Methods of Constructing Suboptimal Solution of Multiple Mixed-Boolean Programming Problems

KnyazMamedov[1], SevinjNagiyeva[2] , HasanVeliyev[3]
[1]Cybernetics Institute of ANAS, Baku, Azerbaijan,
[2]Baku State University, Baku, Azerbaijan,
[3]Azerbaijan Technical University.
[1]mamedov_knyaz@yahoo.com, [2]sevinc.nagiyeva@gmail.com,

Abstract— **In the article it was offered two methods for finding of approximate solution of mixed-Boolean programming problems with many restrictions. These methods are based on certain criterion which takes into account an economical entity of problem. Computational experiments that fulfilled on different problem shows, solutions that got in these methods don't differ from optimal solution so much.**

Key wordsmixed— Boolean Programming Problem; suboptimal solution; upper and low bound of optimal value;nonlinear penalty

I. INTRODUCTION

Let's consider the following mixed-Boolean programming problem with a lot of restrictions:

$$\sum_{j=1}^{N} c_j x_j \to \max \qquad (1)$$

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

$$0 \le x_j \le 1 \qquad j = \overline{1,N} \ (3)$$

$$x_j = 0 \bigvee 1 \quad j = \overline{1,n} (n \le N) \qquad (4)$$

Assume that, $c_j > 0, \ a_{ij} \ge 0, b_i > 0$

are given numbers. Note that, co-ordinates which get 0 or 1 are arranged consecutive. Without loss of generality, we can always achieve by numbering again. Moreover the problem is a generalization of corresponding linear programming problem and Boolean programming problem. If $n = 0$ then, this problem is turned linear programming problem. If $n = N$ then, the problem (1)-(4) coincides Boolean programming problem. That is why, these problems are special form of problem (1)-(4) [5-7].

In this work two methods of solving multiple mixed-Boolean programming problem are offered. If $n = 0$ then, it should be solved as appropriate linear programming problem, if $n = N$ then, it should be get suboptimal solution of Boolean programming problem in these methods[1-3].

II. PROBLEM STATEMENT AND THEORETICAL GROUND OF THE FIRSTMETHOD

Let's give an economical interpretation to the problem (1)-(4). Suppose that, there are N objects. Some parts of these objects are used or aren't used. Reminder $N - n$ numbers are connected with optimal planning. To do this it will be used m number resources. Denote by $b_i(i = \overline{1,m})$ the volume of the $i-th$ resource. Suppose that, the profit (benefit, effect) of the $j-th$ object is $c_j(j = \overline{1,N})$. The volume of the $i-th$ resource used for production of the unit $j-th$ objectdenote by $a_{ij}(i = \overline{1,m}, j = \overline{1,N})$. That is to say, if it is accepted $x_j = 1$ then, the volume of the $i-th$ resource is used $a_{ij}(i = \overline{1,m}, j = \overline{1,N})$. Here the purpose consists of following: such objects selected for to use and put plan that, total expenditure that used for them shouldn't exceed volume of resources and at same time received profit should be maximum.

It is clear that, in this case the benefit for each unit of $a_{ij}(i = \overline{1,m}, j = \overline{1,N})$ becomes at least

$$k_j = \frac{c_j}{\max_i a_{ij}}.$$

Naturally, the number $\exists j_*$ $(1 \le j_* \le N)$ must be selected that, k_{j_*} is maximum. So we can give following criteria:

$j_1 * = arg\{\max_j (c_j / \max_i [a_{ij}])\}$

We will use the criteria (5) for constructing suboptimal solution of the problem (1)-(4). There is vital importance

$$[1, n]$$

of being number j_* to enter or $[n+1, N]$. Bcause of, we accept sets $H = \{1,2,\dots,N\}$ and $T = \{1,2,\dots,n\}$. It is clear that, T is certain subset of H. If $T = \emptyset$ then, the problem (1)-(4) is turned linear programming problem. If $T = H$ then, the problem (1)-(4) coincides Boolean programming problem. If $a_{ij_*} \le b_i$ for all $i = \overline{1, m}$ then, we take $x_{j_*} := 1, b_i := b_i - a_{ij_*}$ $(i = \overline{1, m})$ $H\square(:=) H\{j_1 *\}$,. Then we select a new j_* number by criteria (5). Otherwise, we already have $\exists i a_{ij_*} > b_i$ then there is vital importance of being $j_* \in H$ or $j_* \in H \backslash T$ .If $j_* \in T$ then, we take $x_{j_*} := 0$,$H \coloneqq H\{j_1 *\}$, $T\square(:=) T\{j_1 *\}$ select next j_* by criteria (5). If $j_* \in H \backslash T$ then , we accepted $x_j \coloneqq 0$ for all $j \in T$, by putting the found values of co-ordinates x_j in to the problem (1)-(4), solve lower-dimensional linear programming problem. By adding the found finally co-ordinates to the co-ordinates found before, we constuct suboptimal solution of problem (1)-(4) $X = (x_1, x_2, \dots, x_N)$.

If we calculate maximum number of operations till begining to solve linear programming problem , then wet get following expression:

+const

If we denote L with number of operations that fulfiled for constructing suboptimal solution of linear programming problem then we have prooved following teorem:

Teorem: Maximum number of fulfiling operations for constructing suboptimal solution of problem (1)-(4) by criteria (5) is as following:

+const

III. THEORETICAL GROUND OF THE SECONDMETHOD

(S)ow we offer second method for constructing suboptimal solution of problem (1)-(4) . We'll call this method "nonlinear penalty". Note that, this method is based [6.p 1450]. At the begining we write the problem (1)-(4) in following equivalent form:

$$\sum_{j=1}^{N} c_j x_j \to \max \qquad (6)$$

$$\sum_{j=1}^{N} a_{ij} x_j \le 1 , \qquad i = \overline{1, m} \qquad (7)$$

$$0 \le x_j \le 1 \qquad j = \overline{1, N} \quad (8)$$

$$x_j = 0 \bigvee 1 \quad j = \overline{1, n} (n \le N) \qquad (9)$$

Here $a_{ij} \coloneqq {}^{a_{ij}}/_{b_i}, b_i \coloneqq 1 \; (i = \overline{1, m}, j = \overline{1, N})$. It is clear that, $0 \le a_{ij} \le 1$. We use approximate solution $X = (0,0\dots,0)$ for constructing suboptimal solution of the problem (6)-(9). At first, we accept

$$Q = \{j | x_j = 1 \;\text{ or}\; x_j = 0 \;\} \qquad (10)$$

and

$$r_i = \sum_{j \in Q} a_{ij} x_j , \qquad i = \overline{1, m} (11).$$

It is clear that, at the begining because of $Q = \emptyset$, we get $r_i = 0 \; (i = \overline{1, m})$ .We have accepted a such penalty

$$t_i = \frac{1}{1 - r_i} (i = \overline{1, m}) \qquad (12)$$

$1 - r_i \; (i = \overline{1, m})$ are reduced, penalty $t_i (i = \overline{1, m})$ are increased. If we accept $x_j = 1$ then, total penalty is as following:

$$S_j = \sum_{i=1}^{m} a_{ij} t_i , \qquad j = \overline{1, N}$$

In this time the benefit of each unit of penalty

$$k_j = \frac{c_j}{S_j} , \qquad j = \overline{1, N}.$$

It is clear that, we must select number $\exists j_*$ $(1 \le j_* \le N)$ that, $\max_j \{k_j\} = k_{j_*}$

and at the same time it should be correct $a_{ij_*} \le b_i$ for all $i = \overline{1, m}$. At this time because of Q changes, then quantities r_i $(i = \overline{1, m})$ and penalties $t_i (i = \overline{1, m})$ will change. As a result, we find a new number j_*. Note that, when we accept $x_{j_*} = 1$, then it must be accepted $H\square(:=) H\{j_1 *\}$, $T \coloneqq T\{j_1 *\}$,$Q \coloneqq Q \cup \{j_*\}$. Otherwise, we already have

$\exists i a_{ij_*} > b_i$, then there is vital importance of being $j_* \in H$ or $j_* \in H \backslash T$. If $j_* \in T$ then, we take $x_{j_*} := 0$, $H\square(:=) H\{j_1 *\}, T \coloneqq T\{j_1 *\}$, $Q \coloneqq Q \cup \{j_*\}$. select next j_* . If $j_* \in H \backslash T$ then , we accepted $x_j \coloneqq 0$ for all $j \in T$, by putting the found values of co-ordinates x_j into the problem (6)-(9), solve lower-dimensional linear programming problem. By adding the found finally coordinates to the coordinates found before, we construct suboptimal solution of problem (6)-(9) $X = (x_1, x_2, \ldots, x_N)$.

IV. RESULTS OF CALCULATING EXPERIMENTS

We have to notice that, if $n = 0$ then the best solutions $\overline{X} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_N)$ of boundless problem (1)-(3) are the solutions of problem (1)-(4) and the process of the solving over. If $n = N$ then we get worst approximate solution such as $\underline{X} = (\underline{x_1}, \underline{x_2}, \ldots, \underline{x_N})$ of problem (1)-(4). So, for the most optimal solution of f^* for the task (1)-(4), f^q, f^{cq} are corrresponding values of first and second method, we can determine the upper ($\overline{f}$) and the lower ($\underline{f}$) limits in the following way:

$$\underline{f} \le f^q \le f_q^* \le \overline{f},$$
$$\underline{f} \le f^{cq} \le f_q^* \le \overline{f}.$$

We do computational experiments on different measured problems. The coefficients of problems have been found in [4]. These coefficients are as following:

$$1 \le c_j \le 99, j = \overline{1, N},$$

$$0 \le a_{ij} \le 99, \qquad i = \overline{1, m}, j = \overline{1, N},$$

$$b_i = \left[\alpha \sum_{j=1}^{N} a_{ij}\right], \qquad i = \overline{1, m}.$$

Here $[z]$ is integer part of number z and $\alpha (0 < \alpha < 1)$ have been marked number.
We use following marks in the tables:
MN – number of solved problem(or experiment),
$\underline{f}$ – lower bound of problem (1)-(4),
f^q – the integral part of value of first method,
f^{cq} – the integral part of value of second method ,
$\overline{f}$ – the integral part of value of upper bound of problem (1)-(4)
δ_1 and δ_2 – relative error.

$$\delta_1 = \frac{(f_q^* - f^q)}{f_q^*} = 1 - \frac{f^q}{f_q^*} \le 1 - \frac{f^q}{\overline{f}},$$

$$\delta_2 = \frac{(f_q^* - f^{cq})}{f_q^*} = 1 - \frac{f^q}{f_q^*} \le 1 - \frac{f^{cq}}{\overline{f}}.$$

Table 1. $m \times N = 20 \times 50, \quad n = 15$

MN	1	2	3
$\underline{f}$	1076	1171	1168
f^q	1176	1248	1293
f^{cq}	1295	1540	1422
$\overline{f}$	1302	1540	1492
δ_1	0.10	0.19	0.13
δ_2	0.01	0	0.04

Table 2. $m \times N = 20 \times 50, \quad n = 30$

MN	1	2	3
$\underline{f}$	1076	1171	1168
f^q	1176	1190	1233
f^{cq}	128	1493	1413
$\overline{f}$	1302	1540	1492
δ_1	0.12	0.22	0.17
δ_2	0.01	0.03	0.05

Table 3. $m \times N = 20 \times 50, \quad n = 45$

MN	1	2	3
$\underline{f}$	1076	1171	1168
f^q	1078	1171	1213
f^{cq}	1281	1491	1411
$\overline{f}$	1302	1540	1492
δ_1	0.17	0.23	0.18
δ_2	0.01	0.03	0.05

Table 4. $m \times N = 30 \times 60, \quad n = 18$

MN	1	2	3
$\underline{f}$	1354	1533	1330
f^q	1376	1575	1492
f^{cq}	1658	1788	1740
$\overline{f}$	1710	1841	1810
δ_1	0.19	0.14	0.17
δ_2	0.03	0.02	0.03

Table 5. $m \times N = 30 \times 60, \quad n = 36$

MN	1	2	3
$\underline{f}$	1354	1533	1330
f^q	1376	1570	1431
f^{cq}	1647	1779	1739

$\overline{f}$	1710	1841	1810
δ_1	0.19	0.14	0.21
δ_2	0.03	0.03	0.03

Table 6. $m \times N = 30 \times 60$, $n = 54$

MN	1	2	3
$\underline{f}$	1354	1533	1330
f^q	1355	153	1337
f^{cq}	1643	1779	1722
$\overline{f}$	1710	1841	1810
δ_1	0.20	0.17	0.26
δ_2	0.03	0.03	0.04

Table 7. $m \times N = 5 \times 90$, $n = 28$

MN	1	2	3
$\underline{f}$	2283	2128	2459
f^q	2350	2248	2518
f^{cq}	2400	2350	2670
$\overline{f}$	2503	2428	2889
δ_1	0.06	0.10	0.12
δ_2	0.04	0.03	0.07

Table 8. $m \times N = 5 \times 90$, $n = 56$

MN	1	2	3
$\underline{f}$	2283	2128	2459
f^q	2299	2190	2503
f^{cq}	2330	2310	2595
$\overline{f}$	2503	2428	2889
δ_1	0.08	0.09	0.13
δ_2	0.06	0.04	0.10

Table 9. $m \times N = 5 \times 90$, $n = 84$

MN	1	2	3
$\underline{f}$	2283	2128	2459
f^q	2287	2153	2471
f^{cq}	2300	2205	2503
$\overline{f}$	2503	2428	2889
δ_1	0.08	0.11	0.14
δ_2	0.08	0.09	0.13

We see that, in most of solved problems second method gives better result .

REFERENCE

[1] Martello S.,Toth P. Knapsack problems, Algorithms and computers implementations. Jhon Wiley & Sons, Chichester, 1990, 296 p.

[2] Kellerer H., Pferschy U., Pisinger D. Knapsack problems. Springer-Verlag, Berlin, Heidelberg, New-york,2004, 546 p.

[3] Kovalev M.M. Discrete optimization (integer programming) M.URSA, 2003, 191 pp(Russian).

[4] Babayev G.A., Mamedov K.Sh., Mektiyev M.Q. Constructions methods of suboptimal solution of multidimensional knapsack problem Zh.V.M & MF 1978, №6,pp. 1443-1453.(Russian)

[5] Nuriyev U.Q. An algorithm for solving of the multidimensional knapsack problem., Transactions Of Azerbaijan National Academy Of Sciences ,2001, T.21,№2. pp. 54-58. (Russian)

[6] Mamedov K.Sh., Musayeva T.M.Methods of constructing approximate solutions of the multidimensional knapsack problem and finding the upper bound of the optimum. j. "Automation and Computer Science", 2004, №5. pp. 72-82. (Russian)

[7] Mansimov K.B., Mamedov K.K. The method of solution of mixed integer knapsack problem narrowing the boundaries of the objective function values and variables. j. "Automation and Computer Science", 2010,№4. pp. 40-53 . (Russian)