УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

CONTROL IN TECHNICAL SYSTEMS

UDC 519.852.6

AN INNOVATIVE APPROXIMATE SOLUTION METHOD FOR AN INTEGER PROGRAMMING PROBLEM

Mamedov K. Sh. – Dr. Sc., Professor, Professor of Baku State University, Head of the Department, Institute of Control Systems, Azerbaijan.
Niyazova R. R. – Doctorant and Scientist, Institute of Control Systems, Azerbaijan.

ABSTRACT

Context. There are certain methods for finding the optimal solution to integer programming problems. However, these methods cannot solve large-scale problems in real time. Therefore, approximate solutions to these problems that work quickly have been given. It should be noted that the solutions given by these methods often differ significantly from the optimal solution. Therefore, the problem of taking any known approximate solution as the initial solution and improving it further arises.

Objective. Initially, a certain approximate solution is found. Then, based on proven theorems, the coordinates of this solution that do not coincide with the optimal solution are determined. After that, new solutions are found by sequentially changing these coordinates. The one that gives the largest value to the functional among these solutions is accepted as the final solution.

Method. The method we propose in this work is implemented as follows:

First, a certain approximate solution to the problem is established, then the numbers of the coordinates of this solution that do not coincide with the optimal solution are determined. After that, new solutions are established by sequentially assigning values to these coordinates one by one in their intervals. The best of the solutions found in this process is accepted as the final innovative solution.

Results. A problem was solved in order to visually illustrate the quality and effectiveness of the proposed method.

Conclusions. The method we propose in this article cannot give worse results than any approximate solution method, is simple from an algorithmic point of view, is novel, can be easily programmed, and is important for solving real practical problems.

KEYWORDS: integer programming problem, initial approximate solution, the interval in which the coordinates of the approximate solution may differ from the optimal solution, innovative approximate solution, computational experiments.

NOMENCLATURE

N is a number of issues resolved;

n is a number of unknowns;

a_{ij} is a given positive integer;

c_j is a given positive integer;

d_j is a positive integer;

b_i is a given positive integers;

x_j is an j-th unknown;

j_* is a value of the unknown;

X^0 is an initial approximate solution;

r is an initial solution is the number of the first coordinate that received a value of “0” when constructing the solution;

f^0 is a value of the approximate solution to the objective function;

$\widetilde{X}$ is a solution found when the unknowns in the problem are not required to be integers;

$\frac{\alpha}{\beta}$ is a coordinate of the proper fraction $\widetilde{X}$;

k is a number of the fractional coordinate in the numerical $\widetilde{X}$ solution;

p is a positive integer;

q is a positive integer;

δ is a positive integer;

δ_k^1 is a certain integers expressed as a percentage;

δ_k^0 is a certain integers expressed as a percentage;

$n(0)$ is a minimal number of zeros coordinates in the optimal solution;

$n(d)$ is a minimal number of non-zero coordinates in the optimal solution;

$\underline{X^*}$ is an optimal solution of certain problems;

$\underline{X^{lp}}$ is an optimal solution of certain problems;

ω_1 is a certain set of number;

ω_2 is a certain set of number;

DOI 10.15588/1607-3274-2025-3-18

$n(\omega_1)$ is a number of element of certain sets ω_1;

$n(\omega_2)$ is a number of element of certain sets ω_2;

$\underline{n}(d)$ is an integer not greater than $n(d)$;

n_1 is a lower bound on the minimal number of non-zero coordinates in the approximate solution;

$\bar{n}$ is a maximum number of non-zero coordinates in the optimal solution;

$\underline{n}(0)$ is a lower bound on the minimum number of zeros in the optimal solution;

$\tilde{f}$ is a value of the functional with respect to the solution $\tilde{X}$;

X^{it} is an innovative improved final solution;

f^{it} is a value of the objective function according to the solution X^{it};

X^t is a certain intermediate approximate solution;

f^t is a value of the function according to the solution X^t.

INTRODUCTION

Let's look at the well-known integer programming problem given below:

$$\sum_{j=1}^{n} c_j x_j \to \max, \tag{1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \left(i = \overline{1,m}\right) \tag{2}$$

$$0 \le x_j \le d_j \quad j = (\overline{1,n}). \tag{3}$$

Here $c_j > 0, a_{ij} > 0, b_i > 0$ and $d_j > 0$ $(i = \overline{1,m},\ j = \overline{1,n})$ are integers.

Note that problem (1)–(3) is called an integer linear programming problem in the literature.

In this problem, any vector $X = (x_1, x_2, ..., x_n)$ that satisfies conditions (2)–(3) is called a possible solution to the problem. The optimal solution to the problem is understood to be the solution that gives the largest (maximum) value to the function (1) among the possible solutions.

Note that among the possible solutions to problem (1)–(3), the solution that gives a large value to function (1) based on certain criteria is called an approximate (suboptimal) solution.

It is known that the problem (1)–(3) belongs to the class of "hard-to-solve problems", that is, to the class "*NP*-complete" [1]. In other words, the maximum number of operations required by any of the known methods for finding the optimal solution to this problem (e.g., branch and bounds, combinatorial, etc.) is not limited by any polynomial that depends on the size of the problem. Therefore, approximate solutions of problems (1)–(3) of various nature and speed have been developed [2–7,13–16,21–25 etc.].

First, let us briefly explain one of the methods proposed in [13]. For this purpose, let us give a certain economic interpretation to the problem (1)–(3).

Suppose a certain enterprise must produce *n* different products, expressed in number. For this, *m* number of resources $b_i, \left(i = \overline{1,m}\right)$ are allocated accordingly. Let us assume that the production of one unit of the *j-th* $j = (\overline{1,n})$ product requires the expenditure of the *i-th* $\left(i = \overline{1,m}\right)$ resource a_{ij}, $(i = \overline{1,m},\ j = \overline{1,n})$. In this case, products should be produced such that the total resources spent on their production do not exceed the given corresponding limit resources $b_i, \left(i = \overline{1,m}\right)$ and at the same time the total income from their sale is maximized.

If we denote the price of one unit of the *j-th* $j = (\overline{1,n})$ product by c_j, $(j = \overline{1,n})$ and the quantity of the product to be produced by x_j, $(j = \overline{1,n})$, then we obtain model (1)–(3).

Suppose that a certain specified product *j-th*, $j = (\overline{1,n})$ to be produced. Then, a_{ij}, $(i = \overline{1,m},\ j = \overline{1,n})$ amount of the *i*-th $\left(i = \overline{1,m}\right)$ resource must be spent on the production of one unit of this product. (If these resources are measured in monetary units, then an amount of a_{ij}, $(i = \overline{1,m},\ j = \overline{1,n})$ must be spent).

In this case, the worst-case cost per unit of product *j*-th $j = (\overline{1,n})$ is equal to $\max\limits_i a_{ij}$.

Then the price increase for each unit of product *j*-th $j = (\overline{1,n})$ mentioned above is equal to $\dfrac{c_j}{\max\limits_i a_{ij}}$ $j = (\overline{1,n})$.

Naturally, we need to produce product number j_* such that the expression $\dfrac{c_j}{\max\limits_i a_{ij}}$ $j = (\overline{1,n})$ is the largest.

Thus, we get the following selection criteria:

$$\max_j \frac{c_j}{\max\limits_i a_{ij}} = \frac{c_{j_*}}{\max\limits_i a_{ij_*}}$$

or

$$j_* = \arg\max_i \frac{c_j}{\max\limits_i a_{ij}}. \tag{4}$$

DOI 10.15588/1607-3274-2025-3-18

It should be noted that in studies [8, 10, 14], a criterion of type (4) was used for interval integer and mixed-integer programming problems of type (1)–(3).

When constructing an approximate solution to the problem (1)–(3), most methods initially assume $X=(0,0,...,0)$ as the starting value, and the unknown variable x_{j_*} with the index j_*, determined by a certain criterion, is assigned a value. After this process is carried out for all indices j, $j=(\overline{1,n})$, a certain approximate solution $X^0=\left(x_1^0,x_2^0,...,x_n^0\right)$ is obtained.

It should be noted that the approximate solution method described above is one of the known methods. However, numerous computational experiments have shown that the solution given by known approximate solution methods can differ significantly from the optimal solution. Therefore, there is a need to develop an approximate solution method that does not give a worse solution than the solutions given by known methods, works quickly, is easy to implement, and does not cause difficulties from a programming point of view. It should be noted that such a solution method is called an innovative approximate solution method, and the solution found is called an innovative approximate solution [9, 10, 26]. In this work, we have developed such an approximate solution method.

1 PROBLEM STATEMENT

Without loss of generality, let us assume that in problem (1)–(3) the coefficients are numbered as follows based on condition (4). $\frac{c_1}{\max\limits_i a_{i1}} \ge \frac{c_2}{\max\limits_i a_{i2}} \ge ... \ge \frac{c_k}{\max\limits_i a_{ik}} \ge ... \ge \frac{c_n}{\max\limits_i a_{in}}$ In this case, the initial approximate solution $X^0=\left(x_1^0,x_2^0,...,x_n^0\right)$ of problem (1)–(3) is found analytically by the following formula: for each number j, $j=(\overline{1,n})$

$$x_j^0=\begin{cases} d_j, \ if\ \forall i, i=\left(\overline{1,m}\right),\ a_{ij}d_j \le b_i-\sum\limits_{l=1}^{j-1}a_{il}x_l^0 \\ \min\limits_i\left[\left(b_i-\sum\limits_{l=1}^{j-1}a_{il}x_l^0\right)/a_{ij}\right], \ \text{otherwise}. \end{cases}$$

We can briefly write this formula as follows:

$$x_j^0=\min\left\{d_j;\ \min\limits_i\left[\left(b_i-\sum_{l=1}^{j-1}a_{il}x_l^0\right)/a_{ij}\right]\right\}. \tag{5}$$

Here, the symbol [z] denotes the integer part of the number z. It is clear that the solution found by formula (5) will have the following structure

$$X^0=\left\{x_1^0,x_2^0,...,x_{r-1}^0,0,x_{r+1}^0,...,x_n^0\right\}. \tag{6}$$

In the problem under consideration, the value of the function (1) corresponding to the approximate solution of (6) is

$$f^0=\sum_{j=1}^{n}c_jx_j^0 .$$

Note that in formula (6) we must have $x_j^0>0,\ \ x_r^0=0$ for the $j=\overline{1,r-1}$ numbers. The remaining coordinates $x_{r+1}^0,...,x_n^0$ can be 0 or positive integers.

2 REVIEW OF THE LITERATURE

Integer programming problems, as well as their individual classes, have been known since the middle of the last century. Since these problems are of great practical importance, various exact methods for their solution have been developed. However, it soon became clear that these problems belong to the "NP-complete class", that is, to the class of "hard-to-solve problems" [1]. In other words, there are no polynomial-time methods for finding optimal solutions to these problems. Therefore, methods have been developed to find various types of approximate (suboptimal) solutions to these problems. [2–5, 13, 15–17, 21, 22, 24, 25]. However, although these methods work quickly, the solution they provide may differ significantly from the optimal solution. On the other hand, more general classes of integer programming problems, namely problems with initial data in the form of intervals, have begun to be studied. [6–12, 14, 18 etc]. In addition, various models of the Boolean programming problem, as well as some integer programming problems, have been investigated in [19, 20, 23]. However, there is a need to develop new and more efficient approximate solution methods. Because better approximate solutions to real practical problems must be developed. Such innovative methods have been proposed in [9, 10, 26] for the knapsack problem and the integer knapsack problem. Here, generalization means that the given coefficients are located in certain intervals. However, a new innovative approximate solution method for the more general integer programming problem is implemented in this work. Note that the method proposed in [26] is a special case of the method presented in this paper.

3 MATERIALS AND METHODS

First, let us determine a certain number k that falls within the interval where the coordinates of the optimal solution and the approximate solution differ. For this purpose, let us construct a certain $\widetilde{X}=(\widetilde{x}_1,\widetilde{x}_2,...,\widetilde{x}_n,)$ solution of the problem as follows, by taking $0\le x_j\le d_j$, $\left(j=\overline{1,n}\right)$ instead of the condition (3). For each number j, $j=(\overline{1,n})$

DOI 10.15588/1607-3274-2025-3-18

$$\tilde{x}_j = \begin{cases} a_{ij}d_j \le b_i - \sum_{l=1}^{j-1} a_{il}\tilde{x}_l, \forall i, i = \overline{1,m}, \\ \min_i \left(b_i - \sum_{l=1}^{j-1} a_{il}\tilde{x}_l \Big/ a_{ij} \right), \exists i \ \ a_{ij}d_j > b_i - \sum_{l=1}^{j-1} a_{il}\tilde{x}_l \ (k := j), \\ 0, \ \ j = k+1,...,n \end{cases}$$

Obviously, this solution will have the following structure.

$$\tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_k, ..., \tilde{x}_n) = \left(d_1, d_2, ..., \ d_{k-1}, \frac{\alpha}{\beta}, 0, ..., 0 \right). \qquad (7)$$

Here, should be $0 \le \frac{\alpha}{\beta} < d_k$.

From experiments conducted on numerous random problems of various sizes, it becomes clear that the approximate solution $X^0 = \left(x_1^0, x_2^0, ..., x_n^0\right)$ found by formula (6) and the coordinates of the optimal solution to the problem can differ around a certain $(k-p; k+p)$, the number k in the notation (7). (The choice of numbers p and q will be discussed below.) Because, for numbers j in that neighborhood, the $\frac{c_j}{\max_i a_{ij}}$, $(j \in (k-p; k+q))$ ratios are close to each other. Therefore, when constructing a solution using formula (5), the advantage of which coordinate in that interval is selected and evaluated is of great importance. However, since these advantages do not differ significantly, new solutions can be constructed using formula (5) by assigning separate values to the unknowns in that interval. We can accept the one that gives the largest value to the function (1) among the obtained solutions as the final solution. Note that, based on the principle we have shown, a certain approximate solution method for the problem (1)–(3) was given in [7, 10]. These works differ from each other in the choice of the numbers p and q. In those works, $p=q$ was assumed and the procedure for finding them was shown. It is clear that if the number k determined by expression (7) is close to the last number n, or to the first number, then the numbers k-p and k+p may go beyond the interval [1, n]. Therefore, the methods given in works [7–10] may not work. In work [10], the numbers p and q are chosen as a certain percentage of the number k, with p=q. Naturally, this method may not work if the number k is close to the ends of the interval [1, n]. Taking all this into account, we have given a new and more universal method in this work for choosing the numbers p and q. Thus, we have estimated the corresponding minimal number of zeros and non-zero coordinates in the optimal solution of the problem (1)–(3) and denoted them by n(0) and n(d). Therefore, there is no need to change the values of the first n(d) and last n(0) coordinates in the solution of (6). Because these coordinates are the same as the coordinates of the optimal solution. Thus, it is important to find the numbers n(0) and n(d), and in this work we have proved certain theorems that allow us to find these numbers.

Thus, for each number j, $(j \in [k-p, k+q])$ we can record the possible values of the coordinate x_j and construct the remaining coordinates using formula (5). It is clear that in this case, some coordinate will coincide with the coordinate of the optimal solution.

It should be noted that the numbers p and q can be found, respectively, through the minimal number $n(d)$ of non-zero coordinates and the minimal number $n(0)$ of zeros in the optimal solution of the problem (1)–(3). In other words, must be $p = k - n(d), q = n - n(0) - k$. We will show below the difficulty of finding the numbers $n(d)$ and $n(0)$ used here. Therefore, we will also give the process of evaluating these numbers. Note that such a problem was considered in papers [8, 10, 13, 14]. For example, in paper $[13]$, the neighborhood of $[k-p, k+p]$ is considered and the number p here is found from the relation

$$p = \arg\left\{ \max_i \left(\frac{c_{k-i}}{a_{k-i}} - \frac{c_{k+i}}{a_{k+i}} \right) \le \delta \right\}.$$

The number δ used here is a positive integer and must be given in advance. If the number k found by formula (7) is close to the last number n or the first number, then specifying the interval $[k-p, k+p]$ becomes uncertain. Instead of the $[k+p, k-p]$ neighborhood of the k-th coordinate, the interval $\left[\delta_k^1, \delta_k^0\right]$ was determined in works [8, 14].

Here, the numbers δ_k^1 and δ_k^0 are found from the relation $\delta_k^1 = \left[k \cdot \frac{q}{100}\right]$, $\delta_k^0 = \left[(n-k) \cdot \frac{q}{100}\right]$, and the number q is the minimum number of units or zeros in the optimal solution. However, in this case, uncertainties may also arise. Because, when the minimum number of units or zeros is chosen for the number q, the interval $\left[\delta_k^1, \delta_k^0\right]$ may be different.

Note that in Work $[10]$, the interval integer bag problem was reduced to the corresponding known integer bag problem, and a certain neighborhood of the k-th coordinate taking a fractional value in the corresponding continuous problem was selected.

However, in this work, the interval $[k-p, k+p] = [n(d), n-n(0)]$ is adopted as the circumference of the k-th coordinate. It is clear that the condition $n(d) \le k \le n - n(0)$ will be satisfied for the k-th number. As can be seen, in order to obtain a better solution than the initial approximate solution by performing a small number of calculation operations, we

DOI 10.15588/1607-3274-2025-3-18

need to find the numbers $n(d)$ and $n(0)$ for the problem under consideration. In this case, in the process of constructing an innovative approximate solution, we need to keep the first $n(d)$ and last $n(0)$ coordinates in the solution (7) as they are.

Thus, it is necessary to solve the following problems.

$$\sum_{j=1}^{n} x_j \to \min, \tag{8}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad (i = \overline{1,m}), \tag{9}$$

$$\sum_{j=1}^{n} c_j x_j \ge f^0, \tag{10}$$

$$0 \le x_j \le d_j, \quad j = \left(\overline{1,n}\right), \tag{11}$$

$$x_j - \text{integer}, \quad \left(j = \overline{1,n}\right) \tag{12}$$

and

$$\sum_{j=1}^{n} x_j \to \max, \tag{13}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad (i = \overline{1,m}), \tag{14}$$

$$\sum_{j=1}^{n} c_j x_j \ge f^0, \tag{15}$$

$$0 \le x_j \le d_j, \quad j = \left(\overline{1,n}\right), \tag{16}$$

$$x_j - \text{integer} \quad \left(j = \overline{1,n}\right). \tag{17}$$

We can solve problem (18)–(12) and find its optimal solution $\underline{X}^* = \left(\underline{x}_1^*, \underline{x}_2^*, ..., \underline{x}_n^*\right)$. Then the set $\omega_1 = \left\{ j \mid \underline{x}_j^* > 0 \right\}$ is easily determined. Then the minimal number $n(d)$ of non-zero coordinates in the optimal solution of problem (1)–(3) is $n(d) = n(\omega_1)$. Here, $n(\omega_1)$ is the number of elements of the set ω_1.

Since the problem (8)–(12) shown above belongs to the class of hard-to-solve problems, it may not be possible to solve it in real time. To alleviate this problem, it is necessary to solve the problem (8)–(11) which has a larger domain. By solving the obtained linear programming problem, we can find its optimal solution $\underline{X}^{lp} = \left(\underline{x}_1^{lp}, \underline{x}_2^{lp}, ..., \underline{x}_n^{lp}\right)$. For the $\underline{n}(d)$ number of non-zero coordinates in this solution, the $\underline{n}(d) = n(\omega_2) \le n(d)$ relation is satisfied. Here $\omega_2 = \left\{ j \mid \underline{x}_j^{lp} > 0 \right\}$. Because, the number $\underline{n}(d)$ is taken from the problem with a larger domain.

Note that to find the number $\underline{n}(d)$, the optimal solution of the linear programming problem is used. Naturally, when solving such large-scale problems, time and memory problems may still arise. To overcome this problem, we can solve the problem

$$\sum_{j=1}^{n} x_j \to \min,$$

$$\sum_{j=1}^{n} c_j x_j \ge f^0,$$

$$0 \le x_j \le d_j, \quad j = \left(\overline{1,n}\right).$$

which has a larger possible solution region, instead of the problem (8)–(12).

In this problem, assuming that the conditions $c_1 \ge c_2 \ge ... \ge c_{n-1} \le c_n$ are satisfied, then we can find the minimal number n_1 of nonzero coordinates in its optimal solution from the relation

$$\sum_{j=1}^{n_1} c_j d_j \le f^0 \le \sum_{j=1}^{n_1+1} c_j d_j.$$

Naturally, it will be $n_1 \le \underline{n}(d) \le n(d)$.

Therefore, the obtained number n_1 can be a lower bound on the minimal number of non-zero coordinates in the optimal solution of problem (1)–(3).

Note that to find the minimal number of zeros in the optimal solution of problem (1)–(3), we must first solve problem (13)–(17) or (13)–(16), respectively, and finally solve problem

$$\sum_{j=1}^{n} x_j \to \max, \tag{18}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad (i = \overline{1,m}), \tag{19}$$

$$0 \le x_j \le d_j, \quad j = \left(\overline{1,n}\right), \tag{20}$$

$$x_j - l \quad \textit{is} \ \text{integer}. \tag{21}$$

In this case, we find the maximum number of non-zero coordinates in the optimal solution of problem (1)–(3). However, since we need to find the minimum number of zeros in the optimal solution, we need to subtract the maximum number of non-zero coordinates from the number of unknowns *n*. For this purpose, let us assume that the conditions $a_{i1} \le a_{i2} \le ... \le a_{in}$ are satisfied separately for each *i*-th inequality in the system (19).

Then, in problem (1)–(3), we can find the maximum number of nonzero coordinates $\overline{n}$ of the optimal solution based on the conditions

DOI 10.15588/1607-3274-2025-3-18

$$\sum_{j=1}^{n_i} a_{ij} d_j \le b_i \le \sum_{j=1}^{n_i+1} a_{ij} d_j,\ (i=\overline{1,m})$$

as $\bar{n} = \min_i n_i$.

Then, we can accept the number $\underline{n}(0) = n - \bar{n} \le n(0)$ as the minimal number $\underline{n}(0)$ of zeros in the optimal solution of that problem.

So the following theorem has been proven.

Theorem: If the numbers $n(d)$ and $n(0)$ denote the minimum numbers of ones and zeros, respectively, in the optimal solution of problem (1)–(3), $\underline{n}(d)$ and $\underline{n}(0)$ denotes their corresponding lower bounds, then the relations $\underline{n}(d) \le n(d)$ and $\underline{n}(0) \le n(0)$ are true.

As a result, we can conclude that the coordinates of the optimal solution to problem (1)–(3) and the coordinates of the solution to problem (7) can only take different values in the interval $[\underline{n}(d), \underline{n}(0)]$.

Note that in the solution (7) there are zeros to the right of the k-th coordinate, and d_j numbers to the left. Therefore, in order to construct new solutions using (5), we need to write $x_j = 1,2,...,d_j$ for each $j=k+1$, $k+2,...$ $\underline{n}(0)$ and $x_j = d_{j-1}, d_{j-2},...,0$ for each $j = \underline{n}(d), \underline{n}(d)+1,...,k$ number. We select the best one from the obtained solutions and call it the innovative approximate solution.

Thus, we can write the algorithm for the process of constructing the innovative approximate solution that we propose.

ALGORITHM

Step 1. Must give the numbers n, c_j, a_{ij}, b_i, d_j $(i=\overline{1,m},\ \ j=\overline{1,n})$ and assimilate $bb_i := b_i;\ (i=\overline{1,m})$.

Step 2. In problem (1)–(3), without considering the condition that the unknowns are complete, let us find its solution

$$\tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_k, ..., \tilde{x}_n) = \left(d_1, d_2, ..., d_{k-1}, \frac{\alpha}{\beta}, 0, ..., 0\right)$$

using the following well-known formula. For each number $j, (j=1,2,...,n)$

$$\tilde{x}_j = \begin{cases} for\ d_j, if\ \forall i, i=\overline{(1,m)},\ a_{ij}d_j \le b_i - \sum_{l=1}^{j-1} a_{il}\tilde{x}_l \\ \min_i \left(b_i - \sum_{i=1}^{j-1} a_{il}\tilde{x}_l \Big/ a_{ij} \right), if\ \exists i,\ a_{ij}d_j > b_i - \sum_{l=1}^{j-1} a_{il}\tilde{x}_l\ (k:=j), \\ 0,\ j=k+1,...,n. \end{cases}$$

and accept $kk := k$, $r := 0$;

Step 3. If the obtained $\tilde{x}_k$ coordinate is an integer, then the solution $\tilde{X}$ coincides with the optimal $X^* = (x_1^*, x_2^*, ..., x_n^*)$ solution of problem (1)–(3). Then, $f^* := \sum_{j=1}^{n} c_j \tilde{x}_j$, must accept $X^* = (x_1^*, x_2^*, ..., x_n^*) = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_k, ..., \tilde{x}_n)$, print and go to step 21.

Step 4. Let's find the approximate solution of the problem $X^0 = (x_1^0, x_2^0, ..., x_n^0)$ using the following formula

$$x_j^0 = \begin{cases} d_j,\ \forall i, i=\overline{(1,m)},\ a_{ij}d_j \le b_i - \sum_{l=1}^{j-1} a_{il}x_l^0 \\ \min_i \left[\left(b_i - \sum_{l=1}^{j-1} a_{il}x_l^0 \right) / a_{ij} \right], \text{otherwise} \end{cases}$$

for each number $(j = 1,2,...,n)$.

Step 5. Calculate the numbers $\tilde{f}$ and f^0 as follows.

$$\tilde{f} = \sum_{j=1}^{n} c_j \tilde{x}_j, \quad f^0 = \sum_{j=1}^{n} c_j x_j^0.$$

Accept $f^{it} := f^0$ $X^{it} = (x_1^0, x_2^0, ..., x_n^0)$ and remembered the solution $X^{it} = (x_1^0, x_2^0, ..., x_n^0)$ with f^{it}.

Step 6. To find the minimum number n_1 of non-zero coordinates, let's convert the $c_j, (j=\overline{1,n})$ numbers to $c_1 \ge c_2 \ge ... \ge c_n$ form and use the following relation:

$$\sum_{j=1}^{n_1} c_j d_j \le f^0 \le \sum_{j=1}^{n_1+1} c_j d_j.$$

Step 7. To find the $n_i, (i=\overline{1,m})$ numbers, we need to arrange the coefficients of $a_{ij}, (i=\overline{1,m},\ j=\overline{1,n})$ for each number $i, (i=\overline{1.m})$, separately in increasing order, like $a_{i1} \le a_{i2} \le ... \le a_{in}$. Then, we need to use the following relations:

$$\sum_{j=1}^{n_i} a_{ij} d_j \le b_i \le \sum_{j=1}^{n_i+1} a_{ij} d_j,\ (i=\overline{1,m}).$$

Finally, should be write and remember $\bar{n} = \min_i n_i$ and $\underline{n}(0) = n - \bar{n}$.

Step 8. Should be accept the following prices

$$x_k^t := [\tilde{x}_k];\ b_i := bb_i - a_{ik}x_k^t,\ (i=\overline{1.m}).$$

DOI 10.15588/1607-3274-2025-3-18

Step 9. For each numbers j, $j=1,2,...,n$ and $j \neq k$, let's find the approximate solution of the interval $X^t = \left(x_1^t, x_2^t, ..., x_n^t\right)$ using the following formula.

$$x_j^t = \begin{cases} for\ d_j,\ if\ \forall\ i,\ i=\left(\overline{1,m}\right)\quad a_{ij}d_j \leq b_i - \sum_{l=1}^{j-1} a_{il}x_l^t \\ \min_i \left[\left(b_i - \sum_{l=1}^{j-1} a_{il}x_l^t\right)/a_{ij}\right],\quad else. \end{cases}$$

Step 10. Let's calculate the number f^t

$$f^t := \sum_{j=1}^{n} c_j x_j^t.$$

If $f^t > f^{it}$, then should be accept $f^{it} := f^t$, $X^{it} = (x_1^t, x_2^t, ..., x_n^t)$.

Step 11. If r=0, then should be accept $x_k^t := \lfloor x_k^t \rfloor + 1$; $b_i := bb_i - a_{ik}x_k^y$, $(i=\overline{1,m})$; $r:=1$ and go to step 9.

Step 12. Should be accept $r:=0$

Step 13. To find the intermediate solution $X^t = \left(x_1^t, x_2^t, ..., x_n^t\right)$, let's use the following rule:

For each j, $j=(1,2,...,n;\quad j \neq k)$

$$x_j^t = \begin{cases} for\ d_j,\ if\ \forall\ i,\ i=\left(\overline{1,m}\right),\ a_{ij}d_j \leq b_i - \sum_{l=1}^{j-1} a_{il}x_l^t \\ \min_i \left[\left(b_i - \sum_{l=1}^{j-1} a_{il}x_l^t\right)/a_{ij}\right],\quad else. \end{cases}$$

Then, the corresponding value of the functional is

$$f^t := \sum_{j=1}^{n} c_j x_j^t.$$

Step 14. If $f^t > f^{it}$, $f^{it} := f^t$, $X^{it} = (x_1^t, x_2^t, ..., x_n^t)$ should be written and memorized. If $r:=1$ skip to Step 18.

Step 15. If $x_k^t < d_k$, then skip to Step 17.

Step 16. Should accept $k := k+1$; If $k > n(0)$, then accept $k := kk$ and skip to the Step 18.

Step 17. $x_k^t := 1,2,...,d_k$ and accordingly by taking $b_i := bb_i - a_{ik}x_k^t$ $(i=\overline{1.m})$ skip to the Step 13.

Step 18. $r:=1$; If $x_k^t = d_k$, then $k := k-1$; If $k < n(1)$ skip to the Step 20.

Step 19. $x_k^t := 1,2,...,d_k$ k values corresponding to $b_i := bb_i - a_{ik}x_k^t$ $(i=\overline{1.m})$ and go to Step 13.

Step 20. Should be print f^{it}, $X^{it} = (x_1^{it}, x_2^{it}, ..., x_n^{it})$ $\delta = \left(\left[\tilde{f}\right] - f^t\right)/\left[\tilde{f}\right]$ and $\delta^i = \left(\left[\tilde{f}\right] - f^{it}\right)/\left[\tilde{f}\right]$.

Step 21. STOP.

4 EXPERIMENTS

Let's solve a numerical example using the algorithm we wrote above. Here, we assume that the following conditions are met.

$$\frac{c_1}{\max_i a_{i1}} \geq \frac{c_2}{\max_i a_{i2}} \geq ... \geq \frac{c_k}{\max_i a_{ik}} \geq ... \geq \frac{c_n}{\max_i a_{in}},$$

$$9x_1 + 10x_2 + 8x_3 + 6x_4 + 7x_5 \to \max,$$

$$3x_1 + 3x_2 + 1x_3 + 4x_4 + 2x_5 \leq 18,$$

$$1x_1 + 5x_2 + 4x_3 + 2x_4 + 3x_5 \leq 17,$$

$$4x_1 + 2x_2 + 3x_3 + 2x_4 + 5x_5 \leq 20,$$

$$0 \leq x_j \leq d_j,\quad \text{are integers},\quad (j=\overline{1,5}).$$

Here $d_1 = 2, d_2 = 2, d_3 = 3, d_4 = 4, d_5 = 3$. İn order words we accepted $0 \leq x_1 \leq 2,\ 0 \leq x_2 \leq 2,\ 0 \leq x_3 \leq 3,\ 0 \leq x_4 \leq 4,\ 0 \leq x_5 \leq 3$.
In this problem, let's first construct the initial $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_k, ..., \tilde{x}_n) = \left(d_1, d_2, ..., d_{k-1}, \frac{\alpha}{\beta}, 0, ..., 0\right)$ solution using the above algorithm, without considering the completeness condition on the unknowns. Then we can get $\tilde{X} = (2,2,5/4,0,0)$ and $\tilde{f} = 48$. From here we get $\tilde{x}_k = \tilde{x}_3 = 5/4$ and $k=3$. Then we need to take $kk := k$, $r:=0$ according to the algorithm.

Now let's construct an initial approximate solution $X^0 = \left(x_1^0, x_2^0, ..., x_n^0\right)$ to this problem. Then we can get $X^0 = (2,2,1,0,0)$ and $f^0 = 46$. Then, let's write the results in the following tables.

5 RESULTS

Since $k=3$, let's write the approximate solutions and f^t values found sequentially, noting $x_3^t = 3,\ x_3^t = 2,\ x_3^t = 1,\ x_3^t = 0$, in Table 1.

Table 1 – Evaluating the coordinate x_3^t

X^t	x_1^t	x_2^t	x_3^t	x_4^t	x_5^t	f^t
X_1^t	2	0	**3**	1	0	f_2^t =48
X_2^t	2	1	**2**	0	0	f_3^t =44
X_3^t	2	2	**1**	0	0	f_4^t =46
X_4^t	2	2	**0**	1	1	f_5^t =51

Now, let's write the successive approximate solutions and f^t values in Table 2 below, assuming $k = k-1 = 2$ and writing $x_2^t = 2,\ x_2^t = 1,\ x_2^t = 0$ accordingly.

DOI 10.15588/1607-3274-2025-3-18

Table 2 – Evaluating the coordinate x_2^t

X^t	x_1^t	x_2^t	x_3^t	x_4^t	x_5^t	f^t
X_5^t	2	**2**	1	0	0	f_6^t =44
X_6^t	2	**1**	2	1	0	f_7^t =50
X_7^t	2	**0**	2	2	0	f_8^t =46

Let us write the approximate solutions and f^t values obtained by giving successive values of $x_1^t = 2,\ x_1^t = 1,\ x_1^t = 0$ by assuming $k = 1$ in Table 3.

Table 3 – Evaluating the coordinate x_1^t

X^t	x_1^t	x_2^t	x_3^t	x_4^t	x_5^t	f^t
X_8^t	**2**	2	1	0	0	f_9^t =46
X_9^t	**1**	2	1	1	0	f_{10}^t =43
X_{10}^t	**0**	2	1	1	0	f_{11}^t =34

Then, let's continue the solution process for the unknowns x_4 and x_5 located to the right of the *k*-th coordinate. First, let's find the appropriate approximate solutions and f^t values by noting $x_4^t = 4,\ x_4^t = 3,\ x_4^t = 2,\ x_4^t = 1,\ x_4^t = 0$ and write them in Table 4.

Table 4 – Evaluating the coordinate x_4^t

X^t	x_1^t	x_2^t	x_3^t	x_4^t	x_5^t	f^t
X_{11}^t	0	0	2	**4**	0	f_{11}^t =40
X_{12}^t	2	0	0	**3**	0	f_{12}^t =36
X_{13}^t	2	1	1	**2**	0	f_{13}^t =48
X_{14}^t	2	2	0	**1**	1	f_{14}^t =51
X_{15}^t	2	2	1	**0**	0	f_{15}^t =46

Finally, let's find the appropriate approximate solutions and f^t values by noting $x_5^t = 3,\ x_5^t = 2,\ x_5^t = 1$ and $x_5^t = 0$ and write them in Table 5.

Table 5 – Evaluating the coordinate x_5^t

X^t	x_1^t	x_2^t	x_3^t	x_4^t	x_5^t	f^t
X_{16}^t	1	0	0	0	**3**	f_{16}^t =30
X_{17}^t	2	1	0	0	**2**	f_{17}^t =32
X_{18}^t	2	2	0	1	**1**	f_{18}^t =51
X_{19}^t	2	2	1	0	**0**	f_{19}^t =46

Note that initially, the approximate solution found by the known method was obtained as $X^0 = (2,2,1,0,0)$ and $f^0 = 46$. However, as can be seen from the tables above, the value of $f^0 = 46$ increased to 48, 50 and 51.

Therefore, by choosing the one corresponding to the largest of these values, the innovative approximate solution will be $X^{it} = \left(x_1^{it}, x_2^{it}, x_3^{it}, x_4^{it}, x_5^{it}\right) = (2,2,0,1,1)$ and the innovative approximate value $f^{it} = 51$.

Note that this solution is also the optimal solution to the problem under consideration.

6 DISCUSSION

The main essence and novelty of the method proposed in this article is that a solution provided by any of the known methods is initially accepted as a starting point. Then, this solution is gradually improved**.** For this purpose, the numbers of coordinates at which the approximate solution accepted as the initial solution and the optimal solution can differ are determined. Finding these numbers is based on proven criteria. It is clear that if we know which of the coordinates of the approximate solution are not the same as the coordinates of the optimal solution and we give these coordinates one by one values in their variation interval, at a certain step those coordinates will coincide with the coordinates of the optimal solution. So a better solution can be obtained than the initial approximate solution. Note that we can find the optimal solution by giving values to all of these coordinates at the same time. However, in this case we will have to look at an exponential amount of coordinates. This would require unrealistic computer time. Therefore, we will get a new, better solution by changing these coordinates one by one. Therefore, the solution obtained through the application of this method will be better than the one provided by known approximate methods. The principle of improving the initial solution is rigorously mathematically justified through proven theorems. A specific problem was solved to clearly demonstrate the sequence of implementation of the proposed method, as well as to determine the quality of that method.

CONCLUSIONS

In the article, a new approximate solution method for the integer programming problem has been developed. In all known approximate solution methods, the number of the unknown is found based on certain criteria and the unknown is evaluated. After that, the unknown is removed from the list. As a result, a certain, approximate solution is obtained. In most cases, this solution differs significantly from the optimal solution. Therefore, it is necessary to accept any approximate solution as an initial solution and improve it further.

The scientific novelty of this article is that the numbers of coordinates at which the initially taken approximate solution may differ from the optimal solution are determined using proven theorems. Then, new solutions are constructed by assigning values to these coordinates in their variation interval.

DOI 10.15588/1607-3274-2025-3-18

The best of these solutions is accepted as an innovative approximate solution. From this it is immediately clear that the proposed method should be more effective. The mathematical model considered in the article arises in the problems of optimizing the production of products manufactured by number. Therefore, the method proposed in the article can be effectively applied to solving real practical problems of this type. This shows the practical value of the article. It is clear that if the initial approximate solution differs little from the optimal solution, then the method proposed in this article will provide few improvements. Therefore, in the future, it is planned to develop a new approximate solution method so that the absolute or relative error of the solution it provides is not greater than the optimal solution. It should be noted that in order to clarify the essence of the proposed method in the article, a specific issue was solved.In that problem, the initial value of the functional found by the known method was 46. In the subsequent steps, this value was 48, 50 and 51. More precisely, the initial solution was improved 3 times.

ACKNOWLEDGEMENTS

This article, i.e. “An Innovative approximate solution method for the integer programming problem” was carried out at the Institute of Control Systems of the Ministry of Science and Education of the Republic of Azerbaijan at the expense of the state budget (State Registration No. 0101 Az 00736).

REFERENCES

1. Garey M. R., Jhonson D. S. Computers and Intractability : a Guide to the Theory of NP-Completeness. San Francisco, Freeman, 1979, P. 314.
2. Martello S., Toth P. Knapsack problems: Algorithm and Computers İmplementations. New York, John Wiley & Sons, 1990, P. 296.
3. Erlebach T., Kellerer H., Pferschy U. Approximating multi-objective knapsack problems, *Management Science*, 2002, № 48, pp. 1603–1612. DOI: 10.1287/mnsc.48.12.1603.445
4. Kellerer H., Pferschy U., Pisinger D. Knapsack problems. Berlin, Heidelberg, New-york, Springer-Verlag, 2004, P.546 DOI:10.1007/978-3-540-24777-7
5. Vazirani V. V. Approximation algorithms. Berlin, Springer, 2001, P. 378. DOI:10.1057/palgrave.jors.2601377
6. Libura M. Integer programming problems with inexact objective function, *Control Cyber*, 1980, Vol. 9, No. 4. pp. 189–202.
7. Bukhtoyarov S. E., Emelichev V. A.Stability aspects of Multicriteria integer linear programming problem, Journal of Applied and Industrial mathematics, 2019, Vol. (13), №1, pp. 1–10. DOI:10.33048/daio.2019.26.624
8. Mamedov K. Sh., Mammadli N. O. Two methods for construction of suboptimistic and subpessimistic solutions of the interval problem of mixed-Boolean programming, *Radio Electronics, Computer Science, Control,* 2018, № 3 (46), pp. 57–67.
9. Niyazova R. R., Huseynov S. Y. An İnnovative İmproved Approximate Method for the knapsack Problem with coefficients Given in the Interval Form, *8-th International Conference on Control and Optimization with Industrial Applications*, Baku, 24–26 August, 2022, Vol. II, pp. 210–212.
10. Mammadov K. Sh., Niyazova R. R., Huseynov S. Y. Innovative approximate method for solving Knapsack problems with interval coefficients, *International Independent scientific journal,* 2022, №44, pp. 8–12. doi.org/10.5281/zenodo.7311206. DOI: 10.15588/1607-3274-2018-3-7.
11. Emelichev V. A., Podkopaev D. Quantitative stability analysis for vector problems of 0–1 programming, *Discrete Optimization*, 2010, Vol. 7, pp. 48–63. DOI: 10.1016/j.disopt.2010.02.001.
12. Li W., Liu X., Li H. Generalized solutions to interval linear programmers and related necessary and sufficient optimality conditions, *Optimization Methods Software*, 2015, Vol. 30, №3, pp. 516–530. DOI: 10.1080/10556788.2014.940948
13. Mamedov K. Sh., Huseinov S. Y. Method of Constructing Suboptimal Solutions of Integer Programming Problems and Successive Improvement of these Solutions, *Automatic Control and Computer Science,* 2007, Vol. 41, № 6, pp. 20–31. DOI: 10.3103/S014641160706003X
14. Mamedov K. Sh., Mamedova A. H. Ponyatie suboptimisticheskoqo i subpestimisticheskoqo resheniy i postroeniya ix v intervalnoy zadache Bulevoqo programmirovania, *Radio Electronics Computer Science Control*, 2016, No. 3, pp. 99–108. DOİ: 10.15588/1607-3274-2016-3-13
15. Hifi M., Sadfi S., Sbihi A.. An efficient algorithm for the knapsack sharing problem, *Computational Optimization and Applications*, 2002, № 23, pp. 27–45. DOI:10.1023/A:1019920507008
16. Hifi M., Sadfi S. The knapsack sharing problem: An exact algorithm, *Journal of Combinatorial Optimization,* 2002, № 6, pp. 35–54. DOI:10.1023/A:1013385216761
17. Hladik M. On strong optimality of interval linear programming, *Optimization Letters*, 2017, No. 11 (7), pp. 1459–1468. DOI:10.1007/s11590-016-1088-3
18. Devyaterikova M. V., Kolokolov A. A., Kolosov A. P. L-class enumeration algorithms for one discrete production planning problem with interval input data, *Computers and Operations Research*, 2009, Vol. 36, №2, pp. 316–324. DOI:10.1016/j.cor.2007.10.005
19. Babayev D. A., Mardanov S. S. Reducing the number of variables in integer and linear programming problems, *Computational Optimization and Applications*, 1994, № 3, pp. 99–109. DOI: https://doi.org/10.1007/BF01300969.
20. Basso A., Viscolani B. Linear programming selection of internal financial laws and a knapsack problem, *Calcolo,* 2000, Vol. 37, № 1, pp. 47–57. DOI:10.1007/s100920050003
21. Bertisimas D., Demir R. An approximate dynamic programming approach to multidimensional knapsack problems, *Management Science,* 2002, № 48, pp. 550–565. DOI:10.1287/mnsc.48.4.550.208

DOI 10.15588/1607-3274-2025-3-18

22. Billionnet A. Approximation algorithms for fractional knapsack problems, *Operation Research Letters*, 2002, № 30, pp. 336–342. DOI:10.1016/S0167-6377(02)00157-8
23. Broughan K., Zhu Nan An integer programming problem with a linear programming solution, *Journal American Mathematical Monthly,* 2000, Vol. 107, № 5, pp. 444–446. DOI:10.1080/00029890.2000.12005218
24. Calvin J. M., Leung Y. T. Average-case analysis of a greedy algorithm for the 0–1 knapsack problem, *Operation Research Letters,* 2003, № 31, pp. 202–210. DOI:10.1016/S0167-6377(02)00222-5
25. Mamedov K. Sh., Musaeva T. M. Metodi postroeniya priblijennix resheniy mnoqomernoy zadachi o rance i naxojdenie verxney ocenki optimuma, *Avtomatika i Vichislitelnaya Texnika*, 2004, № 5, pp. 72–82.
26. Mamedov K. Sh., Niyazova R. R. Innovative Improved Approximate Solution Method For the Integer Knapsack Problem, Error Compression and Computational Experiments, *Radio Electronics Computer Science Control,* 2024, № 4, pp. 64–74. DOI: 10.15588/1607-3274-2024-4-6

Received 20.05.2025.
Accepted 11.07.2025.

УДК 519.852.6

ІННОВАЦІЙНИЙ МЕТОД НАБЛИЖЕНОГО РОЗВ’ЯЗАННЯ ЗАДАЧІ ЦІЛОЧИСЛОВОГО ПРОГРАМУВАННЯ

Мамедов К. Ш. – д-р фіз.-мат. наук, професор Бакинського державного університету та завідувач відділу Інституту систем управління Міністерства науки і освіти.
Ніязова Р. Р. – докторант, науковий співробітник Інституту систем управління Міністерства освіти і науки.

АНОТАЦІЯ

Актуальність. Існують певні методи знаходження оптимального розв’язку задач цілочисельного програмування. Однак ці методи не можуть вирішувати масштабні задачі в режимі реального часу. Тому було запропоновано наближені розв’язки цих задач, які працюють швидко. Слід зазначити, що розв’язки, отримані цими методами, часто суттєво відрізняються від оптимального розв’язку. Тому виникає проблема прийняття будь-якого відомого наближеного розв’язку як початкового розв’язку та його подальшого вдосконалення.

Мета роботи Спочатку знаходиться певний наближений розв’язок. Потім, на основі доведених теорем, визначаються координати цього розв’язку, які не збігаються з оптимальним. Після цього, послідовно змінюючи ці координати, знаходять нові розв’язки. За остаточний розв’язок приймається той, який дає найбільше значення функціоналу серед цих розв’язків.

Метод. Метод, який ми пропонуємо в цій роботі, реалізується наступним чином:

Спочатку встановлюється певний наближений розв’язок задачі, потім визначаються номери координат цього розв’язку, які не збігаються з оптимальним розв’язком. Після цього встановлюються нові розв’язки шляхом послідовного присвоєння значень цим координатам по одному в їхніх інтервалах. Найкраще з розв’язків, знайдених у цьому процесі, приймається як остаточне інноваційне рішення.

Результати. Було вирішено задачу з метою візуальної ілюстрації якості та ефективності запропонованого методу.

Висновки. Метод, який ми пропонуємо в цій статті, не може дати гірших результатів, ніж будь-який метод наближеного рішення, простий з алгоритмічної точки зору, є новим, його можна легко програмувати та важливий для вирішення реальних практичних завдань.

КЛЮЧОВІ СЛОВА: задача цілочисельного програмування, вихідний наближений розв’язок, інтервал, в якому координати наближеного розв’язку можуть відрізнятися від оптимального розв’язку, інноваційний наближений-розв’язок, обчислювальні експерименти.

ЛІТЕРАТУРА

1. Garey M. R. Computers and Intractability : a Guide to the Theory of NP-Completeness / M. R. Garey, D. S. Jhonson. – San Francisco, Freeman, 1979. – P. 314.
2. Martello S. Knapsack problems: Algorithm and Computers İmplementations. / S. Martello, P. Toth. – New York : John Wiley & Sons, 1990. – P. 296.
3. Erlebach T. Approximating multi-objective knapsack problems / T. Erlebach, H. Kellerer, U. Pferschy // Management Science. – 2002. – № 48. – P. 1603–1612. DOI: 10.1287/mnsc.48.12.1603.445
4. Kellerer H. Knapsack problems / H. Kellerer, U. Pferschy, D. Pisinger. – Berlin, Heidelberg, New-york : Springer-Verlag, 2004. – P. 546 DOI:10.1007/978-3-540-24777-7
5. Vazirani V. V. Approximation algorithms / V. V. Vazirani. – Berlin : Springer, 2001. – P. 378. DOI:10.1057/palgrave.jors.2601377
6. Libura M. Integer programming problems with inexact objective function / M. Libura // Control Cyber. – 1980. –Vol. 9, No. 4. – P. 189–202.
7. Bukhtoyarov S. E. Stability aspects of Multicriteria integer linear programming problem / S. E. Bukhtoyarov, V. A. Emelichev // Journal of Applied and Industrial mathematics. – 2019. – Vol. (13), № 1. – P. 1–10. DOI:10.33048/daio.2019.26.624

DOI 10.15588/1607-3274-2025-3-18

8. Mamedov K. Sh. Two methods for construction of suboptimistic and subpessimistic solutions of the interval problem of mixed-Boolean programming / K. Sh. Mamedov, N. O. Mammadli // Radio Electronics, Computer Science, Control. – 2018. – № 3 (46). – P. 57–67.
9. Niyazova R. R. An İnnovative İmproved Approximate Method for the knapsack Problem with coefficients Given in the Interval Form / R. R. Niyazova, S. Y. Huseynov // 8-th International Conference on Control and Optimization with Industrial Applications, Baku : 24 – 26 August, 2022. Vol II. – P. 210–212.
10. Mammadov K. Sh. Innovative approximate method for solving Knapsack problems with interval coefficients. / K. Sh. Mammadov, R. R. Niyazova, S. Y. Huseynov // International Independent scientific journal. – 2022. – № 44. – P. 8–12. doi.org/10.5281/zenodo.7311206. DOI: 10.15588/1607-3274-2018-3-7.
11. Emelichev V. A. Quantitative stability analysis for vector problems of 0–1 programming // V. A. Emelichev, D. Podkopaev // Discrete Optimization. –2010. – Vol. 7. – P. 48–63. DOİ: 10.1016/j.disopt.2010.02.001.
12. Li W. Generalized solutions to interval linear programmers and related necessary and sufficient optimality conditions / W. Li, X. Liu, H. Li // Optimization Methods Software. –2015. – Vol. 30, № 3. – P. 516–530. DOI: 10.1080/10556788.2014.940948
13. Mamedov K. Sh. Method of Constructing Suboptimal Solutions of Integer Programming Problems and Successive Improvement of these Solutions / K. Sh. Mamedov, S. Y. Huseinov // Automatic Control and Computer Science. – 2007. – Vol. 41, № 6. – P. 20–31. DOI: 10.3103/S014641160706003X
14. Mamedov K. Sh. Ponyatie suboptimisticheskoqo i subpestimisticheskoqo resheniy i postroeniya ix v intervalnoy zadache Bulevoqo programmirovania / K. Sh. Mamedov, A. H. Mamedova // Radio Electronics, Computer Science, Control. – 2016. – No. 3. – P. 99–108. DOİ: 10.15588/1607-3274-2016-3-13
15. Hifi M. An efficient algorithm for the knapsack sharing problem / M. Hifi, S. Sadfi, A. Sbihi // Computational Optimization and Applications. – 2002. – № 23. – P. 27–45. DOI:10.1023/A:1019920507008
16. Hifi M. The knapsack sharing problem: An exact algorithm / M. Hifi, S. Sadfi // Journal of Combinatorial Optimization. – 2002. – № 6. – P. 35–54. DOI:10.1023/A:1013385216761
17. Hladik M. On strong optimality of interval linear programming / M. Hladik // Optimization Letters. – 2017. – No. 11(7). – P. 1459–1468. DOI:10.1007/s11590-016-1088-3
18. Devyaterikova M. V. L-class enumeration algorithms for one discrete production planning problem with interval input data / M. V. Devyaterikova, A. A. Kolokolov, A. P. Kolosov // Computers and Operations Research. – 2009. – Vol. 36, № 2. – P. 316–324. DOI:10.1016/j.cor.2007.10.005
19. Babayev D. A. Reducing the number of variables in integer and linear programming problems / D. A. Babayev, S. S. Mardanov // Computational Optimization and Applications. – 1994. – № 3. – P. 99–109. DOI: https://doi.org/10.1007/BF01300969.
20. Basso A. Linear programming selection of internal financial laws and a knapsack problem / A. Basso, B. Viscolani // Calcolo. – 2000. – Vol. 37, № 1. – P. 47–57. DOI:10.1007/s100920050003
21. Bertisimas D. An approximate dynamic programming approach to multidimensional knapsack problems / D. Bertisimas, R. Demir // Management Science. – 2002. – № 48. – P. 550–565. DOI:10.1287/mnsc.48.4.550.208
22. Billionnet A. Approximation algorithms for fractional knapsack problems / A. Billionnet // Operation Research Letters. – 2002. – № 30. – P. 336–342. DOI:10.1016/S0167-6377(02)00157-8
23. Broughan K. An integer programming problem with a linear programming solution./ K. Broughan, Nan Zhu // Journal American Mathematical Monthly. – 2000. – Vol. 107, № 5. – P. 444–446. DOI:10.1080/00029890.2000.12005218
24. Calvin J. M. Average-case analysis of a greedy algorithm for the 0–1 knapsack problem / J. M. Calvin, Y. T. Leung // Operation Research Letters. – 2003. – № 31. – P. 202–210. DOI:10.1016/S0167-6377(02)00222-5
25. Mamedov K. Sh. Metodi postroeniya priblijennix resheniy mnoqomernoy zadachi o rance i naxojdenie verxney ocenki optimuma. / K. Sh. Mamedov, T. M. Musaeva // Avtomatika i Vichislitelnaya Texnika. –2004. – № 5. – P. 72–82.
26. Mamedov K. Sh. Innovative Improved Approximate Solution Method For the Integer Knapsack Problem, Error Compression and Computational Experiments / K. Sh. Mamedov, R. R. Niyazova // Radio Electronics Computer Science Control. – 2024. – № 4. – P. 64–74. DOI: 10.15588/1607-3274-2024-4-6.

DOI 10.15588/1607-3274-2025-3-18