New Methods in Mathematical Management of Organization

Mathematician Ciprian Ioan
Bucharest University
Associate Professor Catalin Angelo Ioan, PhD
"Danubius" University of Galaţi
Alin Ioan
Bucharest University

Abstract: In the first part, we shall unify the principal criterions (Wald, Hurwicz, Savage, Laplace) used in the process of choice the best alternative. After the general theory and few examples who illustrate the drawbacks of the existing criterions, we propose six new choices modalities from other points of view. In the second part, 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 third, 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. In part four, we shall give a new solution for the optimal assignation of workers on jobs using the Simplex algorithm which can solve the problem using computers instead the known graphical solution. In part five, we shall give a new algorithm instead of Johnson classical in the process of determination the sequence of pieces execution on two installations without initial deliverance times.

Keywords: games theory, Savage, Simplex, assignation, minimization, Electre method

Jel Classification: C70, C35, C30, C15

1. An Unified Theory Concerning Principal Decision Making Methods

1.1. Introduction

In the process of decision making we have as well as principal methods the following:

- Wald's criterion (the maximin criterion)
- Laplace's criterion
- Hurwicz's optimist criterion
- Savage's regret criterion

Each of these criterions becomes with a series of inconveniences because they broach the problem from a narrow point of view.

For example, let make an analysis of the following problem:

Let a_1, a_2, a_3, a_4 the alternatives and b_1, b_2, b_3, b_4, b_5 uncontrollable states. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	b_3	b_4	b_5
a_1	6	2	8	0	9
a_2	0	3	8	2	3
a_3	0	5	5	1	2
a_4	1	2	3	6	8

If we apply Wald, Hurwicz with 0.1 and Savage we will find that the alternative a_4 is the best, but the Laplace's criterion gives us the alternative a_1.

For the problem:

	b_1	b_2	b_3	b_4	b_5
a_1	8	3	7	2	1
a_2	7	5	3	1	5
a_3	3	4	2	1	9
a_4	5	5	8	2	3

we will find that Wald, Hurwicz with 0.2 and Laplace give us the alternative a_4, but Savage: a_2.

For the problem:

	b_1	b_2	b_3	b_4	b_5
a_1	0	4	1	7	5

a_2	6	9	5	0	2
a_3	8	5	2	5	6
a_4	3	4	5	4	3

we will find that Laplace, Hurwicz with 0.6 and Savage give us the alternative a_3, but Wald: a_4.

For the problem:

	b_1	b_2	b_3	b_4	b_5
a_1	3	7	6	8	3
a_2	1	5	9	9	2
a_3	6	8	1	7	0
a_4	0	9	2	0	0

we will find that Laplace, Wald and Savage give us the alternative a_1, but Hurwicz with 0.7: a_2.

Another example gives us for each criterion another alternative:

	b_1	b_2	b_3	b_4	b_5
a_1	7	2	10	6	2
a_2	9	10	2	1	8
a_3	5	3	6	4	8
a_4	5	6	9	2	6

- Wald's criterion gives us a_3
- Laplace's criterion gives us a_2
- Hurwicz's optimist criterion with 0.6 gives us a_1
- Savage's regret criterion gives us a_4

It is therefore a necessity to broach the problem from two points of view: to create a general criterion applicable on all situations and which recover in particular cases the upper criterions and, on the other hand, to create departing from this general criterion other news.

1. 2. The general problem and criterion

Let $A=\{a_1,\dots,a_m\}$ the set of the alternatives and $B=\{b_1,\dots,b_n\}$ the set of uncontrollable states. The payoffs for each pair (a_i,b_j) are c_{ij}, $i=\overline{1,m}$, $j=\overline{1,n}$.

We shall group in what follows the uncontrollable states in p>=1 subsets of B: $G_k=\{ b_{j_{k-1}+1},\dots, b_{j_k} \}$, $k=\overline{1,p}$ where $0=j_0<j_1<\dots<j_p=n$. These subsets can appear, for example, in the process of grouping the states after their origin.

To each group G_k we assign a risk coefficient $\omega_k\in[0,1]$, $k=\overline{1,p}$ and a weight in decision $\eta_k\in[0,1]$, $k=\overline{1,p}$ such that $\sum_{k=1}^{p}\eta_k=1$.

For each state b_j we note with R_j, $j=\overline{1,n}$ the potential gain if we know apriority the occurrence of b_j.

Finally, let group the alternatives A in q subsets: $H_v=\{ a_{i_{v-1}+1},\dots, a_{i_v} \}$, $v=\overline{1,q}$ where $0=i_0<i_1<\dots<i_q=m$ and for each H_v we assign a coefficient of preference $\lambda_v\in[0,1]$, $v=\overline{1,q}$. Even if at the first sight all the alternatives are equals in probability, in fact the factor of decision has preferences and he split A in the subsets H_v with coefficients of preference λ_v.

We shall define for each row the function:

$$f(i)=\mathrm{sgn}\left(1-2\,\mathrm{sgn}\left(\sum_{j=1}^{n}R_j^2\right)\right)\sum_{k=1}^{p}\eta_k\left(\omega_k\max_{j_{k-1}+1\le s\le j_k}(c_{is}-R_s)+(1-\omega_k)\min_{j_{k-1}+1\le s\le j_k}(c_{is}-R_s)\right)$$

called the expected gain function.

We define the selection group function:

$$g(v)=\lambda_v\max_{i_{v-1}+1\le s\le i_v}f(s)+(1-\lambda_v)\min_{i_{v-1}+1\le s\le i_v}f(s),\ v=\overline{1,q}$$

The alternative's group finally selected is that H_r for which

$$g(r)=\mathrm{sgn}\left(1-2\,\mathrm{sgn}\left(\sum_{j=1}^{n}R_j^2\right)\right)\max_{1\le v\le q}\left(\mathrm{sgn}\left(1-2\,\mathrm{sgn}\left(\sum_{j=1}^{n}R_j^2\right)\right)g(v)\right)$$

The final alternative is that for which the difference $|f(i)\text{-}g(r)|$, $i=\overline{i_{r-1}+1, i_r}$ is minimum.

The table of values and strategies of α and β respectively has the following format:

p groups

q groups

<table>
<tr><td colspan="3" rowspan="3">α/β</td><td>…</td><td colspan="3">G_k</td><td>…</td><td rowspan="3">f</td><td rowspan="3">g</td></tr>
<tr><td>…</td><td colspan="3">ω_k / η_k</td><td>…</td></tr>
<tr><td>…</td><td>$b_{j_{k-1}+1}$</td><td>…</td><td>b_{j_k}</td><td>…</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td rowspan="3">H_v</td><td rowspan="3">λ_v</td><td>$a_{i_{v-1}+1}$</td><td>…</td><td>$c_{i_{v-1}+1, j_{k-1}+1}$</td><td>…</td><td>$c_{i_{v-1}+1, j_k}$</td><td>…</td><td>$f(i_{v-1}+1)$</td><td rowspan="3">g(v)</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>a_{i_v}</td><td>…</td><td>$c_{i_v, j_{k-1}+1}$</td><td>…</td><td>c_{i_v, j_k}</td><td>…</td><td>$f(i_v)$</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td colspan="3"></td><td>…</td><td>$R_{j_{k-1}+1}$</td><td>…</td><td>R_{j_k}</td><td>…</td><td></td><td>$\max\limits_{1\le v\le q} g(v)$ / $\min\limits_{1\le v\le q} g(v)$</td></tr>
</table>

1.3. Particular cases

We shall present, in what follows, the values of the expected gain function f corresponding at different values of p and R_j, $j=\overline{1,n}$ respectively.

Table 1

p	R_j	f(i)
1	$\sum\limits_{j=1}^{n} R_j^2 = 0$	$\omega_1 \max\limits_{1\le s\le n}(c_{is}) + (1-\omega_1)\min\limits_{1\le s\le n}(c_{is})$

1	$\sum_{j=1}^{n} R_j^2 \neq 0$	$\omega_1 \min_{1\le s\le n}(R_s - c_{is}) + (1-\omega_1)\max_{1\le s\le n}(R_s - c_{is})$
1<p<n	$\sum_{j=1}^{n} R_j^2 = 0$	$\sum_{k=1}^{p} \eta_k \left(\omega_k \max_{j_{k-1}+1\le s\le j_k}(c_{is}) + (1-\omega_k)\min_{j_{k-1}+1\le s\le j_k}(c_{is}) \right)$
1<p<n	$\sum_{j=1}^{n} R_j^2 \neq 0$	$\sum_{k=1}^{p} \eta_k \left(\omega_k \min_{j_{k-1}+1\le s\le j_k}(R_s - c_{is}) + (1-\omega_k)\max_{j_{k-1}+1\le s\le j_k}(R_s - c_{is}) \right)$
n	$\sum_{j=1}^{n} R_j^2 = 0$	$\sum_{k=1}^{n} \eta_k c_{ik}$
n	$\sum_{j=1}^{n} R_j^2 \neq 0$	$\sum_{k=1}^{n} \eta_k (R_k - c_{ik})$

The selection group function is for q=m ($i_0=0$, $i_1=1,\ldots, i_m=m$):

$$g(v)= \lambda_v \max_{v\le s\le v} f(s) + (1-\lambda_v)\min_{v\le s\le v} f(s) = \lambda_v f(v)+(1-\lambda_v)f(v)=f(v),\ v=\overline{1,m}$$

If q=1 then: $i_0=0$, $i_1=m$ therefore:

$$g(1)= \lambda_1 \max_{1\le s\le m} f(s) + (1-\lambda_1)\min_{1\le s\le m} f(s)$$

If $R_j=0$, $j=\overline{1,n}$ we have that the alternative's group finally selected is that H_r for which:

$$g(r)= \max_{1\le v\le q}\big(g(v)\big)$$

and if $\exists$ $j=\overline{1,n}$ such that $R_j\neq 0$ follows:

$$g(r)= -\max_{1\le v\le q}\big(-g(v)\big) = \min_{1\le v\le q}\big(g(v)\big)$$

1.4. Known criterions like particular cases

1.4.1. Hurwicz's criterion

For p=1, q=m, R_j=0, j=$\overline{1,n}$ we have that: f(i)=$\omega_1 \max_{1\le s\le n}(c_{is}) + (1-\omega_1)\min_{1\le s\le n}(c_{is})$ and g(v)=f(v), v=$\overline{1,m}$.

The best alternative is that a_r for which g(r)=$\max_{1\le v\le m} f(v)$.

The extreme values of ω_1 are:

- ω_1=0 who lead us to the expected gain function: f(i)=$\min_{1\le s\le n}(c_{is})$ and after: g(r)=$\max_{1\le v\le m} f(v) = \max_{1\le v\le m}\min_{1\le s\le n}(c_{vs})$ - α showing a pessimistic maximum in the choice of the strategy;
- ω_1=1 who lead us to the expected gain function f(i)=$\max_{1\le s\le n}(c_{is})$ and after: g(r)=$\max_{1\le v\le m} f(v) = \max_{1\le v\le m}\max_{1\le s\le n}(c_{vs})$ - α showing an optimistic maximum in the choice of the strategy.

If the first strategy, corresponding to ω_1=0 is a little realistic (punting on a doubtless gain), the second is totally irrational, α ignoring all the opponent's actions (hoping in the weakest choice of β).

Example

Let a_1, a_2 the alternatives of α and b_1, b_2 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table where with ω_1=0,6 we shall apply the Hurwicz criterion:

	b_1	b_2	$c_i = \min_{j=1,\dots,n} c_{ij}$	$C_i = \max_{j=1,\dots,n} c_{ij}$	$0{,}6C_i+0{,}4c_i$
a_1	10	20	10	20	16
a_2	-10	100	-10	100	56

The maximum of the quantities in the last column is 56, therefore the alternative a_2 will be the best from the point of view of Hurwicz's criterion.

If we shall carefully analyze the upper table, we shall see that α will win in the situation of a_2 if and only if β will choose the state b_2. On the other hand, β will never choose this strategy, because regardless what α will adopt, he will lose. Like a consequence we can say that the Hurwicz's criterion is good when the values corresponding to the rows of the table will be near on to the other, the differencies between minimal and maximal values being little.

1.4.2. Wald's criterion

The Wald's criterion is a particular case of Hurwicz's for $\omega_1=0$. Like in the preceding criterion, those of Wald neglected much of the information, treating only minimal values on rows.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	$\min\limits_{j=1,\dots,n} c_{ij}$
a_1	0,9	100	0,9
a_2	1	2	1

The maximum quantities in the last column is 1, therefore the alternative a_2 will be the best after Wald.

If we shall examine carrefully the upper table, we shall see that α will gain with one unit. It is hard to believe that α not choose the alternative a_1, because this can bring an earning between 0,9 to 100 units.

1.4.3. Laplace's criterion

For p=n and $R_j=0$, $j=\overline{1,n}$ we have: $f(i)=\sum_{k=1}^{n}\eta_k c_{ik}$. If $\eta_k=\frac{1}{n}$, $k=\overline{1,n}$ then we obtain:

$f(i)=\frac{1}{n}\sum_{k=1}^{n}c_{ik}$. For q=m we have: $g(v)=f(v)$, $v=\overline{1,m}$.

The winner group of strategies will be a_r for which $g(r)=\max\limits_{1\le v\le m} g(v)$.

The Laplace's criterion has like drawback the equal treatment of all actions of β. In fact, β looking at the values can prefer one or other from his alternatives, who leads to an inequality in the probabilities upper considered.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	$\frac{1}{2}b_1+\frac{1}{2}b_2$
a_1	-10	10	0
a_2	-10	20	5

The maximum in the last column is 5, therefore the alternative a_2 will be choosed after Laplace's criterion.

We can easly see that if α will choose the alternative a_2, he will win if β will choose the strategy b_2. On the other hand, β will choose always b_1 that brings it, regardless α 10 units. As a conclusion, the actions of β cannot have equal probability, in the upper case the probability of β to choose b_1 being 1 and for b_2 being 0.

1.4.4. Savage's criterion

For p=1, $\omega_1=0$ and $R_j=\max_{1\le i\le m} c_{ij}$, $j=\overline{1,n}$, respectively q=m we have: $f(i)=\max_{1\le s\le n}(R_s - c_{is})$ and g(v)=f(v), $v=\overline{1,m}$. The winner alternative will be those a_r for which $g(r)=\min_{1\le v\le m}\left(g(v)\right)$.

The Savage's criterion bring us, at the first sight, a new point of view.

A number of remarks appear however: Savage defines the regret like difference between how much can α win if he had known apriori the decision of β and how much he wins in fact. This definition is credible, but pushes this notion to an extreme. Maybe, in fact a best regret's definition can be an average (with differents weights or not) of possible gains. Also, finally the last section of this algorithm uses the minimax criterion; therefore like in the precedings criterions it not takes in calculus all the values on rows.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2
a_1	-98	101
a_2	1	1
$\max\limits_{1\le i\le m} c_{ij}$	1	101

The regrets table is:

	b_1	b_2	$\max\limits_{1\le s\le n}(R_s - c_{is})$
a_1	99	0	99
a_2	0	100	100

After the minimax criterion we find that the alternative a_1 is the best. We can see that this decision has the highest risk (α can win 101 units, but he can loose also 98 units). The alternative a_2 is, in this case a little practical (it guarantees an earning of 1 unit in any situation).

1.5. New criteria

In what follows, we shall suggest a few criterions deduced from the general formulas.

1.5.1. Hurwicz-Savage's criterion

For p=1, $\omega_1 \neq 0$ and $R_j = \max\limits_{1\le i\le m} c_{ij}$, $j=\overline{1,n}$, respectively q=m we have:

f(i)= $\omega_1 \min_{1\le s\le n}(R_s - c_{is}) + (1-\omega_1)\max_{1\le s\le n}(R_s - c_{is})$ and g(v)=f(v), v=$\overline{1,m}$.

The winner alternative will be a_r for which g(r)=$\min_{1\le v\le m}(g(v))$.

This criterion proposes the determination of the best strategy, assigning a risk factor $\omega_1 \neq 0$ in the process of regrets analyzing.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table, where ω_1=0,6:

	b_1	b_2
a_1	10	20
a_2	-10	100
$\max_{1\le i\le m} c_{ij}$	10	100

	b_1	b_2	$c_i = \min_{j=1,...,n} c_{ij}$	$C_i = \max_{j=1,...,n} c_{ij}$	$0,6C_i+0,4c_i$
a_1	0	80	0	80	48
a_2	20	0	0	20	12

From the last column we see that the strategy a_1 will be the best following this criterion.

We can see that this example is those from Hurwicz's criterion, the alternative a_1, obtained here, being acceptable in comparasion with those of Hurwicz.

1.5.2. Weight Laplace's criterion

For p=n and R_j=0, j=$\overline{1,n}$ we have: f(i)=$\sum_{k=1}^{n}\eta_k c_{ik}$ with $\sum_{k=1}^{p}\eta_k$ =1. For q=m we have: g(v)=f(v), v=$\overline{1,m}$.

The winner alternative will be a_r for which g(r)=$\max_{1\le v\le m} g(v)$.

The Weight Laplace's criterion is a refinement of the classical criterion, assigning to the strategies of β, differents probabilities.

The principal problem who arrived here is that of the choice modality of the weights η_k, $k=\overline{1,n}$.

1.5.3. Proportionally weight Laplace's criterion

For p=n and $R_j=0$, $j=\overline{1,n}$ we have: $f(i)=\sum_{k=1}^{n}\eta_k c_{ik}$ with $\sum_{k=1}^{p}\eta_k=1$.

We shall compute first: $\nu_k=\sum_{t=1}^{m}c_{tk}$, $k=\overline{1,n}$ that is the sum of gains corresponding to the k-th column.

If $\max_{p=\overline{1,n}}\nu_p=\min_{p=\overline{1,n}}\nu_p$ then ν_k=constant, $k=\overline{1,n}$. In this case, we shall apply the Laplace's criterion with all weights equal: $\eta_k=\frac{1}{n}$.

If $\max_{p=\overline{1,n}}\nu_p>\min_{p=\overline{1,n}}\nu_p$, we shall compute: $\varepsilon_k=\dfrac{\max_{p=\overline{1,n}}\nu_p-\nu_k}{\max_{p=\overline{1,n}}\nu_p-\min_{p=\overline{1,n}}\nu_p}$, $k=\overline{1,n}$ and, finally:

$\eta_k=\dfrac{\varepsilon_k}{\sum_{p=1}^{n}\varepsilon_p}$, $k=\overline{1,n}$. We have therefore: $\eta_k=\dfrac{\dfrac{\max_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}-\sum_{t=1}^{m}c_{tk}}{\max_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}-\min_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}}}{\sum_{p=1}^{n}\dfrac{\max_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}-\sum_{t=1}^{m}c_{tp}}{\max_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}-\min_{s=\overline{1,n}}\sum_{t=1}^{m}c_{ts}}}$, $k=\overline{1,n}$.

For q=m we have: $g(v)=f(v)$, $v=\overline{1,m}$.

The winner alternative will be a_r for which $g(r)=\max_{1\le v\le m}g(v)$.

The proportionally weight Laplace's criterion propose a rational choice of β's probabilities of action because, how much the values corresponding to a column of β

are less (designiting β's loses) so much the values ε_k will be elder and, implicit, those of η_k.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2, b_3 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	b_3	$\frac{1}{3}b_1+\frac{1}{3}b_2+\frac{1}{3}b_3$
a_1	-10	10	6	2
a_2	-15	20	7	4

Applying the Laplace criterion we have that the alternative a_2 is the best.

We can easly see that if α choose a_2, he win if β will choose b_2 or b_3. On the other hand, β will choose always the strategy b_1 that brings a greater gain than or equal with 10 units.

We shall apply now, the proportionally weight Laplace's criterion.

We have first: ν_1=-10-15=-25, ν_2=10+20=30, ν_3=6+7=13, from where $\max\{\nu_1,\nu_2,\nu_3\}=30$, $\min\{\nu_1,\nu_2,\nu_3\}=-25$.

We have therefore:

$$\varepsilon_1=\frac{30+25}{55}=1,\ \varepsilon_2=\frac{30-30}{55}=0,\ \varepsilon_3=\frac{30-13}{55}=\frac{17}{55}$$

and finally:

$$\eta_1=\frac{1}{1+0+\frac{17}{55}}=\frac{55}{72},\ \eta_2=0,\ \eta_3=\frac{\frac{17}{55}}{\frac{72}{55}}=\frac{17}{72}$$

The table is:

	55/72	0	17/72	
	b_1	b_2	b_3	$\frac{55}{72}b_1+0\cdot b_2+\frac{17}{72}b_3$
a_1	-10	10	6	-448/72
a_2	-15	20	7	-706/72

The maximum value in the last column is -448/72 therefore the best alternative will be a_1 - most rationally because β will choose b_1 and α will loose less.

1.5.4. Proportionally weight with regrets Laplace's criterion

For p=n and $R_j=\max_{1\le i\le m} c_{ij}$, $j=\overline{1,n}$ we have: $f(i)=\sum_{k=1}^{n}\eta_k(R_k-c_{ik})$ with $\sum_{k=1}^{p}\eta_k=1$.

We shall compute the weights η_k like in the preceding criterion but, in this case, for the regrets table of β.

First, we shall compute, the regrets: $S_i=\max_{1\le j\le n}(-c_{ij})=-\min_{1\le j\le n} c_{ij}$, $i=\overline{1,m}$ and after we shall build the regrets table of β, having the elements $d_{ik}=-c_{ik}-S_i=\min_{1\le j\le n} c_{ij}-c_{ik}$, $i=\overline{1,m}$, $k=\overline{1,n}$.

Determining after: $\nu_k=\sum_{t=1}^{m} d_{tk}$, $k=\overline{1,n}$ that is the sum of gains in the column k, we have that if $\max_{p=\overline{1,n}}\nu_p=\min_{p=\overline{1,n}}\nu_p$ then ν_k=constant, $k=\overline{1,n}$. In this case we shall apply the Laplace's criterion with equal weights: $\eta_k=\frac{1}{n}$.

If $\max_{p=\overline{1,n}}\nu_p>\min_{p=\overline{1,n}}\nu_p$, we compute: $\varepsilon_k=\frac{\nu_k-\min_{p=\overline{1,n}}\nu_p}{\max_{p=\overline{1,n}}\nu_p-\min_{p=\overline{1,n}}\nu_p}$, $k=\overline{1,n}$ and, finally: $\eta_k=\frac{\varepsilon_k}{\sum_{p=1}^{n}\varepsilon_p}$, $k=\overline{1,n}$. We have therefore:

$$\eta_k=\frac{\dfrac{\sum_{t=1}^{m} d_{tk}-\min\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}}{\max\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}-\min\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}}}{\sum_{p=1}^{n}\dfrac{\sum_{t=1}^{m} d_{tp}-\min\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}}{\max\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}-\min\limits_{s=\overline{1,n}}\sum_{t=1}^{m} d_{ts}}},\ k=\overline{1,n}$$

For q=m we find that: $g(v)=f(v)$, $v=\overline{1,m}$.

The winner alternative will be a_r for which $g(r)=\min\limits_{1\le v\le m}(g(v))$.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2, b_3 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	b_3	$\min\limits_{1\le j\le n} c_{ij}$
a_1	-10	10	6	-10
a_2	-15	20	7	-15
$\max\limits_{1\le i\le m} c_{ij}$	-10	20	7	

The regrets table of β (*in terms of gains of* α) is:

	b_1	b_2	b_3
a_1	0	-20	-16
a_2	0	-35	-22

We have therefore:

$\nu_1=0$, $\nu_2=-55$, $\nu_3=-38$, from where: $\max\{\nu_1,\nu_2,\nu_3\}=0$, $\min\{\nu_1,\nu_2,\nu_3\}=-55$.

$$\varepsilon_1=\frac{0+55}{0+55}=1,\ \varepsilon_2=\frac{-55+55}{0+55}=0,\ \varepsilon_3=\frac{-38+55}{0+55}=\frac{17}{55}$$

$$\eta_1=\frac{1}{1+0+\frac{17}{55}}=\frac{55}{72},\ \eta_2=0,\ \eta_3=\frac{\frac{17}{55}}{\frac{72}{55}}=\frac{17}{72}$$

The regrets table of α will be:

	55/72	0	17/72	
	b_1	b_2	b_3	$\frac{55}{72}b_1+0\cdot b_2+\frac{17}{72}b_3$
a_1	0	10	1	17/72
a_2	5	0	0	275/72

The minimum of the last column being 17/72 it follows that the best strategy is a_1.

1.5.5. Regrets Laplace's criterion

For p=n and $R_j=\max_{1\le i\le m} c_{ij}$, $j=\overline{1,n}$, we have: $f(i)=\sum_{k=1}^{n}\eta_k(R_k-c_{ik})$ with $\sum_{k=1}^{p}\eta_k=1$. If we shall choose $\eta_k=\frac{1}{n}$, and after, for q=m, we have that: g(v)=f(v), $v=\overline{1,m}$.

The winner alternative will be a_r for which $g(r)=\min_{1\le v\le m}(g(v))$.

Example

Let a_1, a_2 the alternatives of α and b_1, b_2, b_3 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	b_3
a_1	-10	10	6
a_2	-15	20	7
$\max_{1\le i\le m} c_{ij}$	-10	20	7

The regrets table of α is:

	b_1	b_2	b_3	$\frac{1}{3}b_1+\frac{1}{3}b_2+\frac{1}{3}b_3$
a_1	0	10	1	11/3
a_2	5	0	0	5/3

The best alternative from this criterion is a_2 but from the facts exposed upper this is not acceptable.

1.5.6. The nostalgia criterion

This criterion alludes, in fact, to the final selection of the alternative. After each of the exposed criterions we obtain a series of values of the function f which in the absence of regrets is maximized, and in the presence - minimized.

In many cases, we can group the alternatives of α in categories, clases after the satisfactions offered in the past. We can also group, for example, after the implement expenses of those (*advertising if the problem study the launching of a product*).

Thus, we shall associate to each q groups of alternatives of α a coefficient of importance λ_v, $v=\overline{1,q}$. We shall after determine the selection function: $g(v)=\lambda_v \max\limits_{i_{v-1}+1\le s\le i_v} f(s)+(1-\lambda_v)\min\limits_{i_{v-1}+1\le s\le i_v} f(s)$, $v=\overline{1,q}$ acting after like in section 2 that is: the alternative's group finally selected is that H_r for which

$$g(r)=\operatorname{sgn}\left(1-2\operatorname{sgn}\left(\sum_{j=1}^{n} R_j^2\right)\right)\max_{1\le v\le q}\left(\operatorname{sgn}\left(1-2\operatorname{sgn}\left(\sum_{j=1}^{n} R_j^2\right)\right)g(v)\right)$$

and the final alternative is that for which the difference $|f(i)-g(r)|$, $i=\overline{i_{r-1}+1, i_r}$ is minimum.

The groups of alterantives will be determined, in principle, arbitrary. We can group, for example, in good, medium or weak strategies after the sum of gains. The coeficcients λ_v will be determined after the method indicated in the proportionally weight Laplace's criterion applied on the rows of the groups.

We have therefore the following steps:

We compute first: ν_v, $v=\overline{1,q}$ - the sum of gains of the group v.

If $\max\limits_{p=\overline{1,q}} \nu_p = \min\limits_{p=\overline{1,q}} \nu_p$ then ν_v=constant, $v=\overline{1,q}$. In this case, it follows that isn't a preference for one of the group, and the algorithm will be close like those initial.

If $\max\limits_{p=\overline{1,q}} \nu_p > \min\limits_{p=\overline{1,q}} \nu_p$, we compute: $\varepsilon_v=\dfrac{\nu_v - \min\limits_{p=\overline{1,q}} \nu_p}{\max\limits_{p=\overline{1,q}} \nu_p - \min\limits_{p=\overline{1,q}} \nu_p}$, $v=\overline{1,q}$ and, finally:

$\lambda_v=\dfrac{\varepsilon_v}{\sum\limits_{p=1}^{q} \varepsilon_p}$, $v=\overline{1,q}$.

Example

Let a_1, a_2, a_3, a_4 the alternatives of α and b_1, b_2, b_3 the uncontrollable states of β. The payoffs for each pair (a_i,b_j) are c_{ij} and are find out in the following table:

	b_1	b_2	b_3	$\min\limits_{1\le i\le m} c_{ij}$
a_1	10	10	6	6
a_2	-15	-9	7	-15
a_3	-5	40	50	-5
a_4	30	-8	6	-8

If we apply the Wald criterion we have that the best alternative is a_1.

We shall broach in a different way the problem. The sum on the rows is:

	b_1	b_2	b_3	$\sum\limits_{j=1}^{n} c_{ij}$
a_1	10	10	6	26
a_2	-15	-9	7	-17
a_3	-5	40	50	85
a_4	3	-8	6	28

We shall therefore group the alternatives a_1 and a_2 which offer the less prices and a_3 with a_4. Let therefore: $H_1=\{a_1,a_2\}$ and $H_2=\{a_3,a_4\}$.

We have now: $\nu_1=26-17=9$, $\nu_2=85+28=113$ and $\max\{\nu_1,\nu_2\}=113$, $\min\{\nu_1,\nu_2\}=9$.

Because $\varepsilon_1=\frac{9-9}{113-9}=0$, $\varepsilon_2=\frac{113-9}{113-9}=1$ we obtain: $\lambda_1=0$ and $\lambda_2=1$.

The selection function is:

- $g(1)=\lambda_1 \max\limits_{1\le s\le 2} f(s)+(1-\lambda_1)\min\limits_{1\le s\le 2} f(s)=\min\limits_{1\le s\le 2} f(s)=\min\{26,-17\}=-17$.
- $g(2)=\lambda_2 \max\limits_{3\le s\le 4} f(s)+(1-\lambda_2)\min\limits_{3\le s\le 4} f(s)=\max\limits_{3\le s\le 4} f(s)=\max\{85,28\}=85$.

The winner group is those for which:

$$g(r)=\max_{1\le v\le q}\left(g(v)\right)=\max\{g(1),g(2)\}=85$$

If we compute now the differences: $|f(i)-g(r)|$ $i=3,4$ we obtain: $|f(3)-g(2)|=|85-85|=0$ şi $|f(4)-g(2)|=|28-85|=57$ from which the best alternative is a_3.

2. The Optimal Assignation of Workers from the Point of View of Execution Total Time Minimization

2.1. 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 L_j, the worker A_i spend a time equal with t_{ij} units (hours, minutes, seconds etc.). Supposing thet 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 infinte value to t_{ij} if A_i is not able to execute the job L_j. 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 infinte times of execution to prevent the allocation of them.

The method of Little suggest 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 S_1.

Step 3 For each element equal with 0 in the last table, we shall compute the quantities $\mu_{ij}=\min\{t_{ik} \mid k\neq j\}+\min\{t_{pj} \mid p\neq i\}$ 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 S_1. We shall build after a bend where we shall put the activities (s,r) and non(s,r) who will come with the values α_{sr} and $\beta_{sr}=S_1+\mu_{sr}$ 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 S_2 like sum of the elements of minimum of rows and columns and we shall modify the indicator $\alpha_{sr}=S_1+S_2$.

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 α_{sr} and β_{sr}. If both values will be equal we shall choose the value α_{sr} appropriate to an allocation and not to a reject of allocation.

Step 7 If the choiced value was α_{sr} we shall return at the step 3.

Step 8 If the choiced value was β_{sr} then we shall consider in the table previously of step 1: $t_{sr}=\infty$ and we shall compute the minimum of row s and column r, subtract these form 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.

2.2. The new method

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.

Let therefore $f:A'\rightarrow P(L')$, $f(A_i)=\{L_{i_1},...,L_{i_k}\}$ $\forall i=1,...,n'$ 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 $n\leq n'$ (*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 $m\leq m'$. 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: $f:A\rightarrow P(L)$.

We shall define a matrix:

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

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

Let now consider the matrix $A=(\alpha_{ij})$ 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 $T=(t_{ij})$ of execution times, assigning $t_{ij}=\infty$ if A_i cannot execute L_j.

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

Let now the matrix $B=(\alpha_{ij}a_{ij})$ who's elements belong to the set $\{0,1\}$ and who has the following meaning: $\alpha_{ij}a_{ij}=1$ if A_i will nominate to execute L_j and is also qualified for this thing and $\alpha_{ij}a_{ij}=0$ 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: $a_{ij}\alpha_{ij} \le 1 \ \forall i=\overline{1,n} \ \ \forall j=\overline{1,m}$.

The allocation problem will become:

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

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: $\alpha_{ij}=0$.

3. The Optimal Assignation of Workers on Jobs from the Point of View of Minimization the Maximal Execution Time

3.1. 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 $f(A_i)=\emptyset$ in opposite cases.

We shall restrict the set A' and we shall consider, from the beginning, the subset of those workers for which $f(A_i)\neq\emptyset$ $\forall A_i\in A$. 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:

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

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.

3.2. 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: $\sum_{j=1}^{m} a^t_{ij}\alpha^t_{ij} \le 1 \; \forall i=\overline{1,n}$.

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: $\min t_{ij} \le t_1 \le t_2 \le ... \le t_n \le \max t_{ij}$.

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: $\alpha^{t}_{ij_1} = \alpha^{t}_{ij_2} = ... = \alpha^{t}_{ij_k} = \frac{1}{k}$.

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: $\alpha^{t}_{ij_p} = 1$ for a $1 \le p \le k$.

4. The Optimal Assignation of Workers on Jobs

4.1. 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.

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 they realize too much if it is possible of them.

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≤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 $L_{m+1},...,L_{m'}$ 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:

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

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

The graphical method presented in [1] 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.

4.2. The method of Simplex algorithm

Let now, the matrix A=(α_{ij}) 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=($\alpha_{ij}a_{ij}$) with elements in the set {0,1}. We have that $\alpha_{ij}a_{ij}=1$ if the worker A_i will execute the job L_j and if he is qualified for this thing and $\alpha_{ij}a_{ij}=0$ if the worker A_i will not execute the job L_j 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: $a_{ij}\alpha_{ij}\le 1\ \forall i=\overline{1,n}$ $\forall j=\overline{1,m}$.

The problem becomes now the following linear programming:

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

Because $\alpha_{ij}=0$ 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: $\alpha_{ij_1} = \alpha_{ij_2} = \ldots = \alpha_{ij_k} = \frac{1}{k}$.

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 $1\le p\le k$.

Example

Let the workers A_1,A_2,A_3 and the jobs L_1,L_2,L_3 which posibility of execution is in the following table:

Worker	Jobs
A_1	L_1,L_3
A_2	L_1,L_2
A_3	L_2

Considering the matrix $M=\begin{matrix} & L_1 & L_2 & L_3 & \\ & \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} & & & \begin{matrix} A_1 \\ A_2 \\ A_3 \end{matrix} \end{matrix}$ 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: $\alpha_{13}=1$, $\alpha_{32}=1$, $\alpha_{21}=1$. We have therefore that A_1 will execute the job L_3, $A_2 - L_1$ and $A_3 - L_2$.

5. The Sequence Of Two Installations Without Initial Deliverance Times

5.1. Introduction

The sequence operation in production flows appears in the usual practice for the installations waiting time decreasing when a lot of pieces use the same technology line in the same direction.

Let two installations U_1 and U_2 who process n pieces $P_1,\ldots,P_n$ ($n\ge 2$) in the same order (first U_1 and after U_2). We shall consider that U_1 and U_2 are available from the process beginning and the waiting time to come in execution for U_2 does not implies other prices. In addition we shall suppose that the pieces do not have a finish ending date.

Let note with t_{ij} the processing time of the j-th piece on the i-th installation.

The problem consists in a determination of the pieces execution beginning order such that the waiting time of the installation U2 to be minimum.

Let the matrix $T=(t_{ij})\in M_{2n}(\mathbf{R})$ of the time processing. The classical algorithm of Johnson consists in the following steps:

Step 1 We choose the least element on the first row. This will give us the first piece who will come in execution.

Step 2 We cut the previous column and we choose the least element on the second row. This will give us the last piece who will come in execution.

Step 3 We cut the previous column and we go again at the first step. After this we will obtain the second piece who will come in execution, and after we go again at the second step and we find the penultimate piece and so on.

The algorithm will continue till we shall finish all the pieces.

5.2. The new method

In the proof of Johnson's algorithm it exists a little but essential error. The author extrapolates a transposition between two consecutive terms to all transpositions. This is the reason that, even if it claim to obtain the optimum, it is not true.

The following method will guide us to the optimum but with a little harder calculus.

Let therefore the table of time processing and a permutation $\sigma=\begin{pmatrix} 1 & 2 & ... & n \\ i_1 & i_2 & ... & i_n \end{pmatrix}\in S_n$ – the group of permutations of n elements and an order of pieces, indexed by σ: $P_{i_1}, P_{i_2}, ..., P_{i_n}$:

Piece/Installation	P_{i_1}	P_{i_2}	…	P_{i_k}	…	P_{i_n}
U_1	$d_{i_1 1}$	$d_{i_2 1}$	…	$d_{i_k 1}$	…	$d_{i_n 1}$
U_2	$d_{i_1 2}$	$d_{i_2 2}$	…	$d_{i_k 2}$	…	$d_{i_n 2}$

We define: $g_1,g_2,\dots,g_n \geq 0$ - the pauses before entrence in execution of pieces $P_{i_1}, P_{i_2}, \dots, P_{i_n}$ on the installation U_2. We have, obviously:

- $g_1 = d_{i_1 1}$ (*from the beginning of the process*)
- $g_2 = \max(d_{i_1 1} + d_{i_2 1} - d_{i_1 2} - g_1, 0)$
- $g_3 = \max(d_{i_1 1} + d_{i_2 1} + d_{i_3 1} - d_{i_1 2} - d_{i_2 2} - g_1 - g_2, 0)$

  …

- $g_k = \max(\sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2} - \sum_{p=1}^{k-1} g_p, 0)$

  …

- $g_n = \max(\sum_{p=1}^{n} d_{i_p 1} - \sum_{p=1}^{n-1} d_{i_p 2} - \sum_{p=1}^{n-1} g_p, 0)$

  If we note: $B_{i_1 \dots i_k} = \sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2}$ we have:

- $g_1 = B_{i_1}$
- $g_2 = \max(B_{i_1 i_2} - g_1, 0)$
- $g_3 = \max(B_{i_1 i_2 i_3} - g_1 - g_2, 0)$

  …

- $g_k = \max(B_{i_1 \dots i_k} - \sum_{p=1}^{k-1} g_p, 0)$

  …

- $g_n=\max(B_{i_1\ldots i_n}-\sum_{p=1}^{n-1}g_p\ ,0)$

The objective function is therefore: $z=\min_{\sigma\in S_n}(\sum_{k=1}^{n}g_k)$.

We have by iteration:

$$\sum_{p=1}^{k}g_p=g_k+\sum_{p=1}^{k-1}g_p=\max(B_{i_1\ldots i_k}-\sum_{p=1}^{k-1}g_p\ ,0)+\sum_{p=1}^{k-1}g_p=\max(B_{i_1\ldots i_k},\sum_{p=1}^{k-1}g_p).$$

But: $\sum_{p=1}^{n}g_p=\max(B_{i_1\ldots i_n},\sum_{p=1}^{n-1}g_p)=\max(B_{i_1\ldots i_n},\max(B_{i_1\ldots i_{n-1}},\sum_{p=1}^{n-2}g_p))=$ $\max(B_{i_1\ldots i_n},B_{i_1\ldots i_{n-1}},\sum_{p=1}^{n-2}g_p)=\ldots=\max(B_{i_1\ldots i_n},B_{i_1\ldots i_{n-1}},\ldots,B_{i_1})$ from where:

$$z=\min(\sum_{p=1}^{n}g_p)=\min_{\sigma\in S_n}(\max(B_{i_1\ldots i_n},B_{i_1\ldots i_{n-1}},\ldots,B_{i_1})).$$

We have $B_{i_1\ldots i_k}=\sum_{p=1}^{k}d_{i_p1}-\sum_{p=1}^{k-1}d_{i_p2}=B_{i_1\ldots i_{k-1}}+d_{i_k1}-d_{i_{k-1}2}$ and much generally:

$$B_{i_1\ldots i_k}=\sum_{p=1}^{k}d_{i_p1}-\sum_{p=1}^{k-1}d_{i_p2}=\sum_{p=1}^{s}d_{i_p1}-\sum_{p=1}^{s-1}d_{i_p2}+\sum_{p=s+1}^{k}d_{i_p1}-\sum_{p=s}^{k-1}d_{i_p2}=B_{i_1\ldots i_s}+\sum_{p=s+1}^{k}d_{i_p1}-\sum_{p=s}^{k-1}d_{i_p2}$$

For the permutation $\sigma=\begin{pmatrix}1 & 2 & \ldots & k & \ldots & s & \ldots & n\\ i_1 & i_2 & \ldots & i_k & \ldots & i_s & \ldots & i_n\end{pmatrix}\in S_n$ and $z=\max(B_{i_1\ldots i_n},B_{i_1\ldots i_{n-1}},\ldots,B_{i_1})$, let consider a transposition of σ: $\tau=\begin{pmatrix}1 & 2 & \ldots & k & \ldots & s & \ldots & n\\ i_1 & i_2 & \ldots & i_s & \ldots & i_k & \ldots & i_n\end{pmatrix}\in S_n$.

If we note with bar all the quantities concerning τ we have:

- $t\neq k,s\Rightarrow \overline{d}_{i_t1}=d_{i_t1}$ şi $\overline{d}_{i_t2}=d_{i_t2}$
- $t=k\Rightarrow \overline{d}_{i_k1}=d_{i_s1}$ şi $\overline{d}_{i_k2}=d_{i_s2}$

- $t=s \Rightarrow \overline{d}_{i_s 1} = d_{i_k 1}$ şi $\overline{d}_{i_s 2} = d_{i_k 2}$

from where:

- $1 \le t < k \Rightarrow \overline{B}_{i_1 \ldots i_t} = \sum_{p=1}^{t} \overline{d}_{i_p 1} - \sum_{p=1}^{t-1} \overline{d}_{i_p 2} = \sum_{p=1}^{t} d_{i_p 1} - \sum_{p=1}^{t-1} d_{i_p 2} = B_{i_1 \ldots i_t}$

- $t=k \Rightarrow \overline{B}_{i_1 \ldots i_k} = \sum_{p=1}^{k} \overline{d}_{i_p 1} - \sum_{p=1}^{k-1} \overline{d}_{i_p 2} = \sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2} + d_{i_s 1} - d_{i_k 1} = B_{i_1 \ldots i_k} +$ $d_{i_s 1} - d_{i_k 1}$

- $k<t<s \Rightarrow \overline{B}_{i_1 \ldots i_t} = \sum_{p=1}^{t} \overline{d}_{i_p 1} - \sum_{p=1}^{t-1} \overline{d}_{i_p 2} = \sum_{p=1}^{t} d_{i_p 1} - \sum_{p=1}^{t-1} d_{i_p 2} + d_{i_s 1} - d_{i_k 1}$ - $(d_{i_s 2} - d_{i_k 2})$= $B_{i_1 \ldots i_t} + d_{i_s 1} - d_{i_k 1}$-$(d_{i_s 2} - d_{i_k 2})$

- $t=s \Rightarrow \overline{B}_{i_1 \ldots i_s} = \sum_{p=1}^{s} \overline{d}_{i_p 1} - \sum_{p=1}^{s-1} \overline{d}_{i_p 2} = \sum_{p=1}^{s} d_{i_p 1} - \sum_{p=1}^{s-1} d_{i_p 2}$ -$(d_{i_s 2} - d_{i_k 2})$= $B_{i_1 \ldots i_s}$ - $(d_{i_s 2} - d_{i_k 2})$

- $t>s \Rightarrow \overline{B}_{i_1 \ldots i_t} = \sum_{p=1}^{t} \overline{d}_{i_p 1} - \sum_{p=1}^{t-1} \overline{d}_{i_p 2} = \sum_{p=1}^{t} d_{i_p 1} - \sum_{p=1}^{t-1} d_{i_p 2} = B_{i_1 \ldots i_t}$

Let note now: $\alpha_{sk} = d_{i_s 1} - d_{i_k 1}$ and $\beta_{sk} = d_{i_s 2} - d_{i_k 2}$ for s>k and $\alpha_{sk}=\beta_{sk}=0$ for s≤k. We have now:

$$\overline{B}_{i_1 \ldots i_t} = \begin{cases} B_{i_1 \ldots i_t} \text{ if } t < k \\ B_{i_1 \ldots i_t} + \alpha_{sk} \text{ if } t = k \\ B_{i_1 \ldots i_t} + \alpha_{sk} - \beta_{sk} \text{ if } k < t < s \\ B_{i_1 \ldots i_t} - \beta_{sk} \text{ if } t = s \\ B_{i_1 \ldots i_t} \text{ if } t > s \end{cases}$$

$\overline{z}$=max($\overline{B}_{i_1}$,..., $\overline{B}_{i_1 \ldots i_{n-1}}$, $\overline{B}_{i_1 \ldots i_n}$)=max($B_{i_1}$,..., $B_{i_1 \ldots i_{k-1}}$, $B_{i_1 \ldots i_k}$ +α_{sk}, $B_{i_1 \ldots i_{k+1}}$ +α_{sk}-β_{sk},..., $B_{i_1 \ldots i_{s-1}}$ +α_{sk}-β_{sk}, $B_{i_1 \ldots i_s}$ -β_{sk}, $B_{i_1 \ldots i_{s+1}}$,..., $B_{i_1 \ldots i_n}$).

We must determine the pair (k,s) of pieces which will be permuted such that, after the computing of $\overline{z}$ to obtain a value less then or equal z.

How this thing leads us at a great number of calculations, we shall act in this way:

For an arbitrary distribution of pieces, corresponding to a permutation $\sigma=\begin{pmatrix} 1 & 2 & ... & k & ... & s & ... & n \\ i_1 & i_2 & ... & i_k & ... & i_s & ... & i_n \end{pmatrix} \in S_n$, we shall determine those piece which permute with the first will lead to the minimization of z. Suppoes now that this thing is for the first piece.

Let therefore P_{i_s} - the searched piece, who will take the place of the first piece P_{i_1}. We have therefore:

$\bar{z}$ =max($\overline{B}_{i_1}$,..., $\overline{B}_{i_1...i_{n-1}}$, $\overline{B}_{i_1...i_n}$)=max(B_{i_1} +α_{s1}, $B_{i_1 i_2}$ +α_{s1}-β_{s1},..., $B_{i_1...i_{s-1}}$ +α_{s1}-β_{s1}, $B_{i_1...i_s}$ -β_{sk}, $B_{i_1...i_{s+1}}$,..., $B_{i_1...i_n}$).

We shall continue this process till we cannot diminish the value of z. In this moment, we shall find the permutation with the second piece and so on.

Let conclude: We build the table where on the rows we have the pieces: P_{i_1} ,..., P_{i_n} and on columns alone: P_{i_2} ,..., P_{i_n} .

					...		P_{i_s}		...		P_{i_n}	
					...	...	$d_{i_k 1}$	$d_{i_k 2}$	...	...	$d_{i_n 1}$	$d_{i_n 2}$
					...	...	α_{s1}	$-\beta_{s1}$			α_{n1}	$-\beta_{n1}$
P_{i_1}	B_{i_1}	$d_{i_1 1}$	$d_{i_1 2}$	$\overline{B}_{i_1} = d_{i_1 1}$	...	...	$a_{i_1 i_s} = \overline{B}_{i_1} + \alpha_{s1}$		...	...	$a_{i_1 i_n} = \overline{B}_{i_1} + \alpha_{n1}$	
	...			...	...	...	...		...	...	...	
P_{i_k}	B_{i_k}	$d_{i_k 1}$	$d_{i_k 2}$	$\overline{B}_{i_k} = \sum_{p=1}^{k} d_{i_p 1} - \sum_{p=1}^{k-1} d_{i_p 2}$	...	...	$a_{i_k i_s} = \overline{B}_{i_k} + \alpha_s$ $1 - \beta_{s1}$		...	...	$a_{i_k i_n} = \overline{B}_{i_k} + \alpha_{n1}$ $-\beta_{n1}$	
	...			...	...	...	...		...	...	...	
P_{i_s}	B_{i_s}	$d_{i_s 1}$	$d_{i_s 2}$	$\overline{B}_{i_s} = \sum_{p=1}^{s} d_{i_p 1} - \sum_{p=1}^{s-1} d_{i_p 2}$	...	...	$a_{i_s i_s} = \overline{B}_{i_s} - \beta_{s1}$		...	...	$a_{i_s i_n} = \overline{B}_{i_s} - \beta_{n1}$	
	...			...	...	...	...		...	...	...	
P_{i_n}	B_{i_n}	$d_{i_n 1}$	$d_{i_n 2}$	$\overline{B}_{i_n} = \sum_{p=1}^{n} d_{i_p 1} - \sum_{p=1}^{n-1} d_{i_p 2}$			$a_{i_n i_s} = \overline{B}_{i_n}$				$a_{i_n i_n} = \overline{B}_{i_n}$	
max					...	...	$\max_{t=1,n} a_{i_t i_s}$				$\max_{t=1,n} a_{i_t i_n}$	

Fig.1

We shall choose the piece P_{i_k} for which: $z=\min_{s=2,n} \max_{t=1,n} a_{i_t i_s}$.

The next table will contains the new order of pieces where P_{i_1} will change the place with P_{i_k} .

The process will continue till $z=\min_{s=2,n} \max_{t=1,n} a_{i_t i_s}$ becomes greater then those computed in the preceding table.

This thing suggests the fact that any piece cannot be on the first position without grow the total time. If the value of z remains constant, we can act like in the precedings steps for each pieces order.

In the next table, we shall act analogously, but on the column we shall get only $P_{i_3},..., P_{i_n}$ corresponding to the new permutation.

The process will continue till the last piece.

Example

Piece/Installation	**P_1**	**P_2**	**P_3**	**P_4**
U_1	15	6	8	9
U_2	19	3	13	7

Johnson's algorithm propose us:

Piece/Installation	**P_1**	**P_3**	**P_4**
U_1	15	8	9
U_2	19	13	7

Piece/Installation	**P_1**	**P_3**
U_1	15	8
U_2	19	13

with the final order: P_2,P_3,P_1,P_4, therefore the new table will be, in order of execution:

Piece/Installation	**P_2**	**P_3**	**P_1**	**P_4**
U_1	6	8	15	9
U_2	3	13	19	7

with times:

$B_2=6$

$B_3=6+8-3=11$

$B_1=6+8+15-3-13=13$

$B_4=6+8+15+9-3-13-19=3$

therefore $z=\max(B_2,B_3,B_1,B_4)=13$.

Our algorithm consists from the following tables:

Table 1

					P_2		P_3		P_4	
					6	3	8	13	9	7
					-9	16	-7	6	-6	12
P_1	B_1	**15**	**19**	$\overline{B}_1=15$	6		8		9	
P_2	B_2	6	3	$\overline{B}_2=2$	18		1		8	
P_3	B_3	8	13	$\overline{B}_3=7$	7		13		13	
P_4	B_4	9	7	$\overline{B}_4=3$	3		3		15	
max					**18**		**13**		**15**	

therefore the piece on the first position is P_3.

Table 2

					P_2		P_1		P_4	
					6	3	15	19	9	7
					-2	10	7	-6	1	6
P_3	B_3	**8**	**13**	$\overline{B}_3=8$	6		15		9	
P_2	B_2	6	3	$\overline{B}_2=1$	11		2		8	
P_1	B_1	15	19	$\overline{B}_1=13$	13		7		20	
P_4	B_4	9	7	$\overline{B}_4=3$	3		3		9	
max					**13**		**15**		**20**	

The alternative piece on first position can be P_2.

Table 3

					P_3		P_1		P_4	
					8	13	15	19	9	7
					2	10	9	16	3	4
P_2	B_2	**6**	**3**	$\overline{B}_2=6$	8		15		9	
P_3	B_3	8	13	$\overline{B}_3=11$	21		36		18	
P_1	B_1	15	19	$\overline{B}_1=13$	13		29		20	
P_4	B_4	9	7	$\overline{B}_4=3$	3		3		7	
max					**21**		**36**		**20**	

therefore the permutation process for the first position is closed.

We go back to the table 1 and continue with the piece on the second position.

Table 4

<table>
<tr><td colspan="5" rowspan="3"></td><td colspan="2">P_1</td><td colspan="2">P_4</td></tr>
<tr><td>15</td><td>19</td><td>9</td><td>7</td></tr>
<tr><td>9</td><td>-16</td><td>3</td><td>-4</td></tr>
<tr><td>P_3</td><td>B_3</td><td>8</td><td>13</td><td>$\overline{B}_3=8$</td><td colspan="2">8</td><td colspan="2">8</td></tr>
<tr><td>P_2</td><td>B_2</td><td>**6**</td><td>**3**</td><td>$\overline{B}_2=1$</td><td colspan="2">10</td><td colspan="2">4</td></tr>
<tr><td>P_1</td><td>B_1</td><td>15</td><td>19</td><td>$\overline{B}_1=13$</td><td colspan="2">-3</td><td colspan="2">12</td></tr>
<tr><td>P_4</td><td>B_4</td><td>9</td><td>7</td><td>$\overline{B}_4=3$</td><td colspan="2">3</td><td colspan="2">-1</td></tr>
<tr><td colspan="5">**max**</td><td colspan="2">**10**</td><td colspan="2">**12**</td></tr>
</table>

therefore the piece on second position is P_1.

Table 5

<table>
<tr><td colspan="5" rowspan="3"></td><td colspan="2">P_2</td><td colspan="2">P_4</td></tr>
<tr><td>6</td><td>3</td><td>9</td><td>7</td></tr>
<tr><td>-9</td><td>16</td><td>-6</td><td>12</td></tr>
<tr><td>P_3</td><td>B_3</td><td>8</td><td>13</td><td>$\overline{B}_3=8$</td><td colspan="2">8</td><td colspan="2">8</td></tr>
<tr><td>P_1</td><td>B_1</td><td>**15**</td><td>**19**</td><td>$\overline{B}_2=10$</td><td colspan="2">1</td><td colspan="2">4</td></tr>
<tr><td>P_2</td><td>B_2</td><td>6</td><td>3</td><td>$\overline{B}_1=-3$</td><td colspan="2">13</td><td colspan="2">3</td></tr>
<tr><td>P_4</td><td>B_4</td><td>9</td><td>7</td><td>$\overline{B}_4=3$</td><td colspan="2">3</td><td colspan="2">15</td></tr>
<tr><td colspan="5">**max**</td><td colspan="2">**13**</td><td colspan="2">**15**</td></tr>
</table>

From the table 5 we have that the step is closed.

For the piece on third position:

Table 6

					P_4	
					9	7
					3	-4
P_3	B_3	8	13	$\overline{B}_3=8$	8	
P_1	B_1	15	19	$\overline{B}_2=10$	10	
P_2	B_2	**6**	**3**	$\overline{B}_1=-3$	0	
P_4	B_4	9	7	$\overline{B}_4=3$	-1	
max					**10**	

The process is closed. The order will be: P_3,P_1,P_2,P_4 with total time: 10.

If we come again at the table 2 and continue with the piece on the second position we have:

Table 7

					P_1		P_4	
					15	19	9	7
					7	-6	1	6
P_2	B_2	6	3	$\overline{B}_2=6$	6		6	
P_3	B_3	**8**	**13**	$\overline{B}_3=11$	18		12	
P_1	B_1	15	19	$\overline{B}_1=13$	7		20	
P_4	B_4	9	7	$\overline{B}_4=3$	3		9	
max					**18**		**20**	

Because we obtain a value greater than 13 the process will closed also.

6. Refrences

Ioan C. A. (2008), *Mathematics I, Sinteze,* Galati, Romania.

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*, Bucharest: Tehnica.