Informatics and Control Problems 46 Issue 1 (2026)
journal homepage: www.icp.az

Experimental analysis of an innovative approximate solution method for an integer programming problem

Rayiha Niyazova

Institute of Mathematics, Baku, Azerbaijan

ARTICLE INFO

Article history:
Received 05.09.2025
Received in revised form 17.09.2025
Accepted 25.09.2025
Available online 20.03.2026

Keywords:
Integer programming problem
Innovative approximate solution
Experimental analysis

ABSTRACT

The article presents an experimental analysis of an innovative approximate solution method for the integer programming problem, of which the author is a co-author. This method was published in a Web of Science-indexed journal. In that work, a theoretical investigation of the method was provided, and for clarity, the method was explained using a numerical example. However, no experiments were conducted there to determine the effectiveness of the method. Naturally, a more comprehensive conclusion regarding the effectiveness and quality of the method can be drawn after extensive experiments conducted on problems of various dimensions with random coefficients. Therefore, in this work, numerous problems of different dimensions and with random coefficients are solved by applying said method, and conclusions regarding the quality of the method are drawn.

1. Introduction

The article investigates an integer programming problem formulated as follows:

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

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

$$0 \le x_j \le d_j \qquad \text{and are integers,} \qquad (j = \overline{1,n}). \tag{3}$$

Here, it is assumed that $c_j > 0$, $a_{ij} \ge 0$, $b_i > 0$ and $d_j > 0$, $(i = \overline{1,m}\,,\ j = \overline{1,n})$ are given numbers. Without loss of generality, we can assume these to be integers.

It should be noted that problem (1)-(3) is well-known in the literature, and methods such as “branch and bound” and other combinatorial-type methods have been developed to find its optimal solution [1-4], etc. However, it soon became clear that only problems with a small number of variables could be solved using these methods. This is because the maximum number of operations required by these methods is of exponential order, depending on the dimension of the problem. Such problems belong to the class of NP-complete, i.e., computationally difficult problems [5]. Therefore, various methods have been developed to find approximate (suboptimal) solutions for problem (1)-(3) and its

E-mail address: rayihaniyazova@gmail.com (R.R. Niyazova)

www.icp.az/2026/1-08.pdf https://doi.org/10.54381/icp.2026.1.08

special cases [6-14], etc.

It should be noted that most known approximate (suboptimal) solution methods are based on the principle of assigning a value to a single variable selected according to a specific criterion. As a result, the constructed solution may not coincide with the optimal solution. Therefore, it would be more accurate to call this solution a suboptimal solution rather than an approximate solution [7].

Numerous experiments conducted by various authors show that the discovered approximate (suboptimal) solutions in most cases differ significantly from the optimal solution. That is, the absolute and relative errors of the values these solutions provide to the objective function can be large compared to the optimal value. Therefore, it becomes necessary to develop new solution methods through which an initial approximate solution is found and subsequently improved as innovatively as possible. The problem of constructing an improved solution based on this idea was first addressed in [15]. Certain deficiencies in that method were pointed out in [16, 17], and new innovative solution methods were developed. In work [18], a method for constructing an initial solution and further improving it innovatively was provided for the more general class of integer programming problems. However, the quality of the method proposed in [18], in which the author of this article also participated, was not investigated through experimental analysis. As we see, it is necessary to draw a conclusion about the effectiveness of that method by conducting experiments on various large-dimension problems. To this end, the method was programmed, and extensive computational experiments were conducted.

2. Problem statement

Note that we will present some of the basic principles from [18] here without providing their proofs.

First, we assume that the indices of the variables in problem (1)–(3) correspond to the following principle:

$$\frac{c_1}{\max\limits_i a_{i1}} \geq \frac{c_2}{\max\limits_i a_{i2}} \geq \ldots \geq \frac{c_k}{\max\limits_i a_{ik}} \geq \ldots \geq \frac{c_n}{\max\limits_i a_{in}}$$

It should be noted that such numbering can always be achieved. Then, the coordinates of the initial solution

$$X^0 = (x_1^0, x_2^0, \ldots, x_n^0) \tag{4}$$

are determined as follows.

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

In this case, the initial approximate value f^0 of function (1) is as follows:

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

It should be noted that in the constructed initial solution $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$, an index k is selected according to a specific rule to identify the coordinates that differ from the optimal solution, and by modifying the coordinates within a certain neighborhood of this index, new solutions are constructed. Here, the index k corresponds to the first coordinate in the solution $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ that is not equal to d_k.

In addition, it is not necessary to change the first $n(d)$ non-zero coordinates and the last $n(0)$ coordinates equal to zero in the solution X^0. It should be noted that in work [18], certain theorems were proven to find the numbers $\underline{n}(0)$ and $\underline{n}(d)$. The determination of these numbers is shown in the algorithm below.

Thus, the essence of the method proposed in [18] briefly consists in the following.

It is clear that in solution (4), the d_j numbers are located to the left of the k-th coordinate, while to the right, there are either numbers smaller than d_j or zeros.

Consequently, new solutions are constructed using formula (5) by iteratively assigning values $x_j = 1,2,\dots,d_j$ for each fixed index j=k+1, k+2,...$\underline{n}(0)$, and $x_j = d_{j-1}, d_{j-2}, \dots, 0$ for indices j=$\underline{n}(d)$, $\underline{n}(d)+1$,...,k. From this set of generated solutions, the one that maximizes the objective function (1) is selected and defined as the innovative approximate solution.

Let us write the algorithm for constructing this solution in a slightly modified form based on [18], as we will develop a program and conduct experiments using this specific algorithm.

Algorithm

Step 1. Input the numbers m, n and coefficients c_j, a_{ij}, b_i, d_j $(i=\overline{1,m}\,,\ j=\overline{1,n})$ and assign

$$bb_i := b_i;\,(i=\overline{1,m}).$$

Step 2. For problem (1)-(3) determine the solution

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

For each index j, $(j=1,2,\dots,n)$

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

and set kk:=k, r:=0.

Step 3. If the obtained coordinate $\tilde{x}_k$ is an integer, then the solution $\tilde{X}$ coincides with the optimal solution $X^* = (x_1^*,\ x_2^*, \dots, x_n^*)$ of problem (1)–(3). In this case, set

$$f^* := \sum_{j=1}^{n} c_j\,\tilde{x}_j,$$

$X^* = (x_1^*,\ x_2^*, \dots, x_n^*) = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_k, \dots, \tilde{x}_n)$ print these results and proceed to Step 21.

Step 4. Construct the initial approximate solution $X^0 = (x_1^0, x_2^0, \dots, x_n^0)$ for problem (1)–(3) as follows: for each index $(j=1,2,\dots,n)$

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

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

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

Set $f^{it} := f^0$, and store f^{it} and $X^{it} = (x_1^0,\ x_2^0, \dots, x_n^0)$ as the innovative value and solution.

<u>Step 6</u>. To find the minimum number $\underline{n}(d)$ of non-zero coordinates, arrange the values $c_j, (j = \overline{1,n})$ as $c_1 \geq c_2 \geq \cdots \geq c_n$, and use the following relationship:

$$\sum_{j=1}^{\underline{n}(d)} c_j d_j \leq f^0 \leq \sum_{j=1}^{\underline{n}(d)+1} c_j\, d_j$$

<u>Step 7</u>. For each index $i, (i = \overline{1,m})$ arrange the coefficients a_{ij}, $(i = \overline{1,m}\,,\ j = \overline{1,n})$ individually as $a_{i1} \ \leq a_{i2} \leq \cdots \leq a_{in}$, and determine the numbers $n_i \ (i = \overline{1,m})$:

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

Finally, set $\overline{n} = \min_i n_i$ and $\underline{n}(0) = n - \bar{n}$.

<u>Step 8.</u> Set $x_k^t \coloneqq \lceil \tilde{x}_k \rceil;\ \ b_i \coloneqq bb_i - a_{ik} x_k^t,\ (i = \overline{1,m})$:

<u>Step 9</u>. For each index $j = 1,2, \dots, n$ and $j \neq k$, using the formula

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

determine the intermediate approximate solution $X^t = (x_1^t,\ x_2^t, \dots, x_n^t)$.

<u>Step 10</u>. Calculate f^t:

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

And if $f^t > f^{it}$, set $f^{it} \coloneqq f^t,\ \ X^{it} = (x_1^t,\ x_2^t, \dots, x_n^t)$, and store these values.

<u>Step 11</u>. If $r = 0$, set $x_k^t \coloneqq \lceil x_k^t \rceil + 1$; $b_i \coloneqq bb_i - a_{ik} x_k^t \ \ ,\ \ \ (i = \overline{1,m})$, then write $r\!: = 1$ and proceed to Step 9.

<u>Step 12</u> Assign $r \coloneqq 0$.

<u>Step 13</u>. Determine the intermediate solution $X^t = (x_1^t\ x_2^t, \dots, x_n^t)$ as follows:

For each fixed index $j, (\ j = 1,2, \dots, n;\ \ j \neq k)$

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

Then, the current value of function (1.1) is

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

<u>Step 14.</u> If the condition $f^t > f^{it}$ is satisfied, then set $f^{it} \coloneqq f^t,\ \ X^{it} = (x_1^t,\ x_2^t, \dots, x_n^t)$ and store these values. If $r = 1$, proceed to Step 18.

<u>Step 15</u>. If $x_k^t < d_k$, proceed to Step 17.

<u>Step 16</u>. Set $k\!: = k + 1$. if the condition $k > n(0)$ is satisfied, then set *k:=kk*, and proceed to Step 18.

<u>Step 17</u>. For the values $x_k^t \coloneqq 1,2, \dots, d_k$, set $b_i \coloneqq bb_i - a_{ik} x_k^t \ \ \ , (i = \overline{1,m})$ accordingly and proceed to Step 13.

Step 18. Set $r := 1$. If $x_k^t = d_k$, assign k:=k-1. In this case, if the condition $k < n(d)$ is satisfied, proceed to Step 20.

Step 19. For the values $x_k^t \coloneqq 1,2,\dots,d_k$ set $b_i \coloneqq bb_i - a_{ik}x_k^t(i = \overline{1,m})$, accordingly and proceed to Step 13.

Step 20. Print the obtained final innovative value f^{it}, the innovative approximate solution $X^{it} = (x_1^{it},\ x_2^{it},\dots,x_n^{it})$, and the relative errors $\delta = \left([\tilde{f}] - f^t\right)/\,[\tilde{f}]$ and $\delta^i = \left([\tilde{f}] - f^{it}\right)/[\tilde{f}]$.

Step 21. STOP.

3. Results of the computational experiment

In order to clarify the effectiveness of the innovative approximate solution method proposed in [18] for the integer programming problem, experiments were conducted on numerous problems of various dimensions. The coefficients of these problems were selected as two-digit or three-digit random integers. To conduct the experiments, a program developed based on the algorithm provided above was used. The coefficients for problem (1)-(3) are randomly chosen two-digit or three-digit numbers as follows:

$$0 < c_j < 100, \quad 0 < a_{ij} < 100,\ \left(i = \overline{1,m};\ j = \overline{1,n}\right)$$

or

$$0 < c_j < 1000, \quad 0 < a_{ij} < 1000, \left(i = \overline{1,m};\ j = \overline{1,n}\right),$$

$$b_i = \left[\frac{1}{3}\sum_{j=1}^{n} a_{ij}\right], \quad (i = \overline{1,m}).$$

Here, the notation $[z]$ denotes the integer part of the number z.

It should be noted that during the experiments, 10 problems were solved for each *mxn* dimension. The results obtained are reflected in the tables below. Additionally, we have adopted the following notations in the tables:

N – the problem number.

f^b – the value of function (1) based on the initial solution.

f^i – the value of function (1) provided by the innovative solution.

P – the number of improvements made from the initial solution to the innovative solution.

Δ=$f^i - f^b$ – the difference between the objective function values of the initial and innovative solutions.

$\delta = ((f^i - f^b)/\,f^i) \times 100$ – the relative percentage increase of the innovative value compared to the initial value.

Table 1

Problems with two-digit coefficients($mxn = 10x100$)

N	1	2	3	4	5	6	7	8	9	10
f^b	18821	19636	17781	17736	17940	16518	16562	18301	16521	18025
f^i	18990	19800	18034	17831	18202	16833	17157	18652	16837	18218
P	2	5	4	3	2	6	6	2	6	1
Δ	169	164	253	95	262	315	595	351	31	193
δ	0.889	0.828	1.402	0.533	1.439	1.871	3.468	1.882	0.184	1.059

Table 2

Problems with two-digit coefficients ($mxn = 10x200$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	37553	35234	34660	36477	38814	34712	38225	37018	35470	36003
f^i	37832	35636	34660	36511	38866	34876	38854	37730	35588	36074
P	2	3	0	3	2	3	2	6	2	2
Δ	279	402	0	34	52	164	629	712	118	71
δ	0.737	1.128	0	0.093	0.134	0.003	1.619	1.887	0.331	0.003

Table 3

Problems with two-digit coefficients ($mxn = 10x300$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	53318	54955	55311	51960	55450	54726	55935	56106	55898	54803
f^i	53532	55033	55492	52329	55746	55197	56278	56263	56034	55005
P	2	6	5	6	5	6	6	6	3	2
Δ	214	78	18	36	29	47	343	157	136	202
δ	0.399	0.142	0.032	0.068	0.052	0.085	0.609	0.279	0.243	0.367

Table 4

Problems with two-digit coefficients ($mxn = 10x500$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	91631	92456	91048	92910	91524	89813	91903	92174	91639	93115
f^i	91983	93489	91561	93036	91524	90019	92074	92315	91788	93283
P	2	2	6	2	0	3	4	3	4	6
Δ	352	1033	513	126	0	206	171	141	149	168
δ	0.383	1.104	0.560	0.135	0	0.228	0.186	0.153	0.162	0.180

Table 5

Problems with two-digit coefficients ($mxn = 20x100$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	16776	17041	17787	17466	17198	17568	16724	16958	17393	16890
f^i	16843	17100	18000	17629	17201	17662	16856	17099	17547	17212
P	2	1	2	2	1	1	3	6	2	1
Δ	67	59	213	163	3	94	132	141	154	322
δ	0.398	0.345	1.183	0.925	0.017	0.532	0.783	0.825	0.877	1.871

Table 6

Problems with two-digit coefficients ($mxn = 20x200$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	36184	36080	34766	35481	36855	36568	36114	35118	34502	36108
f^i	36348	36331	34846	35654	37126	36597	36703	35149	34692	36329
P	4	5	2	5	5	2	7	1	3	5
Δ	164	251	80	173	271	29	589	31	190	221
δ	0.451	0.691	0.229	0.485	0.729	0.079	1.605	0.088	0.547	0.608

Table 7

Problems with two-digit coefficients ($mxn = 20x300$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	55198	55269	53353	51764	54767	52516	53104	54175	51779	53263
f^i	55235	55450	53860	51789	54994	52646	53522	54358	51994	53715
P	2	4	5	2	5	3	4	3	5	5
Δ	37	181	507	25	227	130	418	183	215	452
δ	0.067	0.326	0.941	0.048	0.413	0.247	0.781	0.337	0.413	0.841

Table 8

Problems with two-digit coefficients ($mxn = 20x500$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	91268	90388	89230	87994	91133	88792	92699	92124	91475	91045
f^i	91759	90563	89920	88217	91133	88910	92860	92532	92073	91317
P	5	5	7	3	0	5	6	6	4	3
Δ	491	175	690	223	0	118	161	408	598	272
δ	0.535	0.193	0.767	0.252	0	0.133	0.173	0.441	0.649	0.298

Table 9

Problems with three-digit coefficients ($mxn = 10x100$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	189137	192567	176725	177541	183379	161529	168499	186562	162398	176601
f^i	190632	193258	177246	178615	185079	165653	172023	189668	162807	177923
P	2	3	2	3	4	4	3	2	5	4
Δ	1495	691	521	1074	1700	4124	3524	3106	409	1322
δ	0.784	0.357	0.294	0.601	0.918	2.489	2.048	1.637	0.251	0.743

Table 10

Problems with three-digit coefficients ($mxn = 10x200$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	376043	356059	341499	359172	385246	342569	375415	379356	345283	353595
f^i	377044	357537	345237	361398	385324	346653	376140	382070	348344	355661
P	1	4	8	2	1	4	2	6	6	5
Δ	1001	1478	3738	2226	78	4084	725	2714	3061	2066
δ	0.265	0.413	1.083	0.616	0.020	1.178	0.193	0.710	0.879	0.580

Table 11

Problems with three-digit coefficients ($mxn = 10x300$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	537690	557833	552637	518895	558147	539574	555582	549975	559675	535781
f^i	540738	558633	556197	520920	560628	546612	558584	552783	563947	537415
P	3	3	8	4	3	4	7	7	8	3
Δ	3048	800	3560	2025	2481	7038	3002	2808	4272	1634
δ	0.563	0.143	0.640	0.389	0.442	1.287	0.537	0.508	0.757	0.304

Table 12

Problems with three-digit coefficients ($mxn = 10x500$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	918645	911207	897236	922863	945241	892176	914592	918346	903553	921537
f^i	919425	917334	899646	926286	946458	893869	15375	920185	905353	924319
P	2	7	2	2	7	6	2	3	2	3
Δ	780	6127	2410	3423	2481	1693	783	1839	1800	2782
δ	0.084	0.668	0.268	0.369	0.262	0.189	5.092	0.199	0.198	0.300

Table 13

Problems with three-digit coefficients ($mxn = 20x100$)

N	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
f^b	167029	162927	179624	170039	161397	170899	166815	166324	174625	164408
f^i	168622	163848	180791	171696	162699	175227	169308	169104	176742	170243
P	2	1	2	4	1	3	5	2	1	3
Δ	1593	921	1167	1657	1302	4328	2493	2780	2117	5835
δ	0.944	0.562	0.645	0.965	0.800	2.465	1.472	1.643	1.98	3.427

Table 14

Problems with three-digit coefficients ($mxn = 20x200$)

$\boldsymbol{N}$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\boldsymbol{f^b}$	349736	367889	344385	354111	373451	359763	362372	341298	335643	367658
$\boldsymbol{f^i}$	353889	370869	345806	354556	375686	361853	368019	349088	341364	367658
$\boldsymbol{P}$	7	3	4	3	3	4	8	5	5	0
Δ	4153	2980	1421	445	2235	2090	5647	7790	5721	0
$\boldsymbol{\delta}$	1.174	0.803	0.000	0.125	0.595	0.577	1.534	2.231	1.676	0

Table 15

Problems with three-digit coefficients ($mxn = 20x300$)

$\boldsymbol{N}$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\boldsymbol{f^b}$	545303	546506	529668	507809	544962	521301	528902	530259	519653	529442
$\boldsymbol{f^i}$	551774	552694	532980	509162	545433	526404	534831	531666	520031	533150
$\boldsymbol{P}$	4	3	4	3	2	2	5	3	3	4
Δ	6471	6188	3312	1353	471	5103	5929	1407	378	3708
$\boldsymbol{\delta}$	1.172	1.119	0.621	0.265	0.086	0.969	1.108	0.264	0.072	0.695

Table 16

Problems with three-digit coefficients ($mxn = 20x500$)

$\boldsymbol{N}$	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\boldsymbol{f^b}$	904670	899223	905388	884686	894623	886330	915759	910015	905353	915136
$\boldsymbol{f^i}$	907374	900954	908277	886116	898950	890008	924430	913142	908037	918957
$\boldsymbol{P}$	11	5	6	5	4	5	5	5	5	6
Δ	2704	1731	2889	1430	4327	3678	8671	3127	2684	3821
$\boldsymbol{\delta}$	0.298	0.192	0.318	0.161	0.481	0.413	0.937	0.342	0.295	0.416

Based on the tables, the following final conclusions can be drawn.

4. Conclusion

– It is evident from the tables above that a total of 160 problems with both two-digit and three-digit random coefficients were solved. The initially found approximate solution was not improved in only four of these problems. It can be assumed that in those specific cases, the initial solution was already the optimal solution.

– The difference between the innovatively improved value and the initially determined value ranged from 0 to 4,084 units for problems with two-digit coefficients, and from 0 to 8,671 units for problems with three-digit coefficients.

– In the solution process, the number of consecutive improvements of the initial solution was at most 6 for problems with two-digit coefficients, and at most 8 for problems with three-digit coefficients. This means that in problems with larger coefficients, the initial solution differs significantly from the optimal solution. Consequently, in such cases, more frequent improvements occur.

– The maximum relative increase of the value provided by the innovative solution compared to the initial value was 5.092%, which is of significant importance for solving practical problems.

– Based on the tables above, we can conclude that the method proposed in [18] can provide significantly better solutions for real-world applied problems encountered in practice.

References

[1] A.A. Korbut, Yu.Yu. Finkelshteyn, Discrete Programming, Moskva, Nauka, (1969) 368 p.

[2] В.А. Емеличев, В.Н. Комлик, Метод построения последовательности планов для решения задач дискретной оптимизации, Москва, Наука, (1981) 208 p. [In Russian: V.A. Emelichev, V.N. Komlik, Method of Constructing a Sequence of Plans for Solving Discrete Optimization Problems, Moscow: Nauka]

[3] И.Х. Сигал, А.П. Иванова, Введение в прикладное дискретное программирование, Москва, Физмат лит., (2007) p.304. [In Russian: I.Kh. Sigal, A.P. Ivanova, Introduction to Applied Discrete Programming, Moscow: Fizmatlit]

[4] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack problems, Springer-Verlag, Berlin, Heidelberg, New-York, (2004) 546 p.

[5] M.R. Garey, Computers and Intractability: a Guide to the Theory of NP-Completeness, M.R. Garey, D.S. Jhonson, San Francisco, Freeman, (1979) 314 p.

[6] U.U. Finkelshteyn, Priblijennie metodi i prokladnie zadachi diskretnoqo programmirovaniya, Moskva, Nauka, (1976) 264 p.

[7] J.A. Babayev, K. Sh. Mammadov, M.G. Mehdiyev, Methods for constructing suboptimal solutions to the multidimensional knapsack problem, Computational Mathematics and Mathematical Physics, Moscow. 28 No.6 (1978) pp.1443-1453.

[8] M.M. Kovalev, Discrete optimization (integer programming), Moscow, URCA, (2003) 191 p.

[9] K. Sh. Mamedov, Metodi postroeniya priblijennix resheniy mnoqomernoy zadachi o rance i naxojdenie verxney ocenki optimum, K.Sh. Mamedov, T.M. Musaeva, Avtomatika i Vichislitelnaya Texnika. No.5 (2004) pp.72-82. [In Russian: K. Sh. Mamedov, Methods for Constructing Approximate Solutions of the Multidimensional Knapsack Problem and Finding the Upper Bound of the Optimum]

[10] U.G. Nuriyev, An algorithm for solving a multidimensional knapsack problem, Transactions of National Academy of Sciences of Azerbaijan, Series of physical-technical and mathematical sciences. 21 No.2 (2001) pp.54-58.

[11] I.V. Sergienko, V.P. Shilo, Discrete optimization problems: Problems, methods, research, Naukovo Dumka, 2003, 264 p.

[12] И.В. Сергиенко, Л.Н. Козерацкая, Т.Т. Лебедева, Исследование устойчивости и параметрический анализ дискретных оптимизационных задач, Киев, Науково Думка, (1995) 170 p. [In Russian: I.V. Sergienko, L.N. Kozeratskaya, and T.T. Lebedeva, Stability Research and Parametric Analysis of Discrete Optimization Problems, Kyiv: Naukovo Dumka]

[13] T. Erlebach, Approximating multi-objective knapsack problems, T. Erlebach, H. Kellerer, U. Pferschy, Management Science. No.48 (2002) pp.1603-1612.

[14] V.V. Vazirani, Approximation algorithms, V.V. Vazirani, Berlin: Springer, (2001) 378 p.

[15] К.Ш. Мамедов, С.Я. Гусейнов, Методы построения субоптимальных решений задач целочисленного программирования и их последовательное улучшение, Автоматика и Вычислительная Техника. No.6 (2007) pp.20-31. [In Russian: K.Sh. Mammadov and S.Ya. Huseynov, Methods for constructing suboptimal solutions of integer programming problems and their sequential improvement, Avtomatika i Vychislitelnaya Tekhnika]

[16] K.Sh. Mammadov, R.R. Niyazova, S.Y. Huseynov, Innovative approximate method for solving Knapsack problems with interval coefficients, International Independent Scientific Journal. No.44 (2022) pp.8-12.

[17] K.Sh. Mamedov, R.R. Niyazova, Innovative improved approximate solution method for the integer Knapsack problem, Error compression and computational experiments, Radio Electronics, Computer Science, Control. No.4 (2024) pp.64-74.

[18] K.Sh. Mamedov, R.R. Niyazova, An innovative approximate solution method for an integer programming problem, Radio Electronics, Computer Science, Control. No.3 (2025) pp.195-205. https://doi.org/10.15588/1607-3274-2025-3-18