THE OPTIMAL ASSIGNATION OF WORKERS ON JOBS FROM THE POINT OF VIEW OF MINIMIZATION THE MAXIMAL EXECUTION TIME

Catalin Angelo IOAN[1]

[1]*Danubius University of Galati, Department of Economics, catalin_angelo_ioan@univ-danubius.ro*

Abstract. In this paper we shall give a new solution for the optimal assignation of workers on jobs from the point of view of minimization the maximal execution time using the simplex algorithm which can solve the problem using computers instead the known graphical solution.

Keywords: Simplex, assignation, minimization

1 Introduction

The problems of assignation appear usual in the process of targets allocation in an institution.

Let consider $A'=\{A_1,...,A_{n'}\}$ the set of workers in an institution and $L'=\{L_1,...,L_{m'}\}$ the set of jobs which must be executed at a specific moment.

In the execution of job L_j the worker A_i can spend t_{ij} units of time.

Because each worker can has a multiple qualification, but not all necesary for the entire set of jobs we put the problem of allocation on jobs such that the maximum time spent in the execution to be minimal.

Let therefore $f:A'\to P(L')$, $f(A_i)=\{L_{i_1},...,L_{i_k}\}$ $\forall i=1,...,n'$ the function who assign to A_i the jobs: $L_{i_1},...,L_{i_k}$ which he can realize if he has the necessary qualification for at least one job and $f(A_i)=\varnothing$ in opposite cases.

We shall restrict the set A' and we shall consider, from the beginning, the subset of those workers for which $f(A_i)\neq\varnothing$ $\forall A_i\in A$. We shall note therefore $A=\{A_1,...,A_n\}$ with $n\le n'$ (*after a possible renotation of workers*). Let now (*again after a possible renotation of workers*): $\bigcup_{i=1}^{n} f(A_i)=\{L_1,...,L_m\}$ with $m\le m'$. If $m<m'$ we have that the jobs $L_{m+1},...,L_{m'}$ cannot be executed from any workers, therefore will be excludes.

Finally, let consider: $L=\{L_1,...,L_m\}$ and the new allocation function: $f:A\to P(L)$.

We shall define a matrix:

$$M=\begin{array}{cc} \begin{array}{ccc} L_1 & ... & L_m \end{array} & \\ \begin{pmatrix} a_{11} & ... & a_{1m} \\ ... & ... & ... \\ a_{n1} & ... & a_{nm} \end{pmatrix} & \begin{array}{c} A_1 \\ ... \\ A_n \end{array} \end{array}$$

where $a_{ij}=1$ if the worker A_i can execute the job L_j and 0 in the other cases.

We shall build the matrix $T=(t_{ij})$ of execution times, assigning $t_{ij}=\infty$ if A_i cannot execute L_j.

The graphical method of Ducamp, presented in [2], proposes a construction of a simple graph (a decomposition of nodes in two disjoint subsets: workers and jobs) and after an initial allocation a succesion of improvements based on graphical observations. This method is good but cannot be easly implemented on computers.

We shall propose in what follows a new method based on the Simplex algorithm.

2 The method of Simplex algorithm

Let now, the matrix $M_t=(a^t_{ij})$ where:

$$a^t_{ij}=\begin{cases} a_{ij} \text{ if } t_{ij} \le t \\ 0 \text{ if } t_{ij} > t \end{cases}$$

and $A_t=(\alpha^t_{ij})$ where:

$$\alpha^t_{ij}=\begin{cases} 1 \text{ if } A_i \text{ will assign to execute } L_j \text{ in a time less than or equal with } t \\ 0 \text{ if } A_i \text{ will not assign to execute } L_j \text{ in a time less than or equal with } t \end{cases}$$

Let now the matrix $B_t=(\alpha^t_{ij}a^t_{ij})$ which elements belong to {0,1} and who has the following significance: $\alpha^t_{ij}a^t_{ij}=1$ if A_i will be assigned to execute the job L_j in a time less than or equal with t and he is qualified for this thing, and $\alpha^t_{ij}a^t_{ij}=0$ in the other cases.

Because any worker cannot execute two jobs in the same time we shall have: $\sum_{j=1}^{m} a^t_{ij}\alpha^t_{ij} \le 1$ $\forall i=$.

Also, because any job cannot be executed in the same time by two different workers we shall have: $\sum_{i=1}^{n} a^t_{ij}\alpha^t_{ij} \le 1$ $\forall j=\overline{1,m}$.

After these conditions follows: $a^t_{ij}\alpha^t_{ij} \le 1$ $\forall i=\overline{1,n}$ $\forall j=\overline{1,m}$.

The allocation problem becomes (for a maximal time t):

$$\begin{cases} \max\left(\sum_{i=1}^{n}\sum_{j=1}^{m} a^t_{ij}\alpha^t_{ij}\right) \\ \sum_{j=1}^{m} a^t_{ij}\alpha^t_{ij} \le 1 \\ \sum_{i=1}^{n} a^t_{ij}\alpha^t_{ij} \le 1 \\ \alpha^t_{ij} \ge 0 \end{cases}$$

Let remark first that the problem has always a solution for a suitable t.

Let now $t_k=\min\{t \mid M_t$ has at least k rows who have an element equal with $1\}$.

We have obviously: $\min t_{ij} \le t_1 \le t_2 \le \ldots \le t_n \le \max t_{ij}$.

The algorithm will begin with $t=t_n$. If the problem will not have a solution, we shall grow t with one unit until we shall find a maximal allocation.

If we cannot find such allocation, we shall consider $t=t_{n-1}$ and begin again the problem.

One problem can appear after sloving: what is happened if the solutions will not be entire? It is possible, for example, on the i-th row to be a lot of elements equal with 1 (appropriate to the fact that one worker can execute a few jobs), say k elements, and the optimal solution to contains the variables: $\alpha^t_{ij_1}=\alpha^t_{ij_2}=\ldots=\alpha^t_{ij_k}=\frac{1}{k}$. Because the objective function is $\sum_{i=1}^{n}\sum_{j=1}^{m} a^t_{ij}\alpha^t_{ij}$ it follows that it will not modify if we replace all the cited values with, for example: for a $1 \le p \le k$.

3 References

1. Ioan C.A., Mathematics II, Ed. Sinteze, Galati, Romania, 2008
2. Stancu-Minasian I.M., The study of the simulation problem using the graph theory, Studies and researches of economical calculus and cibernetics, nr.5, 1970
3. Teodorescu N. (coord.), Methods of operational researchs, Ed. tehnica, Bucharest, 1972