Serdica
Bulgariacae mathematicae publicationes

Сердика
Българско математическо списание

SERDICA — Bulgaricae
mathematicae publicationes
20 (1994) 74-82

AN HEURISTIC ALGORITHM FOR THE MULTIPLE–CHOICE MIN-MAX PROBLEM

VLADIMIR TZOKOV

ABSTRACT. The Min-Max form of the Multiple–Choice Problem is considered in the present paper. An heuristic algorithm based on some heuristic strategies and information from the last simplex tableau of the Linear Programming Relaxation is given. The results of computational experiments are presented.

1. Introduction. Let Π be an optimization problem. Denote by $F(\Pi)$ the feasible region for the problem Π. If $x \in F(\Pi)$, denote by $Z(x)$ the value of the objective function at point x. Let $N = \{1, \ldots, n\}$ and $M = \{1, \ldots, m\}$ be index sets and let $N_1, \ldots, N_k$ be k mutually exclusive sets of variable indices such that

$$N = \bigcup_{i=1}^{k} N_i \quad \text{and} \quad N_h \bigcap N_t = \emptyset \quad \text{for} \quad h \neq t.$$

For these sets define the following set:

$$G = \{x \in B^n \ : \ \sum_{j \in N_i} x_j = 1, \ \ i = 1, \ldots, k\},$$

where $B = \{0, 1\}$. Then the problem Π is called Multiple Choice Problem if $F(\Pi) \subseteq G$.

If A is a matrix with m rows and n columns, $b \in R^m$, $c \in R^n$, then the well known Multiple Choice Integer Programming $(MCIP)$ is

$$(MCIP) \qquad \min\{cx \ : \ Ax \geq b \, , \, x \in G\}.$$

This problem was posed at first by Healey (1964) [5]. A lot of mathematicians have worked on it, namely: Tomlin (1970), Armstrong (1975), Sweeny and Murphy (1981), Bean (1984), etc. The Multiple Choice Knapsack Problem $(MCKP)$, 0–1 Multiple

Knapsack Problem (0–1 MKP), Generalized Assignment Problem (GAP), formulated and described in [1], are important particular cases of ($MCIP$). In this paper we shall consider the Min-Max form of the Multiple Choice Problem and we shall obtain its relationship with ($MCIP$) in Section 2.

Let A be a matrix whose columns are partitioned into blocks (i.e. into index sets $N_1, \ldots, N_k$ respectively). The Multiple Choice Mini-Max Problem is one of selecting exactly one column from each block so that the resulting sum has a minimal maximal component. Denote by a_i, $i \in M$ the vector–rows of A and the Multiple Choice Min-Max Problem can be described by

$$(MCMP)_{min} \qquad \min_{x \in G} \max_{i \in M} (a_i x).$$

If we want to maximize the minimal component of the resulting sum, then we obtain the Multiple Choice Max-Min Problem:

$$(MCMP)_{max} \qquad \max_{x \in G} \min_{i \in M} (a_i x).$$

After transforming $(MCMP)_{min}$ and $(MCMP)_{max}$ into problems of Integer Linear Programming, their models become equivalent, so further we shall envisage both problems with ($MCMP$).

2. Complexity of the Multiple Choice Problems. Now we shall show that these problems are NP–hard in strong sense (we refer the reader to Garey and Johnson (1979) [2] for a thorough discussion on this concept).

Theorem 2.1. *($MCIP$) is NP–hard in strong sense.*

Proof. The Generalized Assignment Problem is NP–hard in strong sense [1]. Since (GAP) is a particular case of ($MCIP$), hence ($MCIP$) is also NP–hard in strong sense. □

Theorem 2.2. *($MCMP$) is NP–hard in strong sense.*

Proof. We shall find a pseudo-polynomial transformation of ($MCIP$) into ($MCMP$). Consider an instance of ($MCIP$):

$$\min \; cx \tag{1}$$

$$a_i x \geq b_i \,, \; i \in M, \tag{2}$$

$$x \in G. \tag{3}$$

Construct the following problem:

$$\min_{x \in G} \max\big(cx\ ,\ L(b_i - a_i x)\ :\ i \in M\big), \tag{4}$$

$$\text{where} \qquad L = \sum_{i=1}^{k} \max_{j \in N_i} c_j + 1.$$

This is a problem $(MCMP)$ for which c is the first vector-row of the matrix and the others can be obtained by

$$d_{ij} = L(\frac{b_i}{k} - a_{ij})\ ,\ j \in N\ ,\ i \in M.$$

Without loss of generality we shall assume that $c_j > 0$ for $j \in N$ because for every $B > 0$ and $y \in G$ the equation

$$\sum_{j \in N}(c_j + B)y_j = \sum_{j \in N} c_j y_j + kB$$

is satisfied. If a solution y of $(MCIP)$ exists, then $cy < L$. Hence if $y \in G$ satisfies all constraints (2), then the maximum in (4) is in the first component and $cy < L$. If $y \in G$ does not satisfy at least one of constraints (2), then the maximum in (4) is at least L.

Thus function f is found, which transforms every instance of $(MCIP)$ into an instance of $(MCMP)$ and this transformation is pseudo-polynomial [2]. Hence $(MCMP)$ is NP–hard in strong sense. □

3. Heuristic strategies. Let AL be an approximate algorithm for solving the optimization problem Π (assume criterion max). If I is an instance of Π denote by $V_{AL}(I)$ the value of the objective function of the solution, given by AL. Let $V_{OPT}(I)$ be the optimal solution value.

Theorem 3.1. *Unless $P \neq NP$, no polynomial approximate algorithm AL for solving $(MCMP)_{max}$ can exist such that $V_{AL}(I)\,T \geq V_{OPT}(I)$ for every instance I of $(MCMP)_{max}$ and arbitrary constant T.*

Proof. Suppose that there exist such an algorithm and a constant. Consider the recognition version of the Minimal Set Covering Problem $R(MSCP)$: A set of items and n subsets, $k < n$ are given. Question: Is there a set covering of the given set with at most k subsets?

This problem is NP–complete [2]. Let an example of $R(MSCP)$ be given. Construct k identical sets of vector–columns. Each vector represents a subset by

$$a_{ij} = \begin{cases} (k+1)T & \text{item i is in subset j} \\ 1 & \text{otherwise} \end{cases}.$$

If such a set covering exists, then for the corresponding problem $(MCMP)_{max}$ we have $V_{OPT}(I) > kT$ and else $V_{OPT}(I) = k$. Hence for AL we have that $V_{AL}(I) > k$ if such a set covering exists and $V_{AL}(I) \leq k$ otherwise. So the algorithm AL solves exactly every instance of $R(MSCP)$ and hence $P = NP$, which is a contradiction. □

3.1. 2–opt strategy

This strategy was applied to a number of other problems such as the Travelling Salesman Problem [3].

For every $x \in G$ we denote by $Q(x) = \{y \in G : y$ differs from x in exactly two components $\}$.

Definition 3.2. *$x \in G$ is called local minimum for $(MCMP)_{min}$ if $Z(x) \leq Z(y)$ for every $y \in Q(x)$.*

It is obvious that $|Q(x)| = n - k$ for every $x \in G$. A local minimum is found throught the following simple procedure by starting from $x \in G$:

```
PROCEDURE 2-OPT(X);
FLAG := 0;
WHILE (FLAG = 0) DO
   find y ∈ Q(x) with minimal Z(y);
   IF (Z(y) < Z(x)) THEN X := Y ELSE FLAG := 1;
END WHILE;
RETURN(X);
```

3.2. Greedy strategy. This strategy is well known from 0–1 Knapsack Problem [2].For $w \in R^m$ we define $\lceil w \rceil = \max_{i \in M}(w_i)$. Let $\alpha, \beta \in R^m$; $X, Y \in B^n$. Consider the following procedure:

```
PROCEDURE GREEDY(N_i, i = 1, k);
   α := 0⃗;
   β := 0⃗;
   X := 0⃗;
   Y := 0⃗;
   FOR i := 1 TO k DO
select j ∈ N_i : ⌈α + a_j⌉ is minimal;
select l ∈ N_i : Σ_{i∈M} a_il is minimal;
α := α + a_j;
β := β + a_l;
X_j := 1;
```

$Y_l := 1;$
END FOR;
IF ($\lceil\alpha\rceil > \lceil\beta\rceil$) THEN X := Y;
RETURN(X);

The time complexity of this procedure is $O(mn)$.

Lemma 3.3. *Let X be the solution obtained by the Greedy procedure and let $a_{ij} \geq 0$ for $i \in M$, $j \in N$. Assume that X^0 is an optimal solution of the problem $(MCMP)_{min}$. Then $Z(X) \leq \min(k, m)\, Z(X^0)$.*

Proof. (i) Obviously $\sum_{i \in M} \beta_i \leq \sum_{i \in M} (AX^0)_i$ is satisfied and since $a_{ij} \geq 0$ for every $i \in M$ and $j \in N$ hence $\lceil\beta\rceil \leq m \lceil AX^0\rceil = m\, Z(X^0)$.
(ii) We define

$$\mu_i = \min_{j \in N_i} \lceil a_j \rceil$$

and denote by s the index such that $\mu_s \geq \mu_i$ for $i = 1, \ldots, k$.

On the one hand, solution X^0 includes a vector from N_s and it follows that $Z(X^0) = \lceil AX^0\rceil \geq \mu_s$. On the other hand we have that for every $i = 1, \ldots, k$ exists $j(i)$: $\lceil a_{j(i)}\rceil \leq \mu_s$.

We denote by α^i the vector α at the end of i–th pass in FOR loop. Now we show that $\lceil\alpha^i\rceil \leq i\,\mu_s$:

$\lceil\alpha^1\rceil \leq \mu_s$ because the first vector is selected with minimal maximal component.

Suppose that $\lceil\alpha^{i-1}\rceil \leq (i-1)\mu_s$. Then we have $\lceil\alpha^i\rceil \leq \lceil\alpha^{i-1} + a_{j(i)}\rceil \leq \lceil\alpha^{i-1}\rceil + \lceil a_{j(i)}\rceil \leq (i-1)\mu_s + \mu_s = i\,\mu_s$. Hence $\lceil\alpha\rceil \leq k\,\mu_s \leq k\, Z(X^0)$.

From (i) and (ii) we obtain immediately that $Z(X) \leq \min(k, m)\, Z(X^0)$ □

The value $\min(k, m)$ cannot be decreased. Consider the following example : we have $k = m$ identical sets of vector–columns. We have $|N_i| = m + 1 = k + 1$ for $i = 1, \ldots, k$. The vectors of N_i are the vector–columns of the following matrix :

$$M_i = \begin{pmatrix} p & p+1 & 0 & 0 & \ldots & 0 \\ p & 0 & p+2 & 0 & \ldots & 0 \\ p & 0 & 0 & p+2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p & 0 & 0 & 0 & \ldots & p+2 \end{pmatrix},$$

where $p \in Z_+$. Applying the Greedy procedure, we obtain

$$\alpha = \begin{pmatrix} k\,p \\ k\,p \\ \vdots \\ k\,p \end{pmatrix}, \beta = \begin{pmatrix} k(p+1) \\ 0 \\ \vdots \\ 0 \end{pmatrix}, AX^0 = \begin{pmatrix} p+1 \\ p+2 \\ \vdots \\ p+2 \end{pmatrix}$$

with $\lceil AX \rceil = k\,p$. It is obvious that $\lceil AX^0 \rceil = p+2$ and

$$\lim_{p \to \infty} \frac{p+2}{k\,p} = \frac{1}{k}.$$

This example also shows that the 2–OPT strategy cannot improve X because $Z(X) \leq Z(Y)$ for every $Y \in Q(X)$.

4. An heuristic algorithm. A survey on the heuristic algorithms for the $(MCIP)$ shows that in most cases they exploit the information from the last simplex tableau of the Linear Relaxation Problem (LRP) [6], [7]. The idea of the algorithm which we propose is to combine this information with described heuristic strategies.

The Linear Relaxation Problem of the $(MCMP)_{min}$ is

$$(LRP) \qquad \min \ \xi$$

$$(5) \qquad \xi - \sum_{j \in N} a_{ij} x_j \geq 0\,, \ i \in M,$$

$$(6) \qquad \sum_{j \in N_r} x_j = 0\,, \ r = 1, \ldots, k,$$

$$x_j \geq 0\,, \ j \in N.$$

Denote by $\overline{X}$ the optimal solution of this problem; u_i – dual variable values, corresponding to constraints (5), $i \in M$; v_r – dual variable values, corresponding to constraints (6), $r = 1, \ldots, k$.

If $j \in N$, then we denote by $s(j)$ the index of the subset in which j is included, so $j \in N_{s(j)}$. Now we can determine the reduced costs of the variables by

$$r_j = \sum_{i \in M} a_{ij} u_i - v_{s(j)}, \ \ j \in N.$$

Lemma 4.1. *Let $X \in G$ such that $X_j = 1$ for some $j \in N$. Then $Z(X) \geq Z(\overline{X}) + r_j$.*

Proof. The dual problem of (LRP) is equivalent to the Lagrangean dual obtained by dualizing the constraints (5):

$$(LG) \qquad \max_{u\geq 0} \min_{x\in G} \sum_{i\in M} u_i(\sum_{j\in N} a_{ij}x_j) \ : \ \sum_{i\in M} u_i = 1.$$

Consider the problem with the additional constraint $x_j = 1$. Then we have $Z(LG : x_j = 1) \geq Z(LG) + r_j = Z(LRP) + r_j$ and since Z(LR) is a lower bound of every solution of $(MCMP)_{min}$, finally we obtain that $Z(X) \geq Z(\overline{X}) + r_j$. □

We shall eliminate the variables with greater reduced costs. From the dual theory we have that if $\overline{X_j} \neq 0$, then $r_j = 0$. Therefore for every basis variable r_j is zero. Then we propose the following costs of the variables :

$$(7) \qquad p_j = r_j - \overline{X_j} = \sum_{i\in M} a_{ij}u_i - v_{s(j)} - \overline{X_j}, \ j \in N$$

and the heuristic algorithm can be described as follows:

```
PROCEDURE HOPT;
FOR i := 1 TO k DO N̄ᵢ := Nᵢ;
solve LRP and determine X̄, u, v;
determine pⱼ for j ∈ N from (7);
sorting the variables so that p₁ ≥ p₂ ≥ ... ≥ pₙ;
R := ∞ ;
t := 1 ;
WHILE (t ¡ n) DO
   CALL GREEDY(N̄ᵢ,i=1,k);
  X^g := X;
   CALL 2-OPT(X);
   IF (Z(X) < R) THEN
      R := Z(X);
      H := X;
   END IF ;
   FL := 0;
   WHILE (FL=0) DO
      i := s(t);
      IF (|N̄ᵣ| = 1 for r = 1,...,k) THEN
         FL := 1;
         t := n;
      END IF;
```

```
        IF (|N_i| > 1) THEN
          N_i := N_i \ {t};
          IF (X_t^g = 1) THEN FL := 1;
        END IF;
        t := t+1;
      END DO;
    END DO;
    RETURN(H);
```

5. Computational results. The algorithm has been encoded in *C* and applied to ten randomly generated examples with different sizes (numbers vary in the range of 0,999). The program was run on PC 486, 5755.8 KWhetstones/sec. The times, optimal values and the values of the heuristic solutions are given in the following table:

Table 1: PC–AT 486 in seconds.

Problem	m	sets	vectors	V_{OPT}	V_H	Time
p1	2	20	200	4287	4287	0.055
p2	2	50	1000	7172	7196	0.22
p3	5	10	200	2781	2816	0.11
p4	5	20	500	5162	5183	0.385
p5	5	50	1000	13627	13668	2.198
p6	10	10	100	3931	3931	0.165
p7	10	20	200	7643	7643	0.33
p8	10	20	500	6787	6877	1.593
p9	20	10	100	4602	4740	0.824
p10	20	10	200	4468	4508	1.484

The exact algorithm, described in [8] has been used for comparision. The solution of problem $p10$, through this algorithm takes more than 3 hours.

REFERENCES

[1] S. MARTELLO, P. TOTH. Knapsack Problems. John Wiley & Sons, 1990

[2] M.R. GAREY, D.S. JOHNSON. Computers and Intractability, A Guide to the theory of NP–Completeness. W. H. Freeman and Company, San Francisco, 1979.

[3] H. SALKIN, K. MATHUB. Foundations of integer programming. North–Holland, 1989.

[4] J.C. BEAN. A Langrangian algorithm for the Multiple Choice Integer Programming. *Operations Research*, **32** (1984), 1185-1193.

[5] W.C. HEALEY, Multiple Choice Programming. *Operations Research*, **12** (1964), 122-138.

[6] D.J. SWEENEY, R.A. MURPHY. Branch–and–bound methods for multi–item sheduling. *Operations Research*, **29** (1981), 853-864.

[7] S.G. CHANG, D.W.TCHA. A heuristic for multiple Choice Programming. *Computers and Operations Research*, **12** (1985), 25-37.

[8] N. YANEV, V. TZOKOV. An algorithm for a Min-Max Combinatorial Problem. *Mathematics and education of mathematics*, **22** (1993), 163-173 (in Bulgarian).

Sofia University "St. KL. Ohridski"
Faculty of Mathematics and Informatics
5, James Bouchier str
1126 Sofia
BULGARIA

Received 18.05.93
Revised 17.06.93