{k_1+2, \ldots, k_2-1\}$, are also candidate solutions.

(c) Now, let k_2 be the first next index for which

$$\sum_{i=1}^{k_1} a_i + a_{k_2} < M,$$

and let $k_3 \geq k_2$ be the index for which

$$\sum_{i=1}^{k_1} a_i + \sum_{i=k_2}^{k_3} a_i < M \leq \sum_{i=1}^{k_1} a_i + \sum_{i=k_2}^{k_3+1} a_i.$$

Set

$$S_2 = (a_{k_2}, a_{k_2+1}, \ldots, a_{k_3}).$$

Then, $S_1 \cup S_2 \cup \{a_{k_3+1}\}$ is also a candidate solution.

Now iterate from (b), with, in the first iteration, k_3 instead of k_1 and k_4 instead of k_2; in the i-th iteration, use k_{2i+1} and k_{2i+2}. Repeat this until the end of the list. The solution given by heuristic GR is the minimum cost candidate. It is easy to see that Steps 1 and 2 have a computational complexity of $O(n \log n)$ and $O(n)$ respectively.

3 Results

Let us denote the cost of an fixed optimal solution $\overline{X}$ for the list

$$L = (a_1, a_2, \ldots, a_n)$$

by $OPT(L)$, and the cost of the solution given by heuristic GR for the same list by $GR(L)$.

Lemma 1 *For all lists L,*

$$GR(L) \leq 2 \cdot OPT(L).$$

Proof. By applying heuristic GR to list L, we subdivise L into a sequence of alternating sublists as follows:

$$\overbrace{a_1, a_2, \ldots, a_{k_1}}^{S_1}, \overbrace{a_{k_1+1}, \ldots, a_{k_2-1}}^{B_1}, \overbrace{a_{k_2}, \ldots, a_{k_3}}^{S_2}, \overbrace{a_{k_3+1}, \ldots, a_{k_4-1}}^{B_2}, \ldots,$$

$$, \ldots, \overbrace{a_{k_{2\ell-1}+1}, \ldots, a_{k_{2\ell}-1}}^{B_\ell}, \overbrace{a_{k_{2\ell}}, \ldots, a_{k_{2\ell+1}}}^{S_{\ell+1}}, \ldots,$$

$$, \ldots, \overbrace{a_{k_{2m-1}+1}, \ldots, a_{k_{2m}-1}}^{B_m}, \overbrace{a_{k_{2m}}, \ldots, a_n}^{S_{m+1}}$$

where the last set is possibly empty, in which case $k_{2m} - 1 = n$.

Let us call the elements in S-lists *small* and in B-lists *big*. Then, clearly, the heuristic solution has exactly one big element and contains all small elements before this big element. Furthermore, it is the cheapest solution among all such candidates.

From the algorithm, we have:

$$\sum_{a_i \in \cup_{j=1}^{\ell} S_j} a_i + a_r \geq M, \text{for all } \ell = 1, \ldots, m-1 \text{ and all } a_r \in B_\ell, \tag{1}$$

and

$$\sum_{a_i \in \cup_{j=1}^{\ell} S_j} a_i < M, \text{for all } \ell = 1, \ldots, m+1. \tag{2}$$

Since inequality (2) holds in particular for $\ell = m+1$, we know that the optimal solution contains at least one big element. Let a_t be the big element with smallest index in the optimal solution and let B_q be the set containing a_t. From the algorithm, we know that

$$GR(L) \leq \sum_{a_i \in \cup_{j=1}^{q} S_j} c_i + c_t.$$

Now, let $J^* = \{i : 1 \leq i \leq n \& \overline{X}_i = 1\}$, $I = \{i : a_i \in \cup_{j=1}^q S_j\}$, $J = \{i : i \in I \& \overline{X}_i = 1\}$, $K = \{i : i \in I \& \overline{X}_i = 0\}$. Then $I = J \cup K$ and $J \cap K = \emptyset$. Since all items a_i with $i < t$ have a relative cost not larger than c_t/a_t we obtain

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in I}} c_i + c_t = \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \setminus L^{OPT} \\ i \in K}} c_i + c_t \tag{3}$$

$$\leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \frac{c_t}{a_t} \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \setminus L^{OPT} \\ i \in K}} a_i + c_t.$$

Applying (2) to the second term in the above inequality yields that the upperbound in (3) is bounded from above by

$$\sum_{\substack{a_i \in (\cup_{j=1}^q S_j \cap L^{OPT} \\ i \in J}})c_i + \frac{c_t}{a_t}(M - \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} a_i) + c_t$$

and this implies by the feasibility of $\overline{X}$ that

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in I}} c_i + c_t \leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \frac{c_t}{a_t} \sum_{\substack{a_i \in L^{OPT} \setminus (\cup_{j=1}^q S_j) \\ i \in J^* - J}} a_i + c_t. \tag{4}$$

Finally, since the first big element in the optimal solution is $a_t \in B^q$ and hence all items in $J^* - J$ have a relative cost not smaller than c_t/a_t, we obtain by (4) that

$$\sum_{\substack{a_i \in \cup_{j=1}^q S_j \\ i \in J}} c_i + c_t \leq \sum_{\substack{a_i \in (\cup_{j=1}^q S_j) \cap L^{OPT} \\ i \in J}} c_i + \sum_{\substack{a_i \in L^{OPT} \setminus (\cup_{j=1}^q S_j) \\ i \in J^* - J}} c_i + c_t \tag{5}$$

$$= OPT(L) + c_t,$$

and using $\overline{X}_t = 1$ the expression in (5) is bounded above by $2 \cdot OPT(L)$.

It is easy to show that the bound given in Lemma 1 is tight. Consider the list $L = (1, M-2, M-1)$ with relative costs of $(1, 1, 1)$. Then, $OPT(L) = M, GR(L) = 2M - 2$ and that yields that $GR(L)/OPT(L)$ can be arbitrarily close to 2.

It is interesting to note that, contrary to the max-knapsack greedy heuristic, this bound remains the same if the items are small. Let k be a positive integer such that $k \geq 2$ and $k \ll M$. Assume that $a_i \leq M/k$ for all items, and let

$$L = (\underbrace{M/k, \ldots, M/k}_{(k-1) times}, M/k - 1, M/k)$$

with costs of

$$(\underbrace{1,\ldots,1}_{(k-1)times}, M/k-1, M/k).$$

Then, $OPT(L) = M/k+k-1$ and $GR(L) = 2M/k+k-2$. Hence, $GR(L)/OPT(L)$ can be arbitrarily close to 2 if M is large enough.

Using GR, we can derive a better heuristic as follows. For all big items $a_i \in B_L = \cup_{j=1}^m B_j$, we define a new knapsack problem. Let $L_i = L \setminus \{a_i\}$ and let the capacity of the knapsack $M_i = M - a_i$. The **improved heuristic** IGR is: for all $a_i \in B_L$, apply GR to the problem defined by L_i and knapsack capacity M_i. Let

$$IGR_i = GR(L_i) + c_i.$$

Then, the cost of the solution obtained with IGR is

$$IGR(L) = \min\{\min_{a_i \in B_L} IGR_i, GR(L)\}. \tag{6}$$

Since $|B_L| = O(n)$ and since, once the items are ranked by order of nonincreasing relative costs, the application of GR for each big item can be performed in linear time, the time complexity of IGR is $O(n^2)$.

Lemma 2 *For all lists L,*

$$IGR(L) \leq 3/2 \cdot OPT(L).$$

Proof. Let a_t be the smallest-index big item in a fixed optimal solution $\overline{X}$. We distinguish two cases.

(a) $c_t < 1/2 \cdot OPT(L)$.
In this case it follows directly from the proof of Lemma 1, that

$$GR(L) \leq 3/2 \cdot OPT(L),$$

and the result follows from (6).

(b) $c_t \geq 1/2 \cdot OPT(L)$.
In this case we obtain by (6) that

$$IGR(L) \leq GR(L_t) + c_t$$

and hence by the worst-case result for $GR(L)$ mentioned in Lemma 1

$$IGR(L) \leq 2 \cdot OPT(L_t) + c_t.$$

Observing now that $\overline{X}_t = 1$ finally yields

$$IGR(L) \leq 2 \cdot OPT(L_t) + c_t \leq 2 \cdot (OPT(L) - c_t) + c_t \leq 3/2 \cdot OPT(L).$$

We could get heuristics with better and better worst-case bounds by applying successively the improved method to pairs, triplets,... of big elements. This would lead to a heuristic similar to the one given by Sahni [7] for the max- knapsack problem. The result of this series of improvements is a polynomial approximation scheme, which is not fully polynomial.

From a practical point of view, we can improve the behaviour of GR, without changing its worst-case bound. This improved heuristic will be called GR^+ and consists in the following. Let $a'_1, a'_2, \ldots, a'_k$ be a candidate solution for GR (i.e. $a'_1, a'_2, \ldots, a'_{k-1}$ are small and a'_k is a big item). We delete a'_{k-1} if

$$\sum_{i=1}^{k-2} a'_i + a'_k \geq M.$$

If we could delete a'_{k-1}, then we try to delete a'_{k-2}. This is possible if

$$\sum_{i=1}^{k-3} a'_i + a'_k \geq M.$$

Hence, we delete items until

$$\sum_{i=1}^{\ell-1} a'_i + a'_k < M.$$

and the candidate solution for GR^+ is:

$$a'_1, a'_2, \ldots, a'_\ell, a'_k.$$

The solution given by GR^+ is the minimum cost candidate.

Clearly, the candidates for GR^+ are not more expensive than the candidates for GR, so that

$$GR^+(L) \leq GR(L) \leq 2 \cdot OPT(L).$$

References

[1] Babat L.G., Linear functionals on the n-dimensional unit cube, *Soviet. Math. Dokl. 16*(1975), 398-400.

[2] Friese A.M. and M.R.B. Clarke, Approximation algorithms fot the m- dimensiomal 0-1 knapsack problem: worst-case and probabilistic analyses, *Eur. J. Oper. Res. 15*(1984), 100-109.

[3] Gens G.B. and Y.V. Levner, Approximate algorithms for certain universal problems in scheduling theory, *Engineering and Cybernetics 3*(1978),31-36.

[4] Gens G.B. and Y.V. Levner, Computational complexity of approximation algorithms for combinatorial problems,*Lecture Notes in Computer Sciences, Vol. 74*, Mathematical Foundations of Computer Science,292- 300.

[5] Ibarra O.H. and C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *JACM 22*(1975), 463-468.

[6] Lawler E.L., Fast approximation algorithms for knapsack problems, *Math. of O.R. 4*(1979), 339-356.

[7] Sahni S., Approximate algorithms for the 0-1 knapsack problem, *JACM 22*(1975), 115-124.

(Received May 20, 1991)