International Conference
24th Mini EURO Conference
"Continuous Optimization and Information-Based Technologies in the Financial Sector"
(MEC EurOPT 2010)
June 23–26, 2010, Izmir, TURKEY

ISBN 978-9955-28-598-4
R. Kasımbeyli, C. Dinçer, S. Özpeynirci and L. Sakalauskas (Eds.): MEC EurOPT 2010
Selected papers. Vilnius, 2010, pp. 183–189

ALGORITHMS WITH GUARANTEE VALUE FOR BOUNDED KNAPSACK PROBLEMS

Aslı Guler[1], Fidan Nuriyeva[2]

[1]*Department of Mathematics, Faculty of Science and Letter, Yasar University, Izmir, Turkey*
[2]*Department of Mathematics, Faculty of Science, Ege University, Izmir, Turkey*

Abstract: In this study, one-dimensional ***Knapsack Problems***, which have many applications in economic area, have been studied; then greedy algorithms have been discussed for these problems. Guarantee values of the algorithms have been calculated in order to determine how close the results, returned by the algorithms, are to optimal solutions. Furthermore, complementary problem for *Bounded Integer Maximization Knapsack Problem* has been defined; and it has been aimed to improve the guarantee value calculated earlier in terms of the complementary problem.

Keywords: integer programming; knapsack problems; greedy algorithms; guarantee value; worst-case analysis; complementary problem.

1. Introduction

It is known that most of the optimization problems are NP-complete problems; and there is little hope to find exact algorithms for these problems, which run in polynomial time, unless P = NP (Kellerer *et al*., 2004). Therefore, designing approximate algorithms with guarantee value for NP-complete problems became one of the attractive subjects recently (Gens and Levner, 1980; Hromkovic, 2003).

In this study, we will focus on approximation algorithms, which run in polynomial time and find solutions that are guaranteed to be close to optimal solution. At the same time, we will be interested in proving algorithms which find solutions that are guaranteed to be close to the optimum solution.

Another major focus of this paper is to find efficient algorithms for bounded integer maximization knapsack problem through the notion of complementary problem. We will look for a bound on the largest possible running time that the algorithm could have over all inputs of a given size.

Definition: Let P be an optimization (minimization or maximization) problem with positive integral cost function c, and let A be an algorithm which, given an instance I of P returns a feasible solution $z^A(I)$, and let $z^*(I)$ be the optimal solution. Algorithm A is an approximation algorithm with relative performance guarantee Δ if $\left(z^A(I)/z^*(I)\right) \geq \Delta$ holds for all problem instances I. Clearly, given inequality makes sense for maximization problems and $0<\Delta<1$. If it is a minimization problem, then it must be $\left(z^A(I)/z^*(I)\right) \leq \Delta$, $\Delta>1$. An algorithm with relative performance guarantee Δ will be called a Δ***-approximation algorithm*** and Δ is generally called the approximation ratio (Kellerer *et al.*, 2004). In this study, we name Δ as the ***guarantee value*** of the algorithm.

In this study, we will consider greedy algorithms that will be simple and fast with the challenge being to find a greedy rule that leads to solutions provably close to optimal. The greedy method is perhaps the most straightforward design technique (Vazirani, 2001).

The notions which are used in the study, are given below:

$$B = \{0,1\};\ N = \{1,2,...,n\},\ \text{set of items};\ N_0 = \{0,1,2,...\},\ Z^+ = \{1,2,...\}$$

$$BI = \{x_j \in N_0 \mid x_j \leq b_j,\ j \in N,\ b_j \in N\},\ \text{set of variables with bounded integer,}$$

For the notions that are given below, it is chosen one of the signs $i \in \{+,-\}$ as superscript; and it shows that the problem is a maximization/minimization problem. For subscript, it is chosen one of *B, BI* and ($ii \in \{B, BI\}$); it shows respectively the problem with 0-1 variable, bounded integer or integer variable.

K_{ii}^{i} = Optimal solution value of KP,

A_{ii}^{i} = Solution value of KP found by greedy algorithm,

CK_{ii}^{i} = Optimal solution value of complementary problem of given problem,

CA_{ii}^{i} = Solution value of complementary problem of given problem found by greedy algorithm,

Greedy algorithms for the problem types given above are called *A* (**problem notation**).

f_{ii}^{*} = Improved solution value for maximization KP that is implied by subscript.

2. Knapsack problems

KP is one of the easiest NP-hard optimization problems according to the polynomial-time approximability (Martello and Toth, 1990; Nikitin and Nuriyev, 1983)

The knapsack problem can be formally defined as follows (Kellerer *et al.*, 2004):

We are given an instance of the knapsack problem with item set *N*, consisting of n items and let

c : capacity of knapsack, p_j : profit of item j, w_j : weight of item j, $j \in N$;

$$x_j = \begin{cases} 1, & \text{if item j is put into knapsack;} \\ 0, & \text{otherwise;} \end{cases}$$

(Usually, all these values are positive integers.). Then the objective is to select a subset of *N* such that the total profit of the selected items is maximized and the total weight does not exceed *c*.

Alternatively, a KP can be formulated as a solution of the following linear integer programming formulation:

$$K_B^{+} = \max\left\{ \sum_{j\in N} p_j x_j \,\middle|\, \sum_{j\in N} w_j x_j \le c,\ x_j \in B,\ j \in N \right\} \tag{2.1}$$

Minimization version of this problem is:

$$K_B^{-} = \min\left\{ \sum_{j\in N} p_j x_j \,\middle|\, \sum_{j\in N} w_j x_j \ge c,\ x_j \in B,\ j \in N \right\} \tag{2.2}$$

Here, we assume, without loss of generality, that;

$$p_j, w_j, c \in \mathrm{Z}^{+},\ w_j < c \text{ for all } j \in N, \text{ and } \sum_{j\in N} w_j \ge c\,.$$

We can generally subdivide knapsack problems into three types with respect to definition of decision variables:

1. Zero-One Knapsack Problems
2. Bounded Knapsack Problems
3. Integer (Unbounded) Knapsack Problems

Problems (2.1) and (2.2) are 0-1 knapsack problem.

In the given problem, the amount of each item may be limited. The corresponding problem is bounded KP in this case; there is $x_j \in BI, j \in N$ condition instead of $x_j \in B$. Consequently, we can take at most b_j copies of item j into our knapsack.

$$K_{BI}^{+(-)} = \max(\min)\left\{ \sum_{j\in N} p_j x_j \,\middle|\, \sum_{j\in N} w_j x_j \le (\ge) c,\ x_j \in BI,\ j \in N \right\}$$

It is used $x_j \in \mathrm{N}_0,\ j \in N$ condition instead of $x_j \in B$ in the third type KP.

$$K_I^{+(-)} = \max(\min)\left\{\sum_{j\in N} p_j x_j \,\middle|\, \sum_{j\in N} w_j x_j \le (\ge) c,\ x_j \in \mathrm{N}_0,\ j \in N\right\}$$

If there is only one constraint in the given KP, this problem is called "One dimensional knapsack problem"; the problems we discuss in this study are this type of KP.

Moreover, we will use the following orders in greedy algorithms for maximization and minimization problems, respectively:

$$\frac{p_1}{w_1} \ge \frac{p_2}{w_2} \ge \ldots \ge \frac{p_n}{w_n} \tag{2.3}$$

$$\frac{p_1}{w_1} \le \frac{p_2}{w_2} \le \ldots \le \frac{p_n}{w_n} \tag{2.4}$$

3. Zero-one knapsack problems

3.1. Zero-One maximization KP

Mathematical model of this problem was given by (2.1).

There are greedy algorithms with $\Delta \ge 1/2$ for 0-1 maximization KP. One of these algorithms is given below (Nuriyev, 1986). Here, variables are sorted according to (2.3).

Algorithm $A(A_B^+)$

A1) Find ***k*** with $\sum_{j=1}^{k} w_j \le c < \sum_{j=1}^{k+1} w_j$.

A2) $A_B^+ = \max\left\{\sum_{j=1}^{k} p_j, p_{k+1}\right\}$

A3) If $A_B^+ = \sum_{j=1}^{k} p_j$ then $\begin{cases} x_j = 1, & j = \overline{1,k} \\ x_j = 0, & j = \overline{k+1,n} \end{cases}$

If $A_B^+ = p_{k+1}$ then $\begin{cases} x_{k+1} = 1 \\ x_j = 0, & i \ne k+1 \end{cases}$

A4) END.

Theorem 3.1*: It holds for 0-1 maximization KP that*

$$\frac{1}{2} K_B^+ \le A_B^+ \le K_B^+ \tag{3.1}$$

3.2. Zero-One minimization KP

Mathematical model of this problem was shown by (2.2). We will search the following algorithm which is given in (Gens and Levner, 1980). Here, variables are sorted by (2.4).

Algorithm $A(A_B^-)$

A1) $A_B^- = \sum_{j=1}^{n} p_j,\ \ R = \varnothing$;

A2) $\overline{k} = \min\left\{l \,\middle|\, \sum_{j=1}^{l} w_j \ge c\right\}$;

A3) $L = \sum_{j=1}^{\bar{k}} p_j$;

A4) $A_B^- = \min\left\{A_B^-, L\right\}$, $N = N / \bar{k}$, $R = R \cup \{\bar{k}\}$;

A5) If $\sum_{j \in N} w_j \geq c$ then go to Step2;

A6) For $j \in N$ $\quad x_j = 1, \quad j = 1,..,\bar{k} - 1;$

$x_j = 0, \quad j = \bar{k} + 1,...,n;$

$x_{\bar{k}} = 1;$

For $j \in R / \{\bar{k}\}$ $x_j = 0$;

A7) END.

Theorem 3.2: *It holds for 0-1 minimization KP that*

$$K_B^- \leq A_B^- \leq 2K_B^- \tag{3.2}$$

3.3. Complementary problem for 0-1 KP

Minimization or maximization versions of some optimization problems can be considered in a similar way; one of them is known as the complementary problem of the other one (Güntzer and Jungnickel, 2000). Complementary problem of 0-1 KP is expressed as below (Nuriyev, 1986):

$$W = \sum_{j \in N} w_j, \quad \bar{c} = W - c, \quad y_j = 1 - x_{n-j+1} \quad \text{, and}$$

$$\mathrm{CK}_{\mathrm{B}}^{+} = \min\left\{\sum_{j \in N} p_j y_j \,\middle|\, \sum_{j \in N} w_j y_j \geq \bar{c},\ y_j \in B,\ j \in N\right\}$$

We have studied algorithms for 0-1 minimization KP with $\Delta \leq 2$ in the previous section . Here, we create a complementary problem of a 0-1 maximization KP. Then , we observe whether any improvement occurs in the preceding solution found by the algorithm. However, we must be careful about the orders, since order (2.3) is used for maximization problem but order (2.4) is used for the complementary problem of the same maximization problem.

4. Bounded knapsack problems

In this section, maximization and minimization versions of bounded integer knapsack problem (BKP) are studied. In these problems we will assume, without loss of generality, that

$$p_j, w_j, b_j \text{ and } \mathrm{c} \in Z^+ \text{ for } j \in N, \ \sum_{j \in N} b_j w_j \geq c, \ b_j w_j \leq c \text{ for } j \in N$$

4.1. Maximization BKP

Mathematical model of this problem is given below:

$$K_{BI}^{+} = \max\left\{\sum_{j \in N} p_j x_j \mid \sum_{j \in N} w_j x_j \leq c,\ 0 \leq x_j \leq b_j, x_j \in \mathrm{N}_0, j \in N\right\} \tag{4.1}$$

One of the algorithms proposed for this problem is given below and there is sorting (2.3) here.

Algorithm $A(A_{BI}^{+})$

A1) $k = 1$ and $A_{BI}^{+} = 0$;

A2) $x_k = \left\lfloor \frac{c}{w_k} \right\rfloor$;

A3) If $x_k > b_k$, then $x_k = b_k$ and $A_{BI}^+ = A_{BI}^+ + p_k x_k$; $c = c - w_k x_k$;

otherwise; $A_{BI}^+ = A_{BI}^+ + p_k x_k$; $c = c - w_k x_k$;

A4) $k = k + 1$;

A5) If $k > n$, then go to Step7;

A6) If $w_k \le c$ then go to Step2;

A7) $x_k = x_{k+1} = ... = x_n = 0$;

A8) $A_{BI}^+ = \max\left\{A_{BI}^+, \max_{j \in N}\{b_j p_j\}\right\}$

A9) If $A_{BI}^+ = \max_{j \in N}\left\{b_j p_j\right\}$, then let j* be the index that gives this result. In this case,

$x_{j^*} = b_{j^*}$ and $x_j = 0, \quad j \ne j^*$.

Otherwise, solution vector found before remains same.

A10) END**.**

Theorem 4.1: *Algorithm* $A(A_{BI}^+)$ *has a guarantee value of* $\frac{1}{2}$ **.**

$$\frac{1}{2} K_{BI}^+ \le A_{BI}^+ \le K_{BI}^+ \tag{4.2}$$

4.2. Minimization BKP

Mathematical model of this problem is given as below:

$$K_{BI}^- = \min\left\{\sum_{j \in N} p_j x_j \mid \sum_{j \in N} w_j x_j \ge c,\ 0 \le x_j \le b_j, x_j \in \mathrm{N}_0, j \in N\right\}$$

The algorithm we use for this problem is similar to $A(A_B^-)$. For this reason, we have to transform the above problem into 0-1 minimization KP. The followings show how to make this transformation.

We compose b_j copies for each item type j; so there are *m* variables in the resulting instance of (KP), that $m = \sum_{j \in N} b_j$. Let z_i denotes binary variable for i = 1,..,m; then distribution of these variables, profits and weights of these variables will be as below:

$x_1 \to z_1,\ z_2, ...,\ z_{b_1}$	$p_1 = d_1 = d_2 = ... = d_{b_1}$	$w_1 = a_1 = a_2 = ... = a_{b_1}$
$x_2 \to z_{b_1+1},\ z_{b_1+2}, ...,\ z_{b_1+b_2}$	$p_2 \to d_{b_1+1} = d_{b_1+2} = ... = d_{b_1+b_2}$	$w_2 \to a_{b_1+1} = a_{b_1+2} = ... = a_{b_1+b_2}$
. . . .	. . . .	. . . .
. . . .	. . . .	. . . .
. . . .	. . . .	. . . .
$x_n \to z_{b_1+b_2+...+b_{n-1}+1}, ...,\ z_m$	$p_n \to d_{b_1+b_2+...+b_{n-1}+1} = ... = d_m$	$w_n \to a_{b_1+b_2+...+b_{n-1}+1} = ... = a_m$

Eventually, we have variables, $z_i \in B$ ($j \in M = \{1, ..., m\}$); profit d_i and weight a_i of each variable. In this case, there must be new steps in the algorithm for transformation of variables in addition to the steps of algorithm $A(A_B^-)$. The resulting algorithm is below:

Algorithm $A(A_{BI}^{-})$

A1) $A_{BI}^{-} = \sum_{j=1}^{m} d_j, \;\; R = \varnothing$;

A2) $\bar{k} = \min\left\{ l \left| \sum_{j=1}^{l} a_j > c \right. \right\}$;

A3) $L = \sum_{j=1}^{\bar{k}} d_j$;

A4) $A_{BI}^{-} = \min\left\{ A_{BI}^{-}, L \right\}, \;\; M = M / \{\bar{k}\}, \;\; R = R \cup \{\bar{k}\}$;

A5) If $\sum_{j \in M} a_j \geq c$ then go to Step2;

A6) For $j \in M$, $z_j = 1, \;\; j = 1,..,\bar{k} - 1;$ $z_j = 0, \;\; j = \bar{k} + 1,...,m; \; z_{\bar{k}} = 1;$

For $j \in R / \{\bar{k}\}$ $z_j = 0$;

A7) *(Transformation of variables)*

A7-1) $i = 1, j = 1$;

A7-2) $x_i = \sum_{s=1}^{n_i} z_s$, $i = i+1$;

A7-3) If $i \leq n$ then go to Step (7-2);

A7-4) $x_j = x_j - \sum_{i=1}^{j-1} x_i$, $j = j+1$;

A7-5) If $j \leq n$ then go to Step(7-4);

A8) END.

Algorithm $A(A_{BI}^{-})$ has a guarantee value of 2 and it holds for this algorithm that

$$K_{BI}^{-} \leq A_{BI}^{-} \leq 2K_{BI}^{-} \tag{4.3}$$

4.3 Complementary problem for maximization BKP

The way we create complementary problem for maximization BKP is similar to the creation for 0-1 maximization KP. Complementary problem of maximization BKP is expressed as below:

$$W = \sum_{j \in N} b_j w_j, \;\; \bar{c} = W - c, \;\; y_j = \mathrm{b}_j - x_{n-j+1} \qquad \text{and,}$$

$$\mathrm{CK}_{\mathrm{BI}}^{+} = \min\left\{ \sum_{j \in N} p_j y_j \left| \sum_{j \in N} w_j y_j \geq \bar{c}, \; y_j \in BI, \;\; j \in N \right. \right\} \tag{4.4}$$

We have studied algorithms for minimization BKP with $\Delta \leq 2$ in the previous section. Here, we create a complementary problem of a maximization BKP and apply algorithm $A(A_{BI}^{-})$ to this new problem in order to obtain a better guarantee value. Then, we observe whether any improvement occurs in the preceding solution found by the algorithm.

4.4. Some theorems for BKP

Theorem 4.1: *Let* $\mathrm{P} = \sum_{j=1}^{n} b_j p_j$, *then* $K_{BI}^{+} + CK_{BI}^{+} = P$.

Theorem 4.2: $A_{BI}^{+} + CA_{BI}^{+} \leq P$

Theorem 4.3: $K_{BI}^{+} \geq f_{BI}^{*} \geq \begin{cases} (1/2)\, K_{BI}^{+}, & \text{if } \mu<1/3 \\ \left(2\mu/(1+\mu)\right)\cdot K_{BI}^{+}, & \text{if } \mu \geq 1/3 \end{cases}$

Here, $\mu = f_{BI}^{*}/P$.

Theorem 4.4: $CK_{BI}^{+} \leq CA_{BI}^{+} \leq \begin{cases} 2\cdot CK_{BI}^{+}, & \text{if } \lambda < 2/3 \\ \left(\lambda/(2\lambda-1)\right)\cdot CK_{BI}^{+}, & \text{if } \lambda \geq 2/3 \end{cases}$

Here, $\lambda = CA_{BI}^{+}/P$.

5. Conclusions

In this study, maximization and minimization versions of knapsack problems and greedy algorithms for these problems are discussed. Guarantee values of the algorithms are calculated; and complementary problems are created in order to improve known guarantee values. Time complexities of the given algorithms are O(nlogn) because of (2.3) and (2.4) sortings.

References

Gens, G. V.; Levner, E. V. 1980. *Efficient Approximate Algortihms for Combinatorial Problems.* The USSR Academy of Sciences, CEMI Press, Moscow, 66 p.

Güntzer, M. M.; Jungnickel, D. 2000. Approximate minimization algorithms for the 0/1 knapsack and subset-sum problem, *Operations Research Letters* 26: 55–66. doi:10.1016/S0167-6377(99)00066-8

Hromkovic, J. 2003. *Algorithms for Hard Problems*. Springer, Germany, 544 p.

Kellerer, H.; Pferschy, U.; Pisinger, D. 2004. *Knapsack Problems*. Springer, Berlin, 546 p.

Martello, S.; Toth, P. 1990. *Knapsack Problems*. John Wiley&Sons, England, 296 p.

Nikitin, A. I.; Nuriyev, U. G, 1983. On a method of the solution of the knapsack problem, *Kibernetika* 2:108–110.

Nuriyev, U. G. 1986. On the solution of the knapsack problem with guaranteed estimate, *Problems of Computing Mathematics and Theoretical Cybernetics* 66–70.

Vazirani, V. V. 2001. *Approximation Algorithms.* Springer, Berlin, 380 p.