

# AN ALGORITHM FOR THE APPROXIMATE SOLUTION OF A RESOURCE- ALLOCATION PROBLEM IN NETWORK PLANNING

1967

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196701.71903>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

**Abstract**

**Full Text**

UDC 51:330.115

**CYBERNETICS AND CONTROL THEORY**

**L. Ya. LEIFMAN**

**AN ALGORITHM FOR THE APPROXIMATE SOLUTION OF A RESOURCE-ALLOCATION PROBLEM IN NETWORK PLANNING**

*(Presented by Academician V. M. Glushkov, 5 IX 1966)*

1. For a network graph  $G(E; W)$  the problem posed in <sup>(1)</sup> is considered (the notation and terminology of <sup>(1)</sup> are used without explanation):

Find an integer schedule  $A = \{a(x, y)\}$  minimizing the functional

$$F(A) = \max_{e \in E^f} \left\{ \max_{x \prec e} \{a(x, e) + \tau(x, y)\} - t_e^{\text{dir}} \right\} \quad (1)$$

subject to the constraints:

$$a(x, y) \geq \max_{z \succ x} \{a(z, x) + \tau(z, x)\} \equiv \beta_1(x) \quad \text{for all } (x, y) \in W; \quad (2)$$

$$\max_{(x, y) \in W} \{a(x, y) + \tau(x, y)\} \equiv \max_{e \in E^f} \left\{ \max_{x \prec e} \{a(x, e) + \tau(x, e)\} \right\} \leq M; \quad (3)$$

$$\sum_{(x, y) \in W(j)} f_j^m(a(x, y)) \leq h_j^m \quad (1 \leq j \leq n; 0 < m \leq M). \quad (4)$$

This paper gives one of the algorithms for the approximate solution of this problem, which is a concrete realization of the general scheme <sup>(1)</sup>.

2. **Topological ordering of the list of jobs of the system** <sup>(2)</sup>. Denote by  $W_k$  the set of jobs  $(x, y) \in W$  such that the maximum number of jobs on a path from the input of the network graph to event  $y$  is equal to  $k$  (the  $k$ -th class of jobs);  $W_k \cap W_{k'} = \emptyset$  for  $k \neq k'$ . Denote

$$a_2(x, y) = \beta_2(y) - \tau(x, y),$$

where

$$\beta_2(y) = T^{\text{dir}} - \rho_1(y),$$

$$\rho_1(y) = \begin{cases} T^{\text{dir}} - t_y^{\text{dir}}, & \text{for } y \in E^f, \\ \max_{z \succ y} \{\rho_1(z) + \tau(y, z)\}, & \text{for } y \notin E^f. \end{cases}$$

Here  $\beta_2(y)$  is the latest admissible time of occurrence of event  $y$ , and  $a_2(x, y)$  is the latest admissible starting time of job  $(x, y)$ .

Let  $K$  be the maximum class number. Put

$$\tilde{\theta}^i = \min\{a_2(x, y) : (x, y) \in W_i\}$$

for  $1 \leq i \leq K$ . Number all distinct values  $t_e^{\text{dir}} > \tilde{\theta}^K$  in increasing order, denoting them by  $t^q$  ( $1 \leq q \leq Q$ ). Put

$$\tilde{\theta}^i = \max\{a_2(x, e) : e \in E^f, t_e^{\text{dir}} = t^q, x \prec e\} + 1$$

for  $i = K + q$ ,  $1 \leq q < Q$ , and

$$\tilde{\theta}^{K+Q} = t^Q = T^{\text{dir}}.$$

Now

$$\theta^0 = \tilde{\theta}^1, \quad \theta^i = \max\{\tilde{\theta}^i, \theta^{i-1}\} \quad (1 \leq i \leq K + Q).$$

Denote

$$W_k^i = \{(x, y) : (x, y) \in W_k, \theta^i \leq a_2(x, y) < \theta^{i+1}\}$$

—the subclass of jobs with index  $i$ . We have

$$W_k^i \cap W_{k'}^{i'} = \emptyset$$

if  $i \neq i'$  or  $k \neq k'$ .

The jobs  $(x, y) \in W$  are grouped by subclasses, and the subclasses are arranged in the list of jobs in increasing order of indices, and, for the same index, in increasing order of class numbers. Such an ordering is topological.

**3. Partitioning the list of jobs into groups for which the values  $a(x, y)$  are selected simultaneously.** Such groups are the subclasses  $W_k^i$ , since the right-hand side of each inequality (2) contains  $a(x, y)$  only for jobs of preceding subclasses, and when

for all preceding subclasses the values  $\alpha(x, y)$  have been chosen and fixed, this right-hand side becomes the constant number  $\beta_1(x)$ . The quantity  $\beta_1(x)$  has the meaning of the earliest possible time of occurrence of event  $x$ , and hence also of the start of work  $(x, y)$ . Thus the values  $\alpha(x, y)$  for the works of one subclass are topologically independent.

**4. Constructing a schedule for the next subclass.** The principal scheme of the algorithm is shown in Fig. 1. Here we shall consider the operation of the individual blocks.

Fig. 1

Figure 1: Fig. 1

The data are entered in the form of a list of works topologically ordered according to Section 2 (Table 1) and a calendar (Table 2), where for each resource  $P_j$  two rows are allocated:  $\eta_j^m$  and  $\tilde{\eta}_j^m$  ( $0 < m \leq M$ ), with  $\eta_j^m := h_j^m$ .

In block *A*, the schedule for the subclass under consideration is fixed and the next nonempty subclass  $W_k^i$  is selected (if there is none, i.e., the list of works has been exhausted, then  $\Omega := 0$ ). For all  $(x, y) \in W_k^i$ ,  $\beta_1(x)$  and  $\alpha(x, y) := \beta_1(x)$  are computed.

**Fig. 1**

In block *B* we check whether the length of the calendar is sufficient for such a schedule (if not, we stop constructing the schedule, set  $\Omega := 1$ , and print the schedule fixed for the previous subclasses). We compute

$$\tilde{\eta}_j^m = \eta_j^m - \sum_{(x,y) \in W_k^i} f_j^m(x, y)$$

for all  $j$  and  $m$ . For each resource  $P_j$ , we find the earliest time  $N_j$  at which its shortage occurs (i.e.,  $\tilde{\eta}_j^{N_j} < 0$ ) and determine  $N = \min N_j$ , the time of the first resource shortage in the network graph. If there are no shortages, then  $\Omega := 1$ .

In block *C*, we choose those works  $(x, y) \in W_k^i$  that end before the first resource shortage occurs and, fixing for them the previously selected values  $\alpha(x, y)$ , exclude them from consideration. Then we choose, as  $j^*$ , one of the indices  $j$  for which  $N_j = N$ , and check whether the resource  $P_{j^*}$  is sufficient to perform at least one work

$$(x, y) \in \overline{W}_{j^*} = \{(x, y) : (x, y) \in W_k^i, j^* \in J(x, y), N \in (\alpha(x, y), \alpha(x, y) + \tau(x, y))\}.$$

If it is sufficient, then  $\Omega := 0$ ; if not, then  $\Omega := 1$ .

In block *D*, we seek the earliest point in time  $t - 1$  such that, beginning from it, the availability of resource  $P_{j^*}$  will possibly be sufficient to perform at least one work

$$(x, y) \in \overline{W}_{kj^*} = \{(x, y) : (x, y) \in W_k^i, j^* \in J(x, y)\}$$

(i.e., before the point  $t - 1$ , resource  $P_{j^*}$  is certainly insufficient to perform any work from  $W_{kj^*}^i$ ). If there is no such point on the interval  $(0, M]$ , then  $\Omega := 0$ . If there is, then  $\alpha(x, y) := t - 1$  for all  $(x, y) \in \overline{W}_{kj^*}$  for which  $\alpha(x, y) < t$ , and  $\Omega := 1$ .

In block *E*, first of all, we shift to time  $N$  the start  $\alpha(x, y)$  of each work  $(x, y) \in \overline{W}_{j^*}$  whose requirement for resource  $P_{j^*}$  exceeds the availability of this resource

at the  $N$ -th unit of time. All such works are excluded from  $\overline{W}_{j^*}$ . If after this there are still works remaining in  $\overline{W}_{j^*}$ , then for them we determine the total requirement  $\sigma$  for resource  $P_{j^*}$  and compare it with the availability  $\eta_{j^*}^N$  of resource  $P_{j^*}$  at the  $N$ -th unit of time. If there is insufficient resource ( $\sigma > \eta_{j^*}^N$ ), then for each work  $(x, y) \in \overline{W}_{j^*}$  we compute the time reserve

$$\lambda_{xy} = \alpha_2(x, y) - N,$$

which it will have if its start is shifted to time  $N$ . This shift is carried out for the work with maximal  $\lambda_{xy}$ ,

simultaneously excluding it from  $W_{j^*}$ . We proceed in this way until  $\bar{\sigma} \leq \hat{\eta}_{j^*}^N$ . After the resource deficit  $P_{j^*}$  at the  $N$ -th time unit has been eliminated by the enumerated actions, we determine the set  $\overline{W}$  of jobs with positive slacks  $\tau'(x, y) = \alpha(x, y) - \beta_1(x)$  and pass to block  $F$ .

In block  $F$  we compute, for each job  $(x, y) \in \overline{W}$ , the time reserve  $\gamma(x, y) = \beta_2(x) - \alpha(x, y) - \tau(x, y)$ . Then we choose in  $\overline{W}$  a job  $(\bar{x}, \bar{y})$  with the minimum reserve and check whether its idle time can be reduced. If not, then we transfer  $(\bar{x}, \bar{y})$  from  $\overline{W}$  to  $\widehat{W}$ , temporarily excluding it from consideration. If, however, the idle time of the job  $(\bar{x}, \bar{y})$  can be reduced, then we reduce it, marking the instant  $t_W$  of the earliest change in the calendar in the process of executing block  $F$ . Moreover, if the idle time of the job  $(\bar{x}, \bar{y})$  is completely eliminated, then we finally exclude it from consideration in block  $F$ , removing it from  $\overline{W}$ . If the idle time is only reduced but remains positive, then we transfer the job  $(\bar{x}, \bar{y})$  from  $\overline{W}$  to  $\widehat{W}$ . Having considered the job  $(\bar{x}, \bar{y})$  in this way, we again choose in  $\overline{W}$  a job with the minimum reserve and perform on it the actions described above. We continue in this way until the set  $\overline{W}$  is exhausted, after which we return from  $\widehat{W}$  to  $\overline{W}$  those jobs that begin after the attained value  $t_W$ , and exclude all the remaining jobs from  $\widehat{W}$  from further consideration. If, as a result, the set  $\overline{W}$  is not empty, then we repeat for it once again the whole procedure of block  $F$ . Otherwise, we pass to block  $B$ .

Received  
1 IX 1966

## REFERENCES

1. L. Ya. Leifman, DAN, 175, No. 1 (1967).

*Note: Figure translations are in progress. See original paper for figures.*

*Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.*