ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2018, Vol. 12, No. 2, pp. 1–17. © Pleiades Publishing, Ltd., 2018.
Original Russian Text © A.B. Khutoretskii, S.V. Bredikhin, A.A. Zamyatin, 2018, published in Sibirskii Zhurnal Industrial'noi Matematiki, 2018, Vol. XXI, No. 2, pp. 108–121.

A Lexicographic 0.5-Approximation Algorithm for the Multiple Knapsack Problem

A. B. Khutoretskii[1*], S. V. Bredikhin[2], and A. A. Zamyatin[1***]**

[1] *Novosibirsk State University, ul. Pirogova 2, Novosibirsk, 630090 Russia*

[2] *Institute of Computational Mathematics and Mathematical Geophysics, pr. Akad. Lavrent'eva 6, Novosibirsk, 630090 Russia*

Received October 12, 2017

Abstract—We present a 0.5-approximation algorithm for the multiple knapsack problem (MKP). The algorithm uses an ordering of knapsacks according to the nondecreasing of size, and two orderings of items: in nonincreasing utility order and in nonincreasing order of the utility/size ratio. These orderings create two lexicographic orderings on the set $A \times B$ (here A is the set of knapsacks and B is the set of indivisible items). Based on each of these lexicographic orderings, the algorithm creates an admissible solution to MKP by looking through the pairs $(a, b) \in A \times B$ in the corresponding order and placing the item b into the knapsack a if this item is not placed yet and there is enough free space in the knapsack. The algorithm chooses the best of the two obtained solutions. This algorithm is 0.5-approximate and has running time $O(mn)$ (without sorting), where m and n are the sizes of the sets A and B correspondingly.

DOI: 10.1134/S1990478918020011

Keywords: *multiple knapsack problem, lexicographic ordering, approximation algorithm, approximation ratio*

INTRODUCTION

The Multiple Knapsack Problem (MKP) consists in the following (for example, see [1, 2]):

There is a set A of knapsacks, $|A| = m$, and a set B of indivisible items, $|B| = n$; it is necessary to place in the knapsacks the items of maximum total utility.

Such a problem arises, for example, when one needs to distribute the computing resource in a multiprocessor system.

The problem is NP-complete [3, Theorem 15.8], and therefore the approximate algorithms for solving it are of interest. In [4], it is proved that if $P \neq NP$ then there is no fully polynomial time approximation scheme (FPTAS) for MKP even for $m = 2$. A polynomial time approximation scheme (PTAS) for MKP

*E-mail: hab@dus.nsc.ru
**E-mail: bred@nsc.ru
***E-mail: stekbox@gmail.com

is also constructed there, which allows to construct a $(1-\varepsilon)$-approximate solution with the running time of $O(n^{O(\ln(1/\varepsilon)/\varepsilon^8)})$ (this estimate of the running time is given in [5]).

Looking through the pairs $(a, b) \in A \times B$ in some order and placing the item b in the knapsack a if the item has not yet been placed and there is enough space in the knapsack, it is possible to construct an admissible solution of MKP. Different orderings of pairs generate a family of greedy algorithms for MKP; we call them *packing*. The definition of packing can be easily modified for the problems with divisible items (for example, the linear relaxation of MKP): for each next pair (a, b) the maximum admissible part of the item a (taking the previous loading into account) is packed in the knapsack b. The total order relations $\prec_A$ and $\prec_B$ on the sets A and B, respectively, determine a lexicographic ordering on $A \times B$ and, hence, some packing. The special case of MKP for $m = 1$ is the knapsack problem (Knapsack Problem, KP). For the linear relaxation of KP, assuming that the size of every item is not larger than the size of the knapsack, the optimal solution (x^{LP} in the notation of [2]) can be obtained by the packing that uses the ordering of items according to nonincrease of the efficiency (the ratio of utility to size) [6].

If an algorithm $\mathbf{A}$, applied to a problem P from some class $\mathcal{P}$ of maximization problems, finds an admissible solution with the value of the objective function $\mathbf{A}(P)$, then its approximation ratio (worst-case performance ratio [1, p. 9]) for the class of problems $\mathcal{P}$ is the largest number ε such that $\mathbf{A}(P) \geq \varepsilon \operatorname{Opt}(P)$ for all $P \in \mathcal{P}$ (here $\operatorname{Opt}(P)$ is the optimal value of the objective function in the problem P). An algorithm with the approximation ratio ε is called *ε-approximate*. The well-known 0.5-approximate algorithm for KP [1] chooses the best of two admissible solutions. The first includes all items that are entirely included in x^{LP}, the second places in the knapsack only the item of maximum utility. In [7] an improvement of this algorithm is suggested: the second placement is constructed by packing with the ordering of items according to non-increase in utility.

Let $Q(a) > 0$ and $q(b) > 0$ be the sizes of the knapsack a and the item b, respectively. In [2, p. 299], the linear relaxation of MKP is considered with an additional condition: it is forbidden to place a nonzero part of an item b in a knapsack a if $Q(a) < q(b)$. It is also claimed there (without proof) that if we order the items by nonincreasing the efficiency, while the knapsacks, by nondecreasing the size then the packing using only the pairs (a, b) for which $q(b) \leq Q(a)$ gives an optimal solution x^* of this problem. We will prove an even more general result (Theorem 1): if the items are ordered according to nonincreasing efficiency and the set of admissible pairs is consistent with the ordering the knapsacks then the packing gives an optimal solution to the linear relaxation of MKP. The items entirely included in the placement x^* and the items partially included in it form two admissible solutions of MKP: x_1 and x_2. The algorithm that chooses the best of these solutions is 0.5-approximate [2, Theorem 10.4.2]. The time complexity of the algorithm is $O(mn)$ (without sorting).

The purpose of the article is to construct a new algorithm with the same estimates for accuracy and time complexity. The motivation is that, for a particular problem with many knapsacks, different algorithms generally yield different solutions from which we can choose the best.

We consider for MKP two packings with the ordering the knapsacks by nondecreasing size. The first packing orders the objects by nonincreasing the effectiveness, the second, by nonincreasing the utility. These packings generate the placements x_3 and x_4 which in the general case differ from the placements x_1 and x_2 (for KP, the placements x_1 and x_3 coincide). Corollary 5 (below) proves that a combined algorithm choosing the best of the placements x_3 and x_4 is 0.5-approximate. Thus, a new 0.5-approximate algorithm is proposed for MKP with the time complexity of $O(mn)$ (without sorting) which combines the ideas of the studies indicated above and generalizes the algorithm of [7] to the case $m > 1$. An algorithm for MKP with better characteristics is unknown to us.

1. FORMALIZATION OF THE PROBLEM

Let the coordinates of the vectors x, y, y_0, y_k, and z in $\mathbb{R}^{mn}$ be put into correspondence to the pairs $(a, b) \in A \times B$ and be denoted by $x(a, b)$, $y(a, b)$, $y_0(a, b)$, $y_k(a, b)$, and $z(a, b)$ respectively. We consider MKP in the following form:

$$V(x) = \sum_{b \in B} v(b) \sum_{a \in A} x(a, b) \to \max \tag{1}$$

$$\sum_{b \in B} q(b) x(a, b) \leq Q(a) \qquad \text{for all } a, \tag{2}$$

$$\sum_{a \in A} x(a, b) \leq 1 \qquad \text{for all } b, \tag{3}$$

$$x(a, b) \in \{0, 1\} \qquad \text{for all } a, b. \tag{4}$$

Here, $v(b) > 0$ is the utility of the item b, whereas the variable $x(a, b)$ is a "portion" of the item b placed in the knapsack a. Let $K(P)$ be the problem (1)–(4) with the set of parameters $P = \langle A, B, q(\cdot), v(\cdot), Q(\cdot) \rangle$; and let $X(P)$ be the set of all admissible solutions of this problem. The vector $x \in X(P)$ indicates an admissible placement: the item b is "placed" in the knapsack a if and only if $x(a, b) = 1$.

2. LEXICOGRAPHIC ALGORITHMS

For a finite set $M \neq \varnothing$ on which some total order relation $\prec$ is given, we denote by $\min(\prec, M)$ the first element of M (in the sense of the relation $\prec$). Let us describe two lexicographic algorithms (GI and TGI) for MKP.

2.1. Algorithm GI (Greedy Heuristic for Indivisible Items)

Input: a problem $K(P)$, a total order relation $\prec$ on the set $A \times B$;

Output: a vector $x \in X(P)$;

Begin {looking through the pairs $(a, b) \in A \times B$ according to the order $\prec$}

(a) $Q'(a) := Q(a)$ for all a; $q'(b) := q(b)$ for all b; $M := \varnothing$; {initialization; $Q'(a)$ is the unused size of knapsack a; $q'(b)$ is the nonplaced part of item b; and M is the set of pairs (a, b) for which the value $x(a, b)$ is defined }

(b) **while** $M \neq A \times B$ **do** {construction of vector x}

  find $(a_0, b_0) = \min(\prec, (A \times B) \setminus M)$; {beginning of the step (a_0, b_0)}

  if $q'(b_0) = q(b_0)$ **and** $q(b_0) \leq Q'(a_0)$ {item b_0 has not yet been placed and can be placed in knapsack a_0}

(c)   **then** $x(a_0, b_0) := 1$ {item b_0 is placed in knapsack a_0} **else** $x(a_0, b_0) := 0$

  endif;

  $Q'(a_0) := Q'(a_0) - x(a_0, b_0) q(b_0)$; $q'(b_0) := q'(b_0) - x(a_0, b_0) q(b_0)$; $M := M \setminus \{a_0, b_0\}$;

  {the value $x(a_0, b_0)$ is determined, the end of the step (a_0, b_0)}

endwhile

End.

It is clear that, at the beginning of the step (a_0, b_0) of the algorithm, we have

$$Q'(a_0) = Q(a_0) - \sum_{(a_0,b)\prec(a_0,b_0)} q(b)x(a_0, b), \tag{5}$$

$$q'(b_0) = q(b_0)\Big[1 - \sum_{(a,b_0)\prec(a_0,b_0)} x(a, b_0)\Big]. \tag{6}$$

Given $a \in A$, we put $B(a) = \{b \in B \mid q(b) \le Q(a)\}$ to be the set of all items that can be placed in the knapsack a. Using the idea of [7, p. 629], we formulate a "truncated" version of the GI algorithm allowing the placement of the item b_0 in the knapsack a_0 only if all items $b \in B(a_0)$ such that $(a_0, b) \prec (a_0, b_0)$ are placed in the knapsacks $a \preccurlyeq a_0$.

2.2. Algorithm TGI (Truncated Greedy Heuristic for Indivisible Items)

Input: a problem $K(P)$, a total order relation $\prec$ on the set $A \times B$;

Output: a vector $y \in X(P)$;

Begin {looking through the pairs $(a, b) \in A \times B$ according to the order $\prec$}

(a) $Q'(a)$:= $Q(a)$ for all a; $q'(b)$:= $q(b)$ for all b; A':= A; M:= $\varnothing$; {initialization; $Q'(a)$ is the unused size of knapsack a; $q'(b)$ is the unplaced part of item b; A' is the set of knapsacks which can be filled up; M is the set of pairs for which the value $y(a, b)$ is defined }

(b) **while** $M \ne A \times B$ **do** {construction of vector y}

  find $(a_0, b_0) = \min(\prec, (A \times B) \setminus M)$; {beginning of the step (a_0, b_0)}

  if ($a_0 \notin A'$ **or** $b_0 \notin B(a_0)$ **or** $q'(b_0) = 0$)

(c)   **then** $y(a_0, b_0)$:= 0 **else**

   if $q(b_0) \le Q'(a_0)$ {item b_0 can be placed in the knapsack a_0}

(d)   **then** $y(a_0, b_0)$:= 1 {we place item b_0 in the knapsack a_0}

(e)   **else begin** $y(a_0, b_0)$:= 0; $A' := A' \setminus \{a_0\}$ **end** { knapsack a_0 will not be filled anymore }

   endif

  endif

  $M := M \cup \{(a_0, b_0)\}$; {the value $y(a_0, b_0)$ is determined }

  $Q'(a_0)$:= $Q'(a_0) - y(a_0, b_0)q(b_0)$; $q'(b_0)$:= $q'(b_0) - q(b_0)y(a_0, b_0)$; {end of the step (a_0, b_0)}

 endwhile

End.

It is clear that, at the beginning of the step (a_0, b_0) of Algorithm TGI, equalities (5) and (6) for $x = y$ are fulfilled.

2.3. Properties of Algorithms GI and TGI

Let GI$(P,\prec)$ and TGI$(P,\prec)$ be implementations of Algorithms GI and TGI for a problem $K(P)$ respectively, which use an ordering $\prec$ of the set $A\times B$. Let the admissible solutions of the problem $K(P)$ constructed by Algorithms GI$(P,\prec)$ and TGI$(P,\prec)$ be denoted by $x(P,\prec)$ and $y(P,\prec)$, respectively. For $x\in X(P)$ and $(a_0,b)\in A\times B$, we put

$$r(x,a_0,b)=\sum_{(a,b)\preceq(a_0,b)} x(a,b).$$

It follows from (3) that $r(x,a_0,b)=1$ if in the placement x item b goes into the knapsack a for which $(a,b)\preceq(a_0,b)$, otherwise $r(x,a_0,b)=0$.

Lemma 1. *Let $y=y(P,\prec)$.*

(i) *$a_0\notin A'$ at the step (a_0,b_0) of Algorithm* TGI$(P,\prec)$ *if and only if there exists an item b_1 such that $(a_0,b_1)\prec(a_0,b_0)$, $b_1\in B(a_0)$, and $r(y,a_0,b_1)=0$.*

(ii) *If $a_0\notin A'$ at the step (a_0,b_0) of Algorithm* TGI$(P,\prec)$ *then there exists an item b such that $y(a_0,b)=1$.*

(iii) *$V(x(P,\prec))\geq V(y(P,\prec))$.*

Proof. If at the step (a_0,b_0) of Algorithm TGI $(P,\prec)$ we have $a_0\notin A'$ then at some preceding step (a_0,b_1) the algorithm executed a line with the label (e), $b_1\in B(a_0)$ and $r(y,a_0,b_1)=0$. Conversely, let $(a_0,b_1)\prec(a_0,b_0)$, $b_1\in B(a_0)$, and $r(y,a_0,b_1)=0$. Then $q'(b_1)=q(b_1)>0$ at the beginning of the step (a_0,b_1) and $y(a_0,b_1)=0$. The fixation $y(a_0,b_1)=0$ can take place only at the labels (c) and (e) of Algorithm TGI, whereas $a_0\notin A'$ at the step (a_0,b_1) in the first case and after this step, in the second case. In any case, $a_0\notin A'$ at the step (a_0,b_0). The statement (i) is proved.

If $a_0\notin A'$ at the step (a_0,b_0) then at some step $(a_0,b)\prec(a_0,b_0)$ Algorithm TGI comes to the label (e). Then $q(b)\leq Q(a_0)$, $q'(b)=q(b)$, and $q(b)>Q'(a_0)$; thus $Q'(a_0)<Q(a_0)$. Now (ii) follows from (5).

Let $x=x(P,\prec)$. Suppose that $r(x,a,b)\geq y(a,b)$ for all pairs $(a,b)\prec(a_0,b_0)$ (inductive hypothesis). This is also true for the pair (a_0,b_0) for $y(a_0,b_0)=0$. Suppose that $y(a_0,b_0)=1$. Then $b_0\in B(a_0)$ and $a_0\in A'$ at the step (a_0,b_0) of Algorithm TGI$(P,\prec)$. Let us demonstrate that $r(x,a_0,b_0)=1$. Put $B'(a_0)=\{b\in B(a_0)\mid(a_0,b)\prec(a_0,b_0)\}$ and choose $b\in B'(a_0)$. It follows from statement (i) that $r(y,a_0,b)=1$. If $y(a_0,b)=0$ then there exists a such that $(a,b)\prec(a_0,b)$ and $y(a,b)=1$. Then $r(x,a,b)=1$ by the induction hypothesis and $x(a_0,b)=0$ by (3). Hence, $y(a_0,b)\geq x(a_0,b)$ for all $b\in B'(a_0)$, thus

$$\sum_{(a_0,b)\prec(a_0,b_0)} x(a_0,b)q(b)\leq\sum_{(a_0,b)\prec(a_0,b_0)} y(a_0,b)q(b)\leq Q(a_0)-q(b_0). \tag{7}$$

The last inequality in (7) follows from $y\in X(P)$ and $y(a_0,b_0)=1$. If there is a pair $(a,b_0)\prec(a_0,b_0)$ for which $x(a,b_0)=1$ then $r(x,a_0,b_0)=1$. Let $x(a,b_0)=0$ for all $(a,b_0)\prec(a_0,b_0)$. Then at the step (a_0,b_0) of Algorithm GI$(P,\prec)$ we have $q'(b_0)=q(b_0)$; and $q(b_0)\leq Q'(a_0)$ follows from (5) and (7). Hence, Algorithm GI at the step (a_0,b_0) will go to the label (c) and set $x(a_0,b_0)=1$, whence $r(x,a_0,b_0)=1\geq y(a,b)$. By induction, $r(x,a,b)\geq y(a,b)$ for all a and b, from which assertion (iii) follows.

The proof of Lemma 1 is complete. □

Corollary 1. *Let $y = y(P, \prec)$. If $y(a_0, b_0) = 1$ then $r(y, a_0, b) = 1$ for all b such that*

$$(a_0, b) \prec (a_0, b_0), \qquad b \in B(a_0).$$

Proof. Let $y(a_0, b_0) = 1$, $b \in B(a_0)$, $(a_0, b) \prec (a_0, b_0)$, and $r(y, a_0, b) = 0$. Then, by the statement (i) of Lemma 1, $a_0 \notin A'$ at the step (a_0, b_0) of Algorithm TGI; the algorithm comes to the label (c) and sets $y(a_0, b_0) = 0$; a contradiction. □

3. MKP WITH DIVISIBLE ITEMS

We obtain the linear relaxation of problem (1)–(4) by substituting (4) with

$$x(a, b) \geq 0 \qquad \text{for all} \quad a, b. \tag{8}$$

It follows from (3) that $x(a, b) \leq 1$. Condition (8) means that, in knapsack a, the portion $x(a, b)$ of item b can be placed having the size $x(a, b)q(b)$ and the cost $x(a, b)v(b)$. In the case of one knapsack, the optimal solution of the problem (1)–(3), and (8) is constructed by a packing that uses the ordering of items according to efficiency [6].

Suppose that some assignments are forbidden: there is indicated a set $C \subseteq A \times B$ such that

$$x(a, b) = 0 \qquad if \;\; (a, b) \notin C. \tag{9}$$

The set of parameters P and the set C determine the problem (1)–(3), (8), and (9); we denote it by Problem $L(P, C)$. Let $Z(P, C)$ and $Z^*(P, C)$ be, respectively, the sets of all admissible and all optimal solutions of Problem $L(P, C)$. Given $x \in Z(P, C)$ and $(a_0, b_0) \in A \times B$, put

$$S(x, a_0) = \sum_b q(b)x(a, b), \qquad R(x, b_0) = \sum_a x(a, b).$$

Lemma 2. *If $x \in Z^*(P, C)$ and $(a_0, b_0) \in C$, then either $S(x, a_0) = Q(a_0)$ or $R(x, b_0) = 1$.*

Proof. For $(a_0, b_0) \in C$ and $x \in Z^*(P, C)$, we assume the contrary: $S(x, a_0) = Q(a_0) - \delta_1$ and $R(x, b_0) = 1 - \delta_2$, whereas $\delta_1 > 0$ and $\delta_2 > 0$. Put $\delta = \min\{\delta_1/q(b_0), \delta_2\} > 0$. Define $y \in \mathbb{R}^{mn}$ as follows: $y(a_0, b_0) = x(a_0, b_0) + \delta$ and $y(a, b) = x(a, b)$ for all $(a, b) \neq (a_0, b_0)$. It is easy to check that $y \in Z(P, C)$, whereas $V(y) = V(x) + \delta v(b_0) > V(x)$, which contradicts the choice of x.

This completes the proof of Lemma 2. □

Let us formulate a lexicographic algorithm for Problem $L(P, C)$:

3.1. Algorithm GD (Greedy Heuristic for Divisible Items)

Input: Problem $L(P, C)$, a total order relation $\prec$ on $A \times B$;

Output: a vector $z \in Z(P, C)$;

Begin { the algorithm looks through the pairs $(a, b) \in A \times B$ in accordance with the ordering $\prec$}

(a) $Q'(a) := Q(a)$ for all a; $q'(b) := q(b)$ for all b; $M := \varnothing$; {initialization; $Q'(a)$ is the unused size of knapsack a; $q'(b)$ is the size of the unplaced part of item b; M is the set of pairs (a, b) for which the value $z(a, b)$ is defined }

(b) **while** $M \neq A \times B$ **do** {construction of vector z}

find $(a_0, b_0) = \min(\prec, A \times B) \setminus M)$; {beginning of the step (a_0, b_0)}

if $(a_0, b_0) \in C$

(c) **then** $z(a_0, b_0) := \min\{q'(b_0), Q'(a_0)\}/q(b_0)$; {from the unplaced part of item b_0 we place in the knapsack a_0 the maximum that can be still put into it put into it }

else $z(a_0, b_0) := 0$

endif

$M := M \setminus \{(a_0, b_0)\}$; {the value $z(a_0, b_0)$ is determined }

$Q'(a_0) := Q'(a_0) - q(b_0)z(a_0, b_0)$; $q'(b_0) := q'(b_0) - q(b_0)z(a_0, b_0)$; {the end of the step (a_0, b_0)}

endwhile

End.

It is clear from the description of Algorithm GD that at the beginning of step (a_0, b_0) equalities (5) and (6) are satisfied for $x = z$. Let $\mathrm{GD}(P, C, \prec)$ be an implementation of Algorithm GD for Problem $L(P, C)$ using an ordering $\prec$ of the set $A \times B$; and let $z(P, C, \prec)$ be an admissible solution of the problem, constructed by Algorithm $\mathrm{GD}(P, C, \prec)$.

Lemma 3. *Let $z = z(P, C, \prec)$ and $(a_0, b_0) \in A \times B$. If $y \in Z(P, C)$ and $z(a, b) = y(a, b)$ for all $(a, b) \prec (a_0, b_0)$ then $z(a_0, b_0) \geq y(a_0, b_0)$.*

Proof. Let $z(a, b) = y(a, b)$ for $(a, b) \prec (a_0, b_0)$ and $y(a_0, b_0) > z(a_0, b_0) \geq 0$. Then $(a_0, b_0) \in C$. It follows from (5), (6) and $y \in Z(P, C)$ that, at the beginning of the step (a_0, b_0) of Algorithm $\mathrm{GD}(P, C, \prec)$, we have

$$
\begin{aligned}
Q'(a_0) &= Q(a_0) - \sum_{(a_0,b)\prec(a_0,b_0)} q(b)z(a_0, b) \\
&= Q(a_0) - \sum_{(a_0,b)\prec(a_0,b_0)} q(b)y(a_0, b) \geq q(b_0)y(a_0, b_0) > q(b_0)z(a_0, b_0),
\end{aligned}
$$

$$
\begin{aligned}
q'(b_0) &= q(b_0)\left[1 - \sum_{(a,b_0)\prec(a_0,b_0)} z(a, b_0)\right] \\
&= q(b_0)\left[1 - \sum_{(a,b_0)\prec(a_0,b_0)} y(a, b_0)\right] \geq q(b_0)y(a_0, b_0) > q(b_0)z(a_0, b_0).
\end{aligned}
$$

However, by construction, $q(b_0)z(a_0, b_0) = \min\{q'(b_0), Q'(a_0)\}$; a contradiction.

This completes the proof of Lemma 3. □

Next we will assume that the sets A and B are ordered by the relations $\prec_A$ and $\prec_B$ respectively. Let $\prec_{A,B}$ be a lexicographic ordering of $A \times B$ generated by the relations $\prec_A$ and $\prec_B$. For $b \in B$, we put $d(b) = v(b)/q(b)$ (efficiency of item b). The relation $\prec_B$ will be called a *d-ordering* if it orders the items by nonincreasing efficiency: $b_1 \prec_B b_2$ implies $d(b_1) \geq d(b_2)$ for all b_1 and b_2. The set $C \subseteq A \times B$ is consistent with the total order relation $\prec_A$ on A if, for every a_1, a_2, and b, it follows from $(a_1, b) \in C$ and $a_1 \prec_A a_2$ that $(a_2, b) \in C$.

Below, in Theorem 1, we give some sufficient conditions for the optimality of the vector $z(P, C, \prec)$ in Problem $L(P, C)$. Given $(a_0, b_0) \in A \times B$ and $z \in Z(P, C)$, we use the following notation:

$$N_1(z, a_0, b_0) = |\{b \in B \mid z(a_0, b) > 0,\ b_0 \prec_B b\}|,$$

$$N_2(z, a_0, b_0) = |\{a \in A \mid z(a, b_0) > 0,\ a_0 \prec_A a\}|,$$

$$N_3(z, a_0, b_0) = |\{a \in A \mid a \preceq_A a_0,\ S(z, a) < Q(a)\}|,$$

$$N_4(z, a_0, b_0) = |\{b \in B \mid b \preceq_B b_0,\ R(z, b) < 1\}|,$$

$$N(z, a_0, b_0) = \sum_{k=1}^{4} N_k(z, a_0, b_0).$$

Lemma 4. *Suppose that a relation $\prec_B$ is a d-ordering and the set C is consistent with a relation $\prec_A$. Put $x = z(P, C, \prec_{A,B})$. If $y \in Z^*(P, C)$ and $x(a, b) = y(a, b)$ for all $(a, b) \prec_{A,B} (a_0, b_0)$ then there is $z \in Z^*(P, C)$ such that it follows from $(a, b) \prec_{A,B} (a_0, b_0)$ that $z(a, b) = x(a, b)$ and either $z(a_0, b_0) = x(a_0, b_0)$ or $N(z, a_0, b_0) < N(y, a_0, b_0)$.*

Proof. Suppose that the conditions of the lemma are satisfied. If $x(a_0, b_0) = y(a_0, b_0)$ then the assertion of the lemma holds for $z = y$; therefore, we assume that $x(a_0, b_0) \neq y(a_0, b_0)$. Then $(a_0, b_0) \in C$ (since $x(a_0, b_0) = y(a_0, b_0) = 0$ for $(a_0, b_0) \notin C$) and $x(a_0, b_0) > y(a_0, b_0)$ by Lemma 3. Thus, we assume that

$$y(a_0, b_0) < x(a_0, b_0), \qquad y(a_0, b) = x(a_0, b) \qquad \text{for} \quad b \prec_B b_0. \tag{10}$$

Put $\delta_1 = x(a_0, b_0) - y(a_0, b_0) > 0$. It follows from Lemma 2 that $S(y, a_0) = Q(a_0)$ or $R(y, b_0) = 1$. Consider the following cases:

1. $S(y, a_0) = Q(a_0)$ and $R(y, b_0) = 1 - \delta_2$, where $\delta_2 > 0$. Then, by (2), $S(x, a_0) \leq S(y, a_0)$, and it follows from (10) that there is an item b_1 satisfying the conditions $b_0 \prec_B b_1$ and $y(a_0, b_1) > x(a_0, b_1)$. We put $\delta_3 = y(a_0, b_1)q(b_1)/q(b_0) > 0$ and $\delta = \min\{\delta_1, \delta_2, \delta_3\} > 0$. Define the vector z: $z(a, b) = y(a, b)$ if $(a, b) \notin \{(a_0, b_0), (a_0, b_1)\}$, $z(a_0, b_0) = y(a_0, b_0) + \delta$, and $z(a_0, b_1) = y(a_0, b_1) - \delta q(b_0)/q(b_1)$. It is not difficult to verify that $z \in Z(P, C)$. However, $V(z) = V(y) + \delta[v(b_0) - v(b_1)q(b_0)/q(b_1)] \geq V(y)$ since $b_0 \prec_B b_1$ yields $v(b_0)/q(b_0) \geq v(b_1)/q(b_1)$; hence, $z \in Z^*(P, C)$. In addition, $z(a, b) = x(a, b)$ for $(a, b) \prec_{A,B} (a_0, b_0)$, $N_k(z, a_0, b_0) \leq N_k(y, a_0, b_0)$ for $k \in \{1, 4\}$, and $N_k(z, a_0, b_0) = N_k(y, a_0, b_0)$ for $k \in \{2, 3\}$. If $\delta = \delta_1$ then $z(a_0, b_0) = x(a_0, b_0)$. If $\delta = \delta_2$ then $N_4(z, a_0, b_0) = N_4(y, a_0, b_0) - 1$ since $R(z, b_0) = 1$. If $\delta = \delta_3$ then $z(a_0, b_1) = 0$ and $N_1(z, a_0, b_0) = N_1(y, a_0, b_0) - 1$. Therefore, $z(a_0, b_0) = x(a_0, b_0)$ or $N(z, a_0, b_0) < N(y, a_0, b_0)$.

2. $S(y, a_0) = Q(a_0) - \delta_2$, where $\delta_2 > 0$, and $R(y, b_0) = 1$. By (3), $R(x, b_0) \leq R(y, b_0)$ and it follows from (10) that there exists a_1 for which $a_0 \prec_A a_1$ and $y(a_1, b_0) > x(a_1, b_0)$. Let $\delta = \min\{\delta_2/q(b_0), \delta_1, y(a_1, b_0)\} > 0$. Define a vector z: $z(a, b) = y(a, b)$ if $(a, b) \notin \{(a_0, b_0), (a_1, b_0)\}$; $z(a_0, b_0) = y(a_0, b_0) + \delta$, and $z(a_1, b_0) = y(a_1, b_0) - \delta$. It is easy to check that $z \in Z(P, C)$ and $V(z) = V(y)$, whence $z \in Z^*(P, C)$. Moreover, $z(a, b) = x(a, b)$ for $(a, b) \prec_{A,B} (a_0, b_0)$, $N_k(z, a_0, b_0) = N_k(y, a_0, b_0)$ for $k \in \{1, 4\}$, and $N_k(z, a_0, b_0) \leq N_k(y, a_0, b_0)$ for $k \in \{2, 3\}$. In the case of $\delta = \delta_2/q(b_0)$, we have $S(z, a_0) = Q(a_0)$ from which $N_3(z, a_0, b_0) = N_3(y, a_0, b_0) - 1$. If $\delta = \delta_1$ then $z(a_0, b_0) = x(a_0, b_0)$. On the other hand, if $\delta = y(a_1, b_0)$ then $z(a_1, b_0) = 0$, whence $N_2(z, a_0, b_0) = N_2(y, a_0, b_0) - 1$. Therefore, $z(a_0, b_0) = x(a_0, b_0)$ or $N(z, a_0, b_0) < N(y, a_0, b_0)$.

3. $S(y, a_0) = Q(a_0)$ and $R(y, b_0) = 1$. In this case, $S(y, a_0) \geq S(x, a_0)$ and $R(y, b_0) \geq R(x, b_0)$. By (10), there are a_1 and b_1 such that $a_0 \prec_A a_1$, $b_0 \prec_B b_1$, $y(a_0, b_1) > x(a_0, b_1) \geq 0$, and $y(a_1, b_0) > x(a_1, b_0) \geq 0$. Note that $(a_1, b_1) \in C$ since the set C is consistent with the relation $\prec_A$, $a_0 \prec_A a_1$ and $(a_0, b_1) \in C$ (because $y(a_0, b_1) > 0$). We choose $\delta = \min\{\delta_1,\ y(a_1, b_0),\ y(a_0, b_1)\} > 0$ and define the vector z: $z(a_0, b_0) = y(a_0, b_0) + \delta$, $z(a_1, b_0) = y(a_1, b_0) - \delta$, $z(a_0, b_1) = y(a_0, b_1) - \delta$, and $z(a_1, b_1) = y(a_1, b_1) + \delta$; if $(a, b) \notin \{(a_0, b_0), (a_1, b_0), (a_0, b_1), (a_1, b_1)\}$ then $z(a, b) = y(a, b)$. It is easy to see that $z \in Z(P, C)$ and $V(z) = V(y)$, whence $z \in Z^*(P, C)$. Moreover, $z(a, b) = x(a, b)$ for $(a, b) \prec_{A,B} (a_0, b_0)$, $N_k(z, a_0, b_0) = N_k(y, a_0, b_0)$ for $k \in \{3, 4\}$, and $N_k(z, a_0, b_0) \leq N_k(y, a_0, b_0)$ for $k \in \{1, 2\}$. If $\delta = x(a_0, b_0) - y(a_0, b_0)$ then $z(a_0, b_0) = x(a_0, b_0)$. If $\delta = y(a_1, b_0)$ then $z(a_1, b_0) = 0$ from which we have $N_2(z, a_0, b_0) < N_2(y, a_0, b_0)$. If $\delta = y(a_0, b_1)$ then $z(a_0, b_1) = 0$ and $N_1(z, a_0, b_0) < N_1(y, a_0, b_0)$. And in this case $z(a_0, b_0) = x(a_0, b_0)$ or $N(z, a_0, b_0) < N(y, a_0, b_0)$.

The proof of Lemma 4 is complete. □

Theorem 1. *If a set C is compatible with the relation $\prec_A$ and a d-ordering $\prec_B$ is given on B then Algorithm* GD $(P, C, \prec_{A,B})$ *constructs an optimal solution of Problem $L(P, C)$.*

Proof. Put $x = z(P, C, \prec_{A,B})$. If $y \in Z^*(P, C)$, $y \neq x$, and (a_0, b_0) is the first pair (a, b) (in the ordering $\prec_{A,B}$) for which $x(a, b) \neq y(a, b)$ then we denote by $\varphi(x, y)$ the vector z the existence of which is stated by Lemma 4. We arbitrarily choose $y_1 \in Z^*(P, C)$ and construct a sequence $Y = (y_1, y_2, \dots)$ of vectors from $Z^*(P, C)$ as follows: if the vector y_k is determined and $y_k = x$ then the construction is completed; otherwise, $y_{k+1} = \varphi(x, y_k)$.

Suppose that the sequence Y is infinite. Then $y_k \neq x$ for all $k \geq 1$. Let (a_k, b_k) be the first pair (a, b) in the ordering $\prec_{A,B}$ such that $y_k(a, b) \neq x(a, b)$. By Lemma 4, either $(a_{k+1}, b_{k+1}) = (a_k, b_k)$ and $N(y_{k+1}, a_{k+1}, b_{k+1}) < N(y_k, a_k, b_k)$, or $y_k(a, b) = x(a, b)$ for all $(a, b) \preceq_{A,B} (a_k, b_k)$ and, hence, $(a_k, b_k) \prec_{A,B} (a_{k+1}, b_{k+1})$. It is impossible since $N(y_k, a_k, b_k)$ are natural numbers and $A \times B$ is finite. Hence, the sequence Y is finite.

It is clear that its last element coincides with x; therefore, $x = z(P, C, \prec_{A,B}) \in Z^*(P, C)$. The proof of Theorem 1 is complete. □

A total order relation $\prec_A$ on A is called a *Q-ordering* if $a_1 \prec_A a_2$ implies $Q(a_1) \leq Q(a_2)$.

Corollary 2. *If on A there is given some Q-ordering $\prec_A$, while on B, some d-ordering $\prec_B$, and $C = \{(a, b) \mid q(b) \leq Q(a)\}$ then Algorithm* GD$(P, C, \prec_{A,B})$ *solves Problem $L(P, C)$.*

Proof. It is obvious that the set C is compatible with the relation $\prec_A$. It remains to apply Theorem 1. The proof is over. □

4. COMBINED ALGORITHMS

Let us formulate combined algorithms max GI and max TGI which, as will be shown, are 0.5-approximate for MKP. The total order relation $\prec_B$ on B is called a *v-ordering*, if $b_1 \prec_B b_2$ implies $v(b_1) \geq v(b_2)$.

Algorithm max GI

Input: a problem $K(P)$; a Q-ordering $\prec_A$ on A; a d-ordering $\prec_d$ on B; a v-ordering $\prec_v$ on B;

Output: an admissible solution x of a problem $K(P)$;

Begin

(a) $\prec_B:=\prec_d$; **execute** Algorithm GI$(P, \prec_{A,B})$; $x_1 := x(P, \prec_{A,B})$;

$\prec_B:=\prec_v$; **execute** Algorithm GI$(P, \prec_{A,B})$; $x_2 := x(P, \prec_{A,B})$;

if $V(x_1) \geq V(x_2)$ **then** $x := x_1$ **else** $x := x_2$

End.

It is known [1, p. 28; 7, Theorem 5] that, for one knapsack, Algorithm maxGI has an estimate 0.5 for the approximation ratio. The idea of this algorithm was used in [1, p. 166–167] when constructing a $1/(m+1)$-approximate algorithm for MKP.

Algorithm maxTGI is obtained from Algorithm maxGI by substituting the line with the label (a) by the following line:

$$\prec_B:=\prec_d; \textbf{ execute } \text{Algorithm TGI}(P, \prec_{A,B});\ x_1 := y(P, \prec_{A,B}).$$

4.1. Scheme of the Proof of the Main Result

Statement (iii) of Lemma 1 shows that, from the 0.5-approximation degree of Algorithm maxTGI there follows 0.5-approximation degree of Algorithm maxGI. Let $\prec_A$ be a Q-ordering, and let $\prec_B$ be some d-ordering. To prove 0.5-approximation degree of Algorithm maxTGI for the problem $K(P)$, we, using the vector $y = y(P, \prec_{A,B})$, construct a set B' of "additional" items. Then we will form an expanded set of items $B_1 = B \cup B'$, extend the functions $q(\cdot)$ and $v(\cdot)$ to the set B_1, and extend the relation $\prec_B$ to a d-ordering $\prec_{B1}$ of B_1.

Let $P_1 = \langle A, B_1, q(\cdot), v(\cdot), Q(\cdot)\rangle$. We will construct a set $C \subseteq A \times B_1$ consistent with the relation $\prec_A$ and including all pairs (a, b) from $A \times B$ such that $q(b) \leq Q(a)$. By Theorem 1, vector z constructed by Algorithm GD $(P_1, C, \prec_{A,B1})$ is the optimal solution of Problem $L(P_1, C)$. Let V_1^* and V^* be the optimal values of objective functions in Problems $L(P_1, C)$ and $K(P)$ respectively; and let the vector x be constructed by Algorithm GI$(P, \prec_{A,B})$ using the v-ordering of the set B as $\prec_B$. We will show that

$$y = (z(a,b) \mid (a,b) \in A \times B), \qquad V(x) \geq \sum_{b \in B'} v(b), \qquad V(x) + V(y) \geq V_1^* \geq V^*.$$

Hence, $\max\{V(x), V(y)\} \geq 0.5V^*$.

4.2. Construction of Problem $L(P_1, C)$

We choose a Q-ordering $\prec_A$ on A and a d-ordering $\prec_B$ on B. For $x \in X(P)$ and $a \in A$ put

$$B^+(x,a) = \{b \mid x(a,b) = 1\}, \qquad B^-(x,a) = \{b \in B(a) \mid r(x,a,b) = 0\}.$$

If $B^-(x,a) \neq \varnothing$, we put $\beta(a) = \min(\prec_B, B^-(x,a))$.

Lemma 5. *Let $y = y(P, \prec_{A,B})$. Then*

(i) *if $B^-(y,a) \neq \varnothing$ then $q(\beta(a)) > Q(a) - S(y,a)$;*

(ii) *if $B^-(y,a_1) \neq \varnothing$, $a_1 \prec_A a$, and $r(y,a,\beta(a_1)) = 0$ then $B^+(y,a) \neq \varnothing$ and $B^+(y,a) \cap B(a_1) = \varnothing$.*

Proof. (i) Let $B^-(y,a) \neq \varnothing$. By definition $\beta(a) \in B(a)$, $r(y,a,\beta(a)) = 0$ and $r(y,a,b) = 1$ for all $b \prec_B \beta(a)$ such that $b \in B(a)$. Moreover, $y(a,b) = 0$ for all b such that $\beta(a) \prec_B b$ (by Corollary 1). Then at the step $(a, \beta(a))$ of Algorithm TGI we have $a \in A'$ (according to the statement (i) of Lemma 1) and $Q'(a) = Q(a) - S(y,a)$. If $q(\beta(a)) \leq Q(a) - S(y,a)$ then Algorithm TGI sets $y(a,\beta(a)) = 1$ at the label (d), which contradicts $r(y,a,\beta(a)) = 0$.

(ii) Suppose that $B^-(y,a_1) \neq \varnothing$, $a_1 \prec_A a$, and $r(y,a,\beta(a_1)) = 0$. It follows from $r(y,a,\beta(a_1)) = 0$ that $y(a,\beta(a_1)) = 0$. Algorithm TGI will set $y(a,\beta(a_1)) = 0$ only if at the step $(a,\beta(a_1))$ one of the following conditions is fulfilled:

$$\text{(a) } a \notin A', \qquad \text{(b) } q(\beta(a_1)) > Q'(a), \qquad \text{(c) } \beta(a_1) \notin B(a), \qquad \text{(d) } q'(\beta(a_1)) = 0.$$

Conditions (c) and (d) are not satisfied since $\beta(a_1) \in B(a_1) \subseteq B(a)$ by definition and $r(y,a,\beta(a_1)) = 0$ by assumption.

$B^+(y,a) \neq \varnothing$ follows from condition (a) by the statement (ii) of Lemma 1 and from condition (b), by (5). Suppose $b \in B^+(y,a)$. Then $r(y,a,b) = 1$, $b \neq \beta(a_1)$ (since $r(y,a,\beta(a_1)) = 0$ by the hypothesis) and $r(y,a_1,b) = 0$ by (3). Taking into consideration the definition of $\beta(a_1)$, we have either $b \notin B(a_1)$ or $\beta(a_1) \prec_B b$. But $\beta(a_1) \prec_B b$ would imply $r(x,a,\beta(a_1)) = 1$ by Corollary 1, which contradicts the assumption. Hence, $b \notin B(a_1)$.

The proof of Lemma 5 is complete. □

Let $y = y(P, \prec_{A,B})$. For all a such that $B^-(y,a) \neq \varnothing$, we define the items $\delta(a)$ as follows: Suppose that the items $\delta(a)$ have been already chosen for all $a \prec_A a_0$ such that $B^-(y,a) \neq \varnothing$. If $\beta(a_0) \notin \{\delta(a) \mid a \prec_A a_0\}$ then we put $\delta(a_0) = \beta(a_0)$. If $\beta(a_0) = \delta(a_1)$ for some $a_1 \prec_A a_0$ then we find a knapsack a such that

$$a_1 \prec_A a \preceq_A a_0, \qquad B^+(y,a) \cap \{\delta(a') \mid a_1 \prec_A a' \prec_A a_0\} = \varnothing,$$

and select $\delta(a_0)$ from $B^+(y,a)$.

Let us proof the correctness of the preceding definition.

Lemma 6. *If $y = y(P, \prec_{A,B})$ and $B^-(y,a) \neq \varnothing$ then $\delta(a)$ is defined and either $\delta(a) = \beta(a)$ or $r(y,a,\delta(a)) = 1$.*

Proof. Let $y = y(P, \prec_{A,B})$ and $B^-(y,a_0) \neq \varnothing$. Suppose (the induction hypothesis) that the values $\delta(a)$ are defined for all $a \prec_A a_0$ such that $B^-(y,a) \neq \varnothing$, and either $\delta(a) = \beta(a)$ or $r(y,a,\delta(a)) = 1$. If $\beta(a_0) \notin \{\delta(a) \mid a \prec_A a_0\}$ then $\delta(a_0) = \beta(a_0)$. If $\beta(a_0) = \delta(a_1)$ for some $a_1 \prec_A a_0$ then

$$r(y,a_1,\delta(a_1)) \leq r(y,a_0,\beta(a_0)) = 0,$$

and, taking the induction assumption into account, $\delta(a_1) = \beta(a_1)$. Let

$$M_1 = \{a \mid a_1 \prec_A a \preceq_A a_0\}, \qquad M_2 = \{\delta(a) \mid a_1 \prec_A a \prec_A a_0\}.$$

It is clear that $|M_1| > |M_2|$. The sets $B^+(y,a)$ are pairwise disjoint; therefore, there is a knapsack $a \in M_1$ for which $B^+(y,a) \cap M_2 = \varnothing$, whereas $B^+(y,a) \neq \varnothing$ by the statement (ii) of Lemma 5. Thus, we can choose $\delta(a_0) \in B^+(y,a)$ according to the definition. It follows from $a \preceq_A a_0$ that $r(y,a_0,\delta(a_0)) = 1$. Lemma 6 is proved. □

The following describes the properties of the set of items $\delta(a)$ used in further reasoning:

Lemma 7. *If* $y = y(P, \prec_{A,B})$ *and* $B^-(y,a) \neq \varnothing$ *then*

*(*i*)* $\delta(a) \in B(a)$*;*

*(*ii*) the items* $\delta(a)$ *are pairwise different;*

*(*iii*)* $\delta(a) \preceq_B \beta(a)$*;*

*(*iv*)* $q(\delta(a)) > Q(a) - S(y,a)$.

Proof. (i) by Lemma 6, either $\delta(a) = \beta(a)$ or $r(y,a,\delta(a)) = 1$. In any case, $\delta(a) \in B(a)$.

(ii) Let $B^-(y,a_0) \neq \varnothing$. Let us show that $\delta(a_0) \notin \{\delta(a') \mid a' \prec_A a_0\}$.

By the definition of $\delta(a_0)$, it is true if $\delta(a_0) = \beta(a_0)$. On the other hand, if $\delta(a_0) \neq \beta(a_0)$ then there are a_1 and a_2 such that

$$\delta(a_0) \notin \{\delta(a) \mid a_1 \prec_A a \prec_A a_0\}, \tag{11}$$

$$a_1 \prec_A a_2 \preceq_A a_0, \qquad \beta(a_0) = \delta(a_1), \qquad \delta(a_0) \in B^+(y,a_2). \tag{12}$$

Therefore, $r(y,a_1,\delta(a_1)) \le r(y,a_2,\delta(a_1)) \le r(y,a_0,\beta(a_0)) = 0$ and $\delta(a_1) = \beta(a_1)$ by Lemma 6. Hence, $r(y,a,\beta(a_1)) \le r(y,a_0,\beta(a_1)) = r(y,a_0,\beta(a_0)) = 0$ and, by the statement (ii) of Lemma 5, $B^+(y,a_2) \cap B(a_1) = \varnothing$; therefore $\delta(a_0) \notin B(a_1)$. Owing to statement (i), $\delta(a) \in B(a) \subseteq B(a_1)$ for $a \preceq_A a_1$; therefore, $\delta(a_0) \notin \{\delta(a) \mid a \preceq_A a_1\}$. Taking (11) into account, this yields the statement (ii).

(iii) Suppose that $\delta(a) \neq \beta(a)$. Then, by the definition of $\delta(a)$, there exist a_1 and a_2 satisfying the conditions (11) and (12) for $a_0 = a$. By the statement (i), it follows from $\beta(a) = \delta(a_1)$ that $\beta(a) \in B(a_1) \subseteq B(a_2)$, whereas $r(y,a_2,\beta(a)) \le r(y,a,\beta(a)) = 0$. Applying Corollary 1 (while substituting a_0, b_0, and b with a_2, $\delta(a)$, and $\beta(a)$ respectively), we obtain $\delta(a) \prec_B \beta(a)$.

(iv) If $\delta(a) = \beta(a)$ then statement (iv) is equivalent to statement (i) of Lemma 5. On the other hand, if $\delta(a) \neq \beta(a)$ then there exist a_1 and a_2 satisfying conditions (11) and (12) for $a_0 = a$. Then it follows from $r(y,a,\beta(a)) = 0$ that $r(y,a_1,\delta(a_1)) = r(y,a_2,\beta(a)) = 0$; and $\beta(a) = \delta(a_1) = \beta(a_1)$ by Lemma 6. Hence, $r(y,a_2,\beta(a_1)) \le r(y,a,\beta(a) = 0$. Then $B^+(y,a_2) \cap B(a_1) = \varnothing$ by the statement (ii) of Lemma 5; and it follows from $\delta(a) \in B^+(y,a_2)$ that $\delta(a) \notin B(a_1)$. Therefore, using the statement (i) of Lemma 5, from $\beta(a) = \delta(a_1) \in B(a_1)$ we have

$$q(\delta(a)) > Q(a_1) \ge q(\beta(a)) > Q(a) - S(y,a).$$

The proof of Lemma 7 is complete. □

Let $y = y(P, \prec_{A,B})$ and $A_1 = \{a \in A \mid B^-(y,a) \neq \varnothing\}$. Given $a \in A_1$, we introduce a new item $\gamma(a)$ with the parameters $q(\gamma(a)) = Q(a) - S(y,a)$ and $v(\gamma(a)) = q(\gamma(a)) \cdot d(\delta(a))$. The definition of $v(\gamma(a))$ is correct since for $a \in A_1$ the item $\delta(a)$ exists (by Lemma 6).

Put $B' = \{\gamma(a) \mid a \in A_1\}$ and $B_1 = B \cup B'$. The functions $q(\cdot)$ and $v(\cdot)$ are now defined on B_1. Define the relation $\prec_{B1}$ on B_1 as follows: on B the relation $\prec_{B1}$ coincides with $\prec_B$, and the item $\gamma(a)$ immediately precedes $\delta(a)$.

If $A_1 = \varnothing$ then the vector y is the optimal solution of Problem $K(P)$ because all items are placed. Taking this into account, we assume that $A_1 \neq \varnothing$. It is obvious that the relation $\prec_{B1}$ totally orders the set B_1. We will show that this is some d-ordering.

Lemma 8. *The relation* $\prec_{B1}$ *orders* B_1 *according to nonincreasing item efficiency.*

Proof. Let $d(b) = v(b)/q(b)$ for $b \in B_1$. Let us show that

$$b_1 \prec_{B1} b_2 \to d(b_1) \geq d(b_2) \qquad \text{for all } b_1 \text{ and } b_2 \text{ from } B_1. \tag{13}$$

It is obvious that condition (13) is fulfilled for b_1 and b_2 from B. If $b = \gamma(a)$ then

$$d(b) = v(\gamma(a))/q(\gamma(a)) = d(\delta(a));$$

therefore, the placement $\gamma(a)$ immediately before $\delta(a)$ preserves (13). The proof is over. □

So, we have a Q-ordering $\prec_A$ on A, an extended set of items B_1 with a d-ordering $\prec_{B1}$, and functions $q(\cdot)$ and $v(\cdot)$ defined on B_1. Put

$$C = \{(a,b) \in A \times B \mid q(b) \leq Q(a)\} \cup \{(a, \gamma(a')) \mid a' \preceq_A a\}, \qquad P_1 = \langle A, B_1, q(\cdot), v(\cdot), Q(\cdot)\rangle.$$

Problem $L(P_1, C)$ is now defined.

4.3. Properties of the Optimal Solution of Problem $L(P_1, C)$

Given Problem $L(P_1, C)$, let $Z(P_1, C)$ and $V_1(\cdot)$ be the set of admissible solution and the objective function respectively. Let V^* and V_1^* be optimal values of objective functions in Problems $K(P)$ and $L(P_1, C)$ respectively.

Put $u_0 = z(P_1, C, \prec_{A,B1}) = (u_0(a,b) \mid (a,b) \in A \times B_1)$.

Lemma 9. *The relations $V^* \leq V_1^* = V_1(u_0)$ are fulfilled.*

Proof. Let $x \in X(P)$ and $u(x) = (u(a,b) \mid (a,b) \in A \times B_1)$, where $u(a,b) = x(a,b)$ for $b \in B$ and $u(a,b) = 0$ for $b \in B'$. Obviously, $u(x) \in Z(P_1, C)$ and $V_1(u(x)) = V(x)$; and so, $V^* \leq V_1^*$.

Let $(a,b) \in C$ and $a \prec_A a_1$. If $b \in B$ then $q(b) \leq Q(a) \leq Q(a_1)$; therefore, $(a_1, b) \in C$. On the other hand, if $b = \gamma(a')$ then, by definition of C, it follows from $(a,b) \in C$ that $a' \preceq_A a$. Then $a' \prec_A a_1$ and $(a_1, b) \in C$ by definition. Hence, C is consistent with the relation $\prec_A$. Taking Lemma 8 into account, it follows from Theorem 1 that u_0 is an optimal solution of Problem $L(P_1, C)$.

This completes the proof of Lemma 9. □

Given $x \in X(P)$, $u = (u(a,b) \mid (a,b) \in A \times B_1)$ and $a_0 \in A$, we introduce

$$s(x, a_0, b_0) = \sum_{\{b \in B \mid b \prec_B b_0\}} q(b)x(a_0, b), \qquad s_1(u, a_0, b_0) = \sum_{\{b \in B_1 \mid b \prec_{B1} b_0\}} q(b)u(a_0, b)$$

and put $y_0 = y(P, \prec_{A,B})$.

Theorem 2. (i) *For all $(a,b) \in A \times B$, the equality $u_0(a,b) = y_0(a,b)$ holds;*

(ii) *For all $a \in A_1$ the equality $u_0(a, \gamma(a)) = 1$ is fulfilled.*

Proof. At step (a_0, b_0) of Algorithm $\mathrm{GD}(P_1, C, \prec_{A,B1})$, the values $u_0(a,b)$ are already defined for all pairs $(a,b) \prec_{A,B1} (a_0, b_0)$.

Let us assume (the induction hypothesis) that for all these pairs the following conditions are satisfied: if $b \in B$ then $u_0(a,b) = y_0(a,b)$, while if $a \in A_1$ and $b = \gamma(a)$ then $u_0(a,b) = 1$. Under this assumption,

$$u_0(a_0, \gamma(a)) = 0 \qquad \text{if } \gamma(a) \preceq_{B1} b_0, \ a \neq a_0. \tag{14}$$

Indeed, let $\gamma(a) \preceq_{B1} b_0$, $a \neq a_0$. If $a_0 \prec_A a$ then $(a_0, \gamma(a)) \notin C$ and $u_0(a_0, \gamma(a)) = 0$ in accordance with Algorithm GD. If $a \prec_A a_0$ then $(a, \gamma(a)) \prec_{A,B1} (a_0, b_0)$, $u_0(a, \gamma(a)) = 1$ by the induction hypothesis and $u_0(a_0, \gamma(a)) = 0$ by (3).

(i) Let $b_0 \in B$. If $b \prec_{B1} b_0$ and $b \in B$ then $(a_0, b) \prec_{A,B1} (a_0, b_0)$ and, by the induction assumption, $u_0(a_0, b) = y_0(a_0, b)$. Above we defined the items $\gamma(a)$ only for $a \in A_1$. Further it is convenient to assume that for each $a \in A \setminus A_1$ an item $\gamma(a)$ is also defined so that $q(\gamma(a)) = 0$. Then, taking (14) into account, we can write

$$s_1(u_0, a_0, b_0) = q(\gamma(a_0)) \cdot u_0(a_0, \gamma(a_0)) + s(y_0, a_0, b_0). \tag{15}$$

From this, $s_1(u_0, a_0, b_0) \leq q(\gamma(a_0)) + s(y_0, a_0, b_0)$. By definition, $q(\gamma(a_0)) = 0$ if $a_0 \notin A_1$, and $q(\gamma(a_0)) = Q(a_0) - S(y_0, a_0)$ if $a_0 \in A_1$. Moreover, $Q(a_0) \geq S(y_0, a_0)$ by (2). Then for every $a \in A$ we have

$$Q(a_0) - s_1(u_0, a_0, b_0) \geq S(y_0, a_0) - s(y_0, a_0, b_0). \tag{16}$$

If $b_0 \notin B(a_0)$ then $u_0(a_0, b_0) = y_0(a_0, b_0) = 0$ by construction. If $y_0(a, b_0) = 1$ for some $a \prec_A a_0$ then $u(a, b_0) = 1$ by the induction assumption and, by condition (3), $u_0(a_0, b_0) = y_0(a_0, b_0) = 0$. Consider the case $b_0 \in B(a_0)$ and $y_0(a, b_0) = 0$ for all $a \prec_A a_0$. Then, taking the induction assumption into account,

$$\sum_{a \prec_A a_0} y_0(a, b_0) = \sum_{a \prec_A a_0} u_0(a, b_0) = 0. \tag{17}$$

Suppose that $y_0(a_0, b_0) = 1$. Then $b_0 \in B(a_0)$, $(a_0, b_0) \in C$ and $S(y_0, a_0) \geq s(y_0, a_0, b_0) + q(b_0)$. By (16), $Q(a_0) - s_1(u_0, a_0, b_0) \geq q(b_0)$.

In combination with (17), this guarantees that Algorithm GD will set $u_0(a_0, b_0) = 1 = y_0(a_0, b_0)$ at step (a_0, b_0).

We assume now that $y_0(a_0, b_0) = 0$. Then $r(y_0, a_0, b_0) = 0$, and it follows from $b_0 \in B(a_0)$ that $b_0 \in B^-(y_0, a_0)$; so $a_0 \in A_1$ and the items $\beta(a_0)$, $\delta(a_0)$ are defined. From the definition of $\prec_{B1}$, assertion (iii) of Lemma 7, and the definition of $\beta(a)$, we infer

$$\gamma(a_0) \prec_{B1} \delta(a_0) \preceq_B \beta(a_0) \preceq_B b_0.$$

Then $(a_0, \gamma(a_0)) \prec_{A,B1} (a_0, b_0)$ and $u_0(a_0, \gamma(a_0)) = 1$ by the induction assumption. It follows from (15) that $s_1(u_0, a_0, b_0) = q(\gamma(a_0)) + s(y_0, a_0, b_0)$; then $Q(a_0) - s_1(u_0, a_0, b_0) = S(y_0, a_0) - s(y_0, a_0, b_0)$. However, $y_0(a_0, b_0) = 0$ implies $S(y_0, a_0) = s(y_0, a_0, b_0)$ (Corollary 1); therefore $Q(a_0) = s_1(u_0, a_0, b_0)$. The line labeled (c) in Algorithm GD will give $u_0(a_0, b_0) = 0$ at step (a_0, b_0).

(ii) Assume that $a_0 \in A_1$ and $b_0 = \gamma(a_0)$. If $a \prec_A a_0$ then $(a, b_0) \notin C$ and $u_0(a, b_0) = 0$. By (14) and the induction hypothesis, we have

$$s_1(u_0, a_0, b_0) = \sum_{\{b \in B \mid b \prec_B b_0\}} q(b) u(a_0, b) = s(y_0, a_0, b_0) \leq S(y_0, a_0).$$

By definition, $q(\gamma(a_0)) = Q(a_0) - S(y_0, a_0) \leq Q(a_0) - s_1(u_0, a_0, b_0)$. Then the line labeled (c) of Algorithm GD gives $u_0(a_0, b_0) = 1$ at step (a_0, b_0).

The proof of Theorem 2 is complete. □

Corollary 3. $V_1^* \leq V(y_0) + \sum\limits_{a \in A_1} v(\delta(a))$.

Proof. Given $a \in A_1$, we have by definition

$$v(\gamma(a)) = q(\gamma(a)) \cdot d(\delta(a)) = \frac{Q(a) - S(y,a)}{q(\delta(a))}\, v(\delta(a)).$$

Therefore, it follows from (iv) of Lemma 7 that $v(\gamma(a)) < v(\delta(a))$. Using Lemma 9, Theorem 2, and the assertion (iv) of Lemma 7, we infer

$$V_1^* = V_1(u_0) = \sum_{b\in B} v(b) \sum_{a\in A} u_0(a,b) + \sum_{b\in B'} v(b) \sum_{a\in A} u_0(a,b))$$
$$= \sum_{b\in B} v(b) \sum_{a\in A} y_0(a,b) + \sum_{a\in A_1} v(\gamma(a)) \le V(y_0) + \sum_{a\in A_1} v(\delta(a)).$$

The proof is complete. □

4.4. The Main Result

Let some Q-ordering $\prec_A$ be given on A, and let a v-ordering $\prec_B$ be given on B. For each $a \in A_1$ we construct an item $\varphi(a)$ as follows: If the items $\varphi(a') \in B(a')$ are defined for all $a' \in A_1$ such that $a' \prec_A a$ then we put

$$\Phi(a) = \{\varphi(a') \mid a' \in A_1, a' \prec_A a\}, \qquad \varphi(a) = \min(\prec_B, B(a) \setminus \Phi(a)),$$
$$D(a) = \{\delta(a') \mid a' \in A_1, a' \prec_A a\}.$$

The following proves the correctness of the definition of $\varphi(a)$:

Lemma 10. *If $a \in A_1$ then the value of $\varphi(a)$ is defined.*

Proof. It is clear that $|\Phi(a)| \le |\{a' \in A_1 \mid a' \prec_A a\}|$. It follows from the statement (ii) of Lemma 7 that

$$|D(a)| = |\{a' \in A_1 \mid a' \prec_A a\}|, \qquad \delta(a) \notin D(a).$$

The statement (i) of Lemma 7 and the definition of Q-ordering yield $D(a) \cup \{\delta(a)\} \subseteq B(a)$. Then

$$|\Phi(a)| \le |D(a)| < |B(a)|, \qquad B(a) \setminus \Phi(a) \ne \varnothing.$$

Lemma 10 is proved. □

Lemma 11. *Let $\Phi = \{\varphi(a) \mid a \in A_1\}$ and $D = \{\delta(a) \mid a \in A_1\}$. There exists an isomorphism $\pi : D \mapsto \Phi$ such that $v(\pi(b)) \ge v(b)$ for all $b \in D$.*

Proof. Let there be defined an isomorphism $\pi : D(a_0) \mapsto \Phi(a_0)$ satisfying the condition $v(\pi(b)) \ge v(b)$ for all $b \in D(a_0)$ (the induction hypothesis). Put

$$\Phi_1(a_0) = \{\varphi(a) \mid a \prec_A a_0,\ \varphi(a) \prec_B \varphi(a_0)\}$$

and prove that $\Phi_1(a_0) = \{b \in B(a_0) \mid b \prec_B \varphi(a_0)\}$. Indeed, if $b \in B(a_0)$ and $b \prec_B \varphi(a_0)$ then $b = \varphi(a)$ for some $a \prec_A a_0$ by the definition of $\varphi(a_0)$; whence $b \in \Phi_1(a_0)$. Conversely, if $b \in \Phi_1(a_0)$ then there is $a \prec_A a_0$ for which $b = \varphi(a) \prec_B \varphi(a_0)$, whereas $b \in B(a) \subseteq B(a_0)$. Consequently,

$$\Phi_1(a_0) = \{b \in B(a_0) \mid b \prec_B \varphi(a_0)\}.$$

Denote $D_1(a_0) = \pi^{-1}(\Phi_1(a_0))$, $D_1^+ = D_1(a_0) \cup \{\delta(a_0)\}$ and $\Phi_1^+ = \Phi_1(a_0) \cup \{\varphi(a_0)\}$. The sets $\Phi_1(a_0)$ and $D_1(a_0)$ are of equal cardinality, and so we put $k = |\Phi_1(a_0)| = |D_1(a_0)|$. Let us enumerate D_1^+ and Φ_1^+ in accordance with the ordering $\prec_B$:

$$D_1^+ = \{b_1^1, \dots, b_{k+1}^1\}, \qquad \Phi_1^+ = \{b_1^2, \dots, b_{k+1}^2\}.$$

The items $\delta(a)$ are pairwise different by the assertion (ii) of Lemma 7; therefore, D_1^+ includes $k+1$ element from $B(a_0)$. The items $\varphi(a)$ are pairwise distinct by construction, so Φ_1^+ includes $k+1$ first elements from $B(a_0)$ (in the ordering $\prec_B$). Then $v(b_i^1) \le v(b_i^2)$ for all $i \in \{1, \dots, k+1\}$.

Without changing $\pi(b)$ for $b \in D(a_0) \setminus D_1(a_0)$, we redefine the correspondence π on $D_1(a_0)$ and extend it to $\delta(a_0)$: $\pi(b_i^1) = b_i^2$. Now π maps the set $D(a_0) \cup \{\delta(a_0)\}$ onto $\Phi(a_0) \cup \{\varphi(a_0)\}$. One-to-oneness of the mapping is preserved.

Lemma 11 is proved. □

Corollary 4. *The following inequality holds:*

$$\sum_{a \in A_1} v(\delta(a)) \le \sum_{a \in A_1} v(\varphi(a)).$$

Proof. Let $\pi : D \mapsto \Phi$ be the map constructed in Lemma 11. Then

$$\sum_{a \in A_1} v(\delta(a)) = \sum_{b \in D} v(b) \le \sum_{b \in D} v(\varphi(b)) = \sum_{b \in \Phi} v(b).$$

Corollary 4 is proved. □

We show that all items $\varphi(a)$ are included in the placement $x(P, \prec_{A,B})$:

Lemma 12. *Given $x = x(P, \prec_{A,B})$, we have $r(x, a, \varphi(a)) = 1$ for all $a \in A_1$.*

Proof. Let $r(x, a, \varphi(a)) = 1$ for all a from A_1 be such that $a \prec_A a_0$ (the induction hypothesis). Denote

$$\Phi^-(a_0) = B(a_0) \setminus \Phi(a_0), \qquad M(a_0) = \{b \in B(a_0) \mid \sum_{a \prec a_0} x(a, b) = 0\}, \qquad b_0 = \min(\prec_B, M(a_0)).$$

Suppose that $r(x, a_0, \varphi(a_0)) = 0$. Then $\varphi(a_0) \in M(a_0)$. It follows from the induction hypothesis that $b_0 \in \Phi^-(a_0)$. But $\varphi(a_0) = \min(\prec_B, \Phi^-(a_0))$; therefore, $\varphi(a_0) = b_0$. Hence $x(a, \varphi(a_0)) = 0$ for all $a \prec_A a_0$ and $x(a_0, b) = 0$ for all $b \prec_B b_0$ from $B(a_0)$ (since $\sum_{a \prec a_0} x(a, b) = 1$ for such b).

Then it follows from (5) and (6) that at step $(a_0, \varphi(a_0))$ of Algorithm GI$(P, \prec_{A,B})$ we will have $q'(\varphi(a_0)) = 0$ and $Q'(a_0) = Q(a_0)$; in result of this the algorithm will come to the line labeled by (c) and put $x(a_0, \varphi(a_0)) = 1$; which contradicts the assumption $r(x, a_0, \varphi(a_0)) = 0$. □

Theorem 3. *Algorithm* maxTGI *constructs a* 0.5*-approximate solution of the multiple knapsack problem.*

Proof. Successively applying Lemma 9, Corollary 3, Corollary 4, and Lemma 12, we arrive at

$$V^* \le V_1^* \le V(y_0) + \sum_{a \in A_1} v(\delta(a)) \le V(y_0) + \sum_{a \in A_1} v(\varphi(a)) \le V(y_0) + V(x_0).$$

Here, V^* is the optimal value of the objective function for MKP (1)–(4), whereas $V(y_0)$ and $V(x_0)$ are the values of this function at the vectors constructed by Algorithm TGI with a d-ordering on B and by Algorithm GI with some v-ordering on B, under a Q-ordering of the set A in both cases. Theorem 3 is proved. □

Finally, by the statement (iii) of Lemma 1, the 0.5-approximation degree of Algorithm maxTGI implies the 0.5-approximation degree of Algorithm maxGI.

Corollary 5. *Algorithm* maxGI *constructs a* 0.5-*approximate solution of the multiple knapsack problem.*

CONCLUSION

A new 0.5-approximate algorithm is constructed for the multiple knapsack problem, which can be used, for example, when distributing the computing resource in a multiprocessor system. The algorithm has time complexity of $O(mn)$ (without sorting) and generalizes the algorithm from [7] to the case $m > 1$. The algorithm, described in [2, p. 299], have similar estimates of approximation ratio and time complexity.

In general, these algorithms can generate different distributions; therefore, in applications it is expedient to apply both algorithms and choose the best of the obtained results.

ACKNOWLEDGMENTS

The authors are grateful to the participants of the seminar "Discrete Extremal Problems" at Sobolev Institute of Mathematics (Novosibirsk) for useful remarks and recommendations.

REFERENCES

1. S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations* (John Wiley & Sons, Chichester, N. Y., 1990).
2. H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems* (Springer, Berlin, 2004).
3. C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice Hall, Englewood Cliffs, 1982).
4. C. Chekuri and S. Khanna, "A Polynomial Time Approximation Scheme for the Multiple Knapsack Problem," SIAM J. Computing **5** (3), 713–728 (2003).
5. K. Jansen, "Parameterized Approximation Scheme for the Multiple Knapsack Problem," SIAM J. Computing **39** (4), 1392–1412 (2009).
6. G. D. Dantzig, "Discrete Variable Extremum Problems," Oper. Res. **5** (2), 266–277 (1957).
7. A. Mualem and N. Nisan, "Truthful Approximation Mechanisms for Restricted Combinatorial Auctions," Games and Economic Behavior **64** (2), 612–631 (2008).