A SOLVING METHOD OF TWO ECONOMICAL PROBLEMS USING LINEAR PROGRAMMING IN INTEGER NUMBERS

Catalin Angelo IOAN
Danubius University, Galati, Romania

Abstract:

In this paper we shall give a new solution for the optimal assignation of workers on jobs from the point of view of execution total time minimization using the Simplex algorithm which can solve the problem using computers instead the known Little's solution. In the second section, we shall give a new solution for the optimal assignation of workers on jobs from the point of view of maximize the number of allocates workers, using the Simplex algorithm which can solve the problem using computers instead the known graphical solution.

Keywords *Simplex, assignation, minimization*

Classification codes and keywords: *C70*

I The Optimal Assignation Of Workers From The Point Of View Of Execution Total Time Minimization

The problems of assignation appear usual in the process of targets allocation in an institution.
Let consider A={A1,...,An} the set of workers in an institution and L={L1,...,Lm} the set of jobs which must be executed at a specific moment. In the execution of Lj, the worker Ai spends a time equal with tij units (hours, minutes, seconds etc.). Supposing that it exists workers which can execute a lot of jobs we put the problem of allocation on jobs such that the total time spending in the execution to be minimum.

We shall assign an infinite value to tij if Ai is not able to execute the job Lj. Also, we shall understand that the number of workers is equal with those of jobs, in the opposite case introducing fictional workers or jobs with infinite times of execution to prevent the allocation of them.

The method of Little suggests the following steps:

Step 1 It is build the table of times (with workers on columns and jobs on rows) and after we shall compute the minimum on each row. After this we subtract these values from those of rows, compute the minimum on each column and after also, we shall subtract these from the values on the columns. After this step, on each row or column is at least one value equal with 0.

Step 2 We shall compute the sum of all elements subtracted from rows and columns and noted with S1.

Step 3 For each element equal with 0 in the last table, we shall compute the quantities μij=min {tik│k≠j}+min{tpj│p≠i} or, in other words, the sum of the elements on the row and column corresponding to the null quantity. After this, we shall determine the maximum of that values and the appropriate allocation (s,r). We shall build a tree graph where the initial knot comes with the value S1. We shall build after a bend where we shall put the activities (s,r) and non(s,r) who will come with the values αsr and βsr=S1+μsr respectively.

Step 4 We shall erase the row s and the column r and we shall act like in the first step.

Step 5 We shall compute S2 like sum of the elements of minimum of rows and columns and we shall modify the indicator αsr=S1+S2.

Step 6 If the simplified table will has only one row and column the algorithm will close. If not it will be choose the minimum between αsr and βsr. If both values will be equal we shall choose the value αsr appropriate to an allocation and not to a reject of allocation.

Step 7 If the choice value was αsr we shall return at the step 3.

Step 8 If the choice value was βsr then we shall consider in the table previously of step 1: tsr=∞ and we shall compute the minimum of row s and column r, subtract these from the appropriate row and column and return at the third step.

We can see that the algorithm is a little hard therefore we shall propose in what follows a new method based on the Simplex algorithm.

II The New Method For Optimal Assignation Of Workers From The Point Of View Of Execution Total Time Minimization

Let consider A'={A1,...,An'} the set of workers in an institution and L'={L1,...,Lm'} the set of jobs which must be executed at a specific moment.

Let therefore f:A'→P(L'), f(Ai)= $\{L_{i_1},...,L_{i_k}\}$, ∀i=1,...,n' the function who assign to Ai 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(Ai)=∅ in opposite cases.

We shall restrict the set A' and we shall consider, from the beginning, the subset of those workers for which f(Ai)≠∅ ∀Ai∈A. We shall note therefore A={A1,...,An} with n≤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≤m'. If m<m' we have that the jobs Lm+1,...,Lm' cannot be executed from any workers, therefore will be excludes.

Finally, let consider: L={L1,...,Lm} and the new allocation function: f:A→P(L).

We shall define a matrix:

$$M=\begin{array}{cccc} L_1 & \ldots & L_m & \\ \left(\begin{array}{c} a_{11} \\ \ldots \\ a_{n1} \end{array}\right. & \begin{array}{c} \ldots \\ \ldots \\ \ldots \end{array} & \left.\begin{array}{c} a_{1m} \\ \ldots \\ a_{nm} \end{array}\right) & \begin{array}{c} A_1 \\ \ldots \\ A_n \end{array} \end{array}$$

where aij=1 if the worker Ai can execute the job Lj and 0 in the other cases.

Let now consider the matrix A=(αij) where:

$$\alpha_{ij}=\begin{cases} 1\text{ if the worker } A_i \text{ will nominate in the execution of } L_j \\ 0\text{ if the worker } A_i \text{ will not nominate in the execution of } L_j \end{cases}$$

We shall, like in the previous section, build the matrix T=(tij) of execution times, assigning tij=∞ if Ai cannot execute Lj.

In a distinction with Little's method we shall not enjoin restrictions to the number of workers or jobs.

Let now the matrix B=(αijaij) who's elements belong to the set {0,1} and who has the following meaning: αijaij=1 if Ai will nominate to execute Lj and is also qualified for this thing and αijaij=0 in the other cases.

Because no one can execute two jobs simultaneously, we have therefore the condition: $\sum_{j=1}^{m} a_{ij}\alpha_{ij} \le 1$ $\forall i=\overline{1,n}$..

Also, because any job cannot be execute simultaneously by two different workers we have that: $\sum_{i=1}^{n} a_{ij}\alpha_{ij} \le 1$ $\forall j=\overline{1,m}$.

From the above conditions it follows that: aijαij$\le$1 $\forall i=\overline{1,n}$ $\forall j=\overline{1,m}$.

The allocation problem will become:

$$\begin{cases} \min(\sum_{i=1}^{n}\sum_{j=1}^{m} t_{ij}\alpha_{ij}) \\ \sum_{j=1}^{m} a_{ij}\alpha_{ij} \le 1 \\ \sum_{i=1}^{n} a_{ij}\alpha_{ij} \le 1 \\ \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}\alpha_{ij} = M \\ \alpha_{ij} \ge 0 \end{cases}$$

where M is the number of workers proposed for the execution.

Before solving the problem, let remark first that if it isn't a maximal allocation the problem will not have a solution and in other case if it has at the final we shall obtain effective the allocation. The value of minimum will be the searched total time.

The problem will be solved in the following manner: we start with the value M=n. If it has not a solution we diminish M with a unit and we begin again to solve the new problem. Because M is a free term in the upper problem we shall reoptimize the older.

The process is obviously finite because the problem has always a solution at least for M=0: αij=0.

III The Optimal Assignation Of Workers On Jobs

Let consider now A'={A1,...,An'} the set of workers in an institution and L'={L1,...,Lm'} the set of jobs which must be executed at a specific moment.

Because each worker can has a multiple qualification, but not all necessary for the entire set of jobs we put the problem of allocation on jobs such that they realize too much if it is possible of them.

Let therefore f:A'→P(L'), f(Ai)= $\{L_{i_1},...,L_{i_k}\}$ $\forall$i=1,...,n' the function who assign to Ai 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(Ai)=∅ in opposite cases.

We shall restrict the set A' and we shall consider, from the beginning, the subset of those workers for which f(Ai)≠∅ ∀Ai∈A. We shall note therefore A={A1,...,An} with n≤n' (after a possible renotation of workers). Let now (again after a possible renotation of workers): $\bigcup_{i=1}^{n} f(A_i)$ ={L1,...,Lm} with m≤m'. If m<m' we have that the jobs Lm+1,...,Lm' cannot be executed from any workers, therefore will be excludes.

Finally, let consider: L={L1,...,Lm} and the new allocation function: f:A→P(L).

We shall define a matrix:

$$M=\begin{array}{c|ccc|c} & L_1 & \dots & L_m & \\ \hline & a_{11} & \dots & a_{1m} & A_1 \\ & \dots & \dots & \dots & \dots \\ & a_{n1} & \dots & a_{nm} & A_n \end{array}$$

where aij=1 if the worker Ai can execute the job Lj and 0 in the other cases.

The graphical method 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 succession of improvements based on graphical observations. This method is good but cannot be easily implemented on computers.

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

Let now, the matrix A=(αij) where:

$$\alpha_{ij}=\begin{cases}1 \text{ if the worker } A_i \text{ will execute the job } L_j \\ 0 \text{ if the worker } A_i \text{ will not execute the job } L_j\end{cases}$$

and the matrix B=(αijaij) with elements in the set {0,1}. We have that αijaij=1 if the worker Ai will execute the job Lj and if he is qualified for this thing and αijaij=0 if the worker Ai will not execute the job Lj or he is not qualified to do this. How any worker cannot execute two jobs in the same time, we have the condition: $\sum_{j=1}^{m} a_{ij}\alpha_{ij} \le 1 \ \forall i=\overline{1,n}$.

Because a job cannot be executed in the same time by two workers we have also that: $\sum_{i=1}^{n} a_{ij}\alpha_{ij} \le 1$ $\forall j=\overline{1,m}$. From these conditions we have now that: aijαij≤1 ∀i=$\overline{1,n}$ ∀j=$\overline{1,m}$.

The problem becomes now the following linear programming:

$$\begin{cases}\max(\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}\alpha_{ij}) \\ \sum_{j=1}^{m} a_{ij}\alpha_{ij} \le 1 \\ \sum_{i=1}^{n} a_{ij}\alpha_{ij} \le 1 \\ \alpha_{ij} \ge 0\end{cases}$$

Because αij=0 verify the restrictions we have that the problem has always a solution. 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: $\alpha_{ij_1} = \alpha_{ij_2} = \dots = \alpha_{ij_k} = \frac{1}{k}$. Because the objective function is $\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}\alpha_{ij}$ it follows that it will not modify if we replace all the cited values with, for example: $\alpha_{ij_p} = 1$ for a 1≤p≤k.

Example

Let the workers A1,A2,A3 and the jobs L1,L2,L3 which posibility of execution is in the following table:

Worker	Jobs
A1	L1,L3
A2	L1,L2
A3	L2

Considering the matrix M= $\begin{array}{cccc} L_1 & L_2 & L_3 & \\ \left(\begin{matrix} 1 \\ 1 \\ 0 \end{matrix}\right. & \begin{matrix} 0 \\ 1 \\ 1 \end{matrix} & \left.\begin{matrix} 1 \\ 0 \\ 0 \end{matrix}\right) & \begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \end{array}$ and A= $\begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix}$ we have the following linear programming problem:

$$\begin{cases} \max(\alpha_{11}+\alpha_{13}+\alpha_{21}+\alpha_{22}+\alpha_{32}) \\ \alpha_{11}+\alpha_{13}\le 1 \\ \alpha_{21}+\alpha_{22}\le 1 \\ \alpha_{32}\le 1 \\ \alpha_{11}+\alpha_{21}\le 1 \\ \alpha_{22}+\alpha_{32}\le 1 \\ \alpha_{13}\le 1 \\ \alpha_{11},\alpha_{13},\alpha_{21},\alpha_{22},\alpha_{32}\ge 0 \end{cases}$$

with the solution: α13=1, α32=1, α21=1. We have therefore that A1 will execute the job L3, A2 – L1 and A3 – L2.

Bibliography

Ioan C.A., 2008. *Mathematics II*, Ed. Sinteze, Galati

Stancu-Minasian I.M., 1970. *The study of the simulation problem using the graph theory*", Studies and researches of economical calculus and cibernetics, nr.5.

Teodorescu N. (coord.), 1972. *Methods of operational researchs*", Ed. tehnica, Bucharest.