APPROXIMATE METHOD FOR SOLVING BOOLEAN PROGRAMMING PROBLEM

Knyaz Mamedov[1], Sahib Mardanov[2], Sagif Huseynov[3]

Institute of Cybernetics of ANAS, Azerbaijan, Baku
[1]*mamedov_knyaz@yahoo.com* , [2]*sahibmardanov@yahoo.com* ,[3]*H_Saqif@box.az*

It is known that Boolean programming problem (BPP) belongs to NP hard problems. Sometimes it is impossible receiving optimal solution of BPP. Since the mathematical models of many economic problems are written as BPP it is necessary to find some suboptimal solution which may be acceptable in practice. For finding suboptimal solutions of BPP are devoted many works [1-4] and suggested a lot of approximate methods. However, the receiving solutions of those methods may be seriously different from an optimal solution. Therefore, developing efficient methods for finding suboptimal solution for BPP are theoretically and practically important. In present work we suggest a new approximate method for solving BPP.

We consider the following problem BPP:

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

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

$$x_j = 1 \vee 0, \; j = \overline{1,n} \;, \tag{3}$$

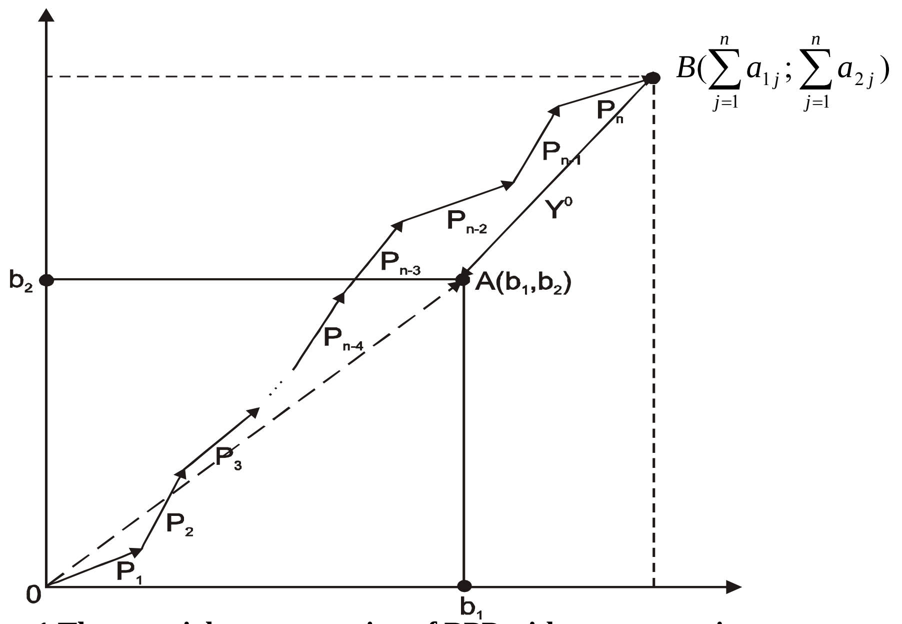
where $c_j > 0$, $a_{ij} \ge 0, b_i > 0$, $j = \overline{1,n}, \; i = \overline{1,m}$.

Here we accept the following notation:

$$P_j = (a_{1j}, a_{2j}, ..., a_{mj})^T, \; \overrightarrow{OA} = P_0 = (b_1, b_2, ..., b_m)^T, \; \overrightarrow{BA} = Y = P_0 - \sum_{j=1}^{n} P_j \;,$$

where $Y^0 = (y_1, y_2, ..., \; y_m \;)$ and $y_i = b_i - \sum_{j=1}^{n} a_{ij}, \; i = \overline{1,m}$.

For understanding of essence of the offered method we will consider geometrical representation of problem (1)-(3) in case $m = 2$.



Picture 1 The vecorial reprezentation of BPP with two constraints

Commonly one may accept that substituting an optimal solution of problem (1)-(3) in system (2) the right side value of system will be closer to point **A** from inside. Therefore, we must construct a process of solution where the sum $P_{j_1} + P_{j_2} + \ldots + P_{j_k}$ will be closer to point **A**. Taking

$$x_j = \begin{cases} 1, & j \in \omega = \{j_1, j_2, \ldots, j_k\} \\ 0, & otherwise \end{cases}$$

we construct a solution of the problem (1)-(3). We will use the following concept for finding all number which belong or not belong to the set ω: we can select as less vectors $P_j \ (j = \overline{1,n})$ which have a little angle with the vector $\overrightarrow{BA}$ to get the nearest side of the point A as it possible and accept $x_j = 0$. It means that we don't include all numbers j to the set ω. In the other hand, we can select as more vectors $P_j \ (j = \overline{1,n})$ which have a little angle with vector $\overrightarrow{OA}$ as it possible and accept $x_j = 1$. In other word we include them to the set ω. The main goal of this concept is that in the approximate solution of the problem (1)-(3) will be more unite than zero.

Notice that in the solution process we should take into account maximization process the value of function (1). For this purpose we can select and include to the set ω the vector P_{j^*} which have a little angle with vector $\overrightarrow{OA}$ and give the maximal added value to the function or we can not select and hence can not include to the set ω the vector P_{j^*} which have a little angle with vector $\overrightarrow{BA}$ and give the minimal added value to the function.

If the angles between vectors P_j and $\overrightarrow{OA}$ or P_j and $\overrightarrow{BA}$ receive a little value then the value of $\cos\left(P_j, \overrightarrow{OA}\right)$ and $\cos\left(P_{j_*}, Y^0\right)$ will be greater, where

$$\cos\left(P_j, \overrightarrow{OA}\right) = \frac{b_1 a_{1j} + b_2 a_{2j} + \ldots + b_m a_{mj}}{\sqrt{b_1^2 + b_2^2 + \ldots + b_m^2}\sqrt{a_{1j}^2 + a_{2j}^2 + \ldots + a_{mj}^2}}$$

$$\cos\left(P_{j_*}, Y^0\right) = \frac{y_1 a_{1j_*} + y_2 a_{2j_*} + \ldots + y_m a_{mj_*}}{\sqrt{y_1^2 + y_2^2 + \ldots + y_m^2}\sqrt{a_{1j_*}^2 + a_{2j_*}^2 + \ldots + a_{mj_*}^2}}$$

Then we can select the following criteria for defining the number j_* for including or not including to the set ω:

$$j_* = arq\max\left\{\max_j\left(c_j \cos\left(P_j, \overrightarrow{OA}\right)\right), \ \max_j\left(\frac{\cos\left(P_j, Y^0\right)}{c_j}\right)\right\} \tag{4}$$

Notice that in the start of calculation we accept $\omega = \{1,2,\ldots,n\}$. If $j_* = arq\max_{j\in\omega}\left\{c_j \cos\left(P_j, \overrightarrow{OA}\right)\right\}$ then we check the condition $a_{ij_*} \le b_i$ for all $i = \overline{1,m}$. If this condition is satisfied for $\forall i \ (i = \overline{1,m})$ then $x_{j_*} = 1$, $b_i := b_i - a_{ij_*}, (i = \overline{1,m})$, otherwise we accept $x_{j_*} = 0$; $\omega = \omega \setminus \{j_*\}$. If we find $j_* = arq\max_{j\in\omega}\left\{\frac{\cos\left(P_j, \overline{Y^0}\right)}{c_j}\right\}$ then we accept

$x_{j_*} = 0,\ \omega := \omega \setminus \{j_*\}$ and from the condition $y_i = b_i - \sum_{j \in \omega} a_{ij}$ we calculate the new value y_i $\left(i = \overline{1,m}\right)$. If $y_i \geq 0$ for $\forall i\ \left(i = \overline{1,m}\right)$, then we accept $x_j = \begin{cases} 1, & j \in \omega \\ 0, & otherwise \end{cases}$ and the vector $X = (x_1, x_2, ..., x_n)$ will be suboptimal solution of the problem (1)-(3). Hence the solution process is stopped. If for some $\exists i\ \left(i = \overline{1,m}\right)$ is satisfied the condition $y_i < 0$, then from condition (4) we find a new number j_*. Then the calculation process to go on as above mentioned rules.

Notice that this presented method is generalization of method [5].

We create computer program for suggesting method and carried out computing experiments. The result of computing experiments is given in the table.

$m \times n$	10×100	10×200	10×1000	50×100	50×200	50×1000
N_1	3	3	4	2	2	1
N_2	7	6	6	7	6	9

Here m be number of constraints, n be number of variables, N_1 and N_2 be the number of best solutions giving the method from work [5] and suggested method. As we see from the table the presented method be better then the method from work [5].

Literature

1. Kellerer H., Pferschy U.,Pisinger D. Knapsack problems. Springer-Verlag, Berlin, Heidelberg, New-York, 2004, 546 p.
2. Martello S., Toth P. Knapsack problems, Algorithm and Computers implementations. John Wiley & Sons, Chichster, 1990.
3. Бабаев Дж.А., Мамедов К.Ш. Приближенные методы решения задач булевого программирования большой размерности. В кн: Вопросы математической кибернетики и прикладной математики. Баку: Элм, 1978.
4. Нуриев У.Г. Об одном алгоритме решения многомерной задачи о ранце // Ж. Изв. НАН Азербайджана, сер.физ.-техн. и матем. наук, 2001, Т.21, № 2, с.54-58.
5. Hüseynov S.Y. Bul proqramlaşdırma məsələsinin suboptimal həllərinin tapılması üçün bəzi alqoritmlər // AMEA-nın xəbərləri fiz.-tex.və riyaziyyat elmləri seriyası. XXIII cild, Bakı, 2003, №3, səh.109-112.