LARGE-SCALE SCHEDULING PROBLEMS IN LONG-RANGE PLANNING

Igor Zimin

February 1975 WP-75-17

1. Introduction.

The statement of a calculation problem of a development program and its involvement in general planning for social-economic systems was discussed in [1]. Difficulties of solving the problem are due to its essential non-linearity (combinatorial character) and high dimension.

Attempts to solve similar problems by using dynamic programming techniques are well known to be inefficient and impractical.

Therefore we need to develop and apply other various approximate and heuristic methods to deal with these problems. In particular, the idea of utilizing indirect methods of optimal control theory seems very promising to us.

In this paper we consider a model in differential form, and also some approximate and heuristic techniques for solving optimization problems on the basis of the model.

2. The Problem Statement.

Given a list of operations (jobs, actions) P, the performance of which leads to achieving the system goals. Let these operations be numbered.

The state of the operation i at the given moment t we characterize by the number $z^i(t)$. We assume $z^i(t)$ is a portion of the completed section of the operation at the instant t. The action i is terminated if

$$z^i(t) = 1, \quad i = 1,...,N \quad . \tag{1}$$

where N is a number of jobs in the program. We designate the moment t when (1) is completed by t_f^j.

The initial state of the i-th operation could be assumed equal to zero:

$$z^i(0) = 0, \quad i = 1,\ldots,N \tag{2}$$

The mark or job number z^i increases during its performance. The rate or intensity of z^i we denote by u^i. So we have the relationship

$$\frac{dz^i}{dt}(t) = u^i(t), \quad i = 1,\ldots,N \tag{3}$$

The performance of jobs is usually subjected to constraints of two kinds.

Group (α) is a group of logical constraints. It includes constraints to the sequence in which some of the operations are performed. These constraints are performed by the prescribed partial ordering of operation performance. For instance, figure 1 shows the logical sequence of a certain program.

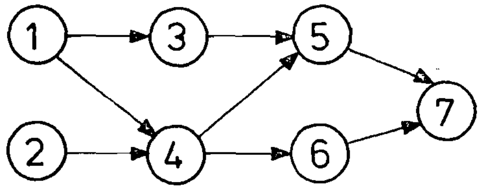


Figure 1

The representation in the figure means that operation 4 must be completed before the two independent operations 5 and 6 could start. We have assumed that operation numbers are placed within circles and predecessor relations among the activities are shown by arrows.

Group (β) includes resource constraints and some others. For instance, various conditions may be imposed on maximum and minimum intensity of the job performance.

The (β) constraints we express in the following way:

$$\sum_{j=i}^{n} r_i^j(t)u^j(t) \leq R^j(t) \quad , \quad j = 1,2,...M \tag{4}$$

where R^j is the inflow intensity of the type j resource, and r_i^j is the intensity with which the type j resources are consumed while performing the i-th job with unit intensity. The R^i and r_j^i we assume to be given for each instant t.

$$0 \leq u^i(t) \leq h^i(t) \quad , \quad i = 1,...,N \tag{5}$$

where $h^i(t)$ is maximum feasible intensity of the job performance at the instant t.

In the model we consider $z(t) = (z^1(t),...,z^n(t)$ as a phase vector and the vector $u(t) = (u^1(t),...,u^n(t))$ as a control. We define the best control, or the best schedule u^* as the one in which a certain objective function I(u) is minimized.

Remarks

Note that the model presented is quite general and overlaps a wide range of scheduling problems. Below we consider some of them. It is also a dynamic model because the program development is considered over time and space, all operations may change their intensities while being performed, and the inflow intensity is an arbitary function of time.

3. Some Examples of Dynamic Scheduling Problems

Problem A:

$$\text{Min } T$$

subject to

$$(2),\ (3),$$

$$(\alpha),\ (\beta)$$

$$z(T) \geqq e, \tag{6}$$

Where e ≡ N-dimensional vector with all components equal to the unit (e=(1,1,...,1)).

The problem is to perform the program for a minimum time. In that case I(u) = T. T is the time of completing the program. We shall say the program is completed if all its operations are completed.

Problem B

$$\text{Min } ||z(T) - Z_f|| \tag{7}$$

s.t.

$$(2),\ (3),$$

$$(\alpha),\ (\beta),$$

where

T ≡ given period of time (the period of planning);

||x-y|| ≡ the distance between two points x and y in N-dimensional space for some given metric.

For instance, the preference of some program states may be given in form

$$\sum_{j=1}^{N} \lambda_j (1 - z^j(T))^2 \quad . \tag{8}$$

where

$\lambda_j \equiv$ relative "weight" of j-th job in the program.

Problem C:

$$\text{Min } c(u, z)$$

s.t. (2), (3)

(α), (β),

(6).

The problem is to determine the time T of completing the program and the schedule which minimize capital (direct and/or indirect) costs of program performance c(u, z).

Problem D:

$$\text{Min} \max_{j \varepsilon P} (t_f^j - t_D^j)+$$

s.t. (2), (3)

(α), (β)

(6).

The problem is to minimize maximum deviation between the time t_f^j (when the corresponding job is completed) and the due time which is designated by t_D^j. Where the function (x) is defined as follows

$$(x)_+ = \begin{cases} 0, & x \leq 0 \\ x, & X > 0 \end{cases} \quad .$$

Problem E:

$$\text{Min } ||u - u^*||$$

s.t. (2), (3)

(α), (β).

(6).

The problem is to minimize the deviation between a given plan u^* and actual modified control, which may be realized under actual available resources. For instance, the function may be defined in form.

$$\int_0^T \sum_{j=1}^{N} \mu_j (u^j(t) - u^{*j}(t))^2 dt \quad .$$

Where $\mu_j \geq 0$ are relative "penalties" for the deviation.

Problem F:

$$\text{Min } ||F^* - F(u)||$$

s.t. (2), (3), (α), (β), (6).

where

$F^*(t)$ ≡ certain given (desirable) resource consumption by the program;

$F(u(t))$ ≡ resource consumption under given constraints.

For example the objective function may be assumed as

$$\int_0^T \sum_{i=1}^{M} (f^i(t) - \sum_{j=1}^{N} r_j^i(t) u^j(t))^2 dt \quad .$$

In a similar way many other problems may be stated and also all objective functions described may be combined.

To simplify the discussion we shall consider the solution of Problem B. It should be emphasized that some of the problems A-F are interconnected, in the sense that the solution of a problem may be obtained on this basis of the solution of another problem. For instance, instead of the time minimization problem, a set of problems B with fixed time $T < T^*$ may be solved, where T^* is the optimal time of carrying out the program (i.e. an optimal solution of the Problem).

If $T > T^*$, there is an infinite set of ways to carry out the program and problem A degenerates. But this difficulty is easily overcome by the introduction of a fictitious *) job, which can start when all the final actions of the program have terminated. The intensity of this job should be assumed to be constant and equal to 1/T.

The optimal time for performing this "lengthened" program will be greater than T^* and also greater than T. (T is chosen in advance. Now, instead of the minimization problem for the objective function (8) for the initial program, the minimization problem can be solved for a similar objective function for the "lengthened" program, that is for extended vector $\hat{z} = (z, z_{N+1})$.

If the fictitious job has not started its performance at a given T, then $T < T^*$. On the other hand, if the number z^{N+1} of the fictitious job becomes non-zero, then $T > T^*$. Thus during computation, a dual upper and lower estimate is obtained for the optimal time of performing the program.

---

*) We define a fictitious job as one which has non zero duration and does not consume resource.

4. Solution of Problem B

Let us consider the solution of the following problem:

$$\text{Min } I(u) \equiv \frac{1}{2} \sum_{j=1}^{N} \lambda_j (1 - z^j(T))^2 \tag{9}$$

s.t. (2), (3)

(α), (β), (6) .

Note that most of the computational methods for obtaining optimal solution can be treated as a utilization of penalties. In some cases, there is a "feed-back" between the deviation from the optimal solution and sixe of the penalty. This feed-back is realized by means of the solution of a dual problem. Here we use penalties of a discontinuous kind for violation of the (α) constraints. Instead of system (3) with logical constraints of (α) type, introduce the system (modified system)

$$\frac{dz^j}{dt} = u^j \prod_{k \varepsilon \Gamma_j^-} \theta_+(z^k - 1)\theta_-(1 - z^j) \tag{10}$$

with the (α) constraints and the condition (6) deleted.

Now every job can be performed (the corresponding u^j may be positive) until the previous operations have been completed, or after z^j has reached its final value 1. But the mark (job number) z^j will not increase under these conditions. The intensity of performing an operation which has terminated or is inadmissible under the (α) conditions can take values in an interval $[0, h(t)]$ $(t \; \varepsilon \; [0, T])$. To avoid this lack of uniqueness, at such instants we shall choose $u^j = 0$ from the set of permissible values.

Under these conditions the problem B is equivalent to the modified problem:

$$\text{Min } \frac{1}{2} \sum_{j=1}^{N} \lambda_j (1 - z^j(T))^2$$

$$\text{s.t.} \qquad \frac{dz^j}{dt} = u^j \, \theta_-(1 - z^j) \prod_{k \in \Gamma_j^-} \theta_+ \, (z^k - 1), \quad z^j(0) = 0 \tag{11}$$

$$(\beta) \qquad \sum_{j=1}^{N} r_i^j(t) \; u^j(t) \leq R^j(t)$$

$$0 \leq u^i(t) \leq h^j(t)$$

The difference between this problem and ordinary control theory problems is due to discontinuous multipliers in the right-hand sides of the equations (10). Nevertheless the maximum principle is valid in this case. The necessity of maximum principle conditions for more general problems with discontinuous right-hand sides of equations has been proved by by V.V. Velitchenko in [2]. Moreover the maximum principle conditions are (locally) sufficient for this problem. The proof can be found in [3].

These conditions can be written as follows. Let the control (schedule) $u^*(t)$ maximize, on the phase trajectory defined by it, the hamiltonian function

$$H(u, z, p) = \sum_{j=1}^{N} u^j(t) p^j(t) \theta_+(1 - z^j(t)) \prod_{k \in \Gamma^-} \theta_-(z^k - 1) \tag{12}$$

with respect to all admissible (β) constraints on the controls. Where

$p(t) \equiv$ corresponding to $u^*(t)$ vector of dual (or conjugate) variables (Lagrange multipliers) $p(t)$ is a solution of the conjugate system:

$$\frac{dp_j}{dt} = 0 \quad , \tag{13}$$

with jump conditions for the instants coinciding with the instants at which the operations terminate.

$$p_j(t_f^j - 0) - p_j(t_f^j + 0) =$$

$$\frac{1}{u^j(t_f^j - 0)} \sum_{\ell \varepsilon \Gamma^+} p_\ell(t_f^j + 0)\; u^\ell(t_f^j + 0) \prod_{k \varepsilon \Gamma_\ell^-} \theta_-(t_f^j - t_f^k) \tag{14}$$

and boundary conditions

$$p_j(T) = \lambda_j(1 - z^j(T)) \quad , \tag{15}$$

$j = 1,2,\ldots,N$

It is clear that all p_j are piecewise constant functions.

The aim of the method is to find controls $u(t)$ and corresponding $p(t)$, which satisfy conditions (13) - (15) and maximize function $H(u, z, p)$. Similar methods are usually called indirect optimal control methods [4].

The following algorithm, based on the method of succesive approximations, will be used for solving the problem.

(i) Given any admissible control $u^{(1)}(t)$, $t \; \varepsilon \; [0, T]$. We may always use $u^{j(1)}(t) = 0$, $j=1,\ldots N$, $t \; \varepsilon \; [0, T]$. For a given $u^{(1)}$ system (10) is integrated from $t = 0$ to $t = T$. Simultaneously we determine the value $t_f^{j(1)}$.

We denote this trajectory by $z^{(1)}(t)$.

(ii) Substitute $u^{(1)}$, $z^{(1)}$, $t_f^{(1)}$ into the system (13) - (14) and integrate it from t = T to t = 0 for the given "initial" conditions (15).

(iii) Determine a new approximation of control $u^{(k+1)}$ using the condition

$$H(u(t)^{(k+1)}, z^{(k)}(t), p^{(k)}(t)) = \text{Max } H(u, z^{(k)}, p^{(k)}) \quad .$$

where the maximum is taken under (β) constraints.

(iv) Compare $I^{(k)}$ and $I^{(k+1)}$ **)

If

$$I^{(k)} < I^{(k+1)}$$

then replace $u^{(1)}$ by $u^{(k+1)}$ and pass to (i).

If

$$I^{(k)} \geq I^{(k+1)}$$

(v) Calculate new control as follows

$$\tilde{u}(t) = u^{(k)}(t) + \rho(u^{(k+1)}(t) - u^{(k)}(t)) \quad ,$$

$$0 \leq \rho \leq 1 \quad . \qquad (16)$$

($\tilde{u}(t)$ belongs to admissible control domain, because this domain is a convex set). Pass to (vi).

**) We denote $I^{(k)}$ as the value of the objective function I(u), when $u = u^{(k)}$. $I^{(1)}$ is assumed to be equal $+\infty$.

(vi) Calculate $I(\tilde{u})$ and compare with $I^{(k)}$.

If

$$I(\tilde{u}) < I^{(k)}$$

then set $u^{(k+1)} = \tilde{u}$, $\rho = 1$ and pass to (i).

Otherwise, if

$$I(\tilde{u}) \geq I^{(k)}$$

then reduce ρ (for instance, one may set $\rho = \rho/2$) and pass to (vii).

(vii) Compare ρ and ε (ε is the external parameter, small positive number). If

$$\rho < \varepsilon$$

we assume the iterative process to be finished and consider $u^{(k)}$ to be the solution of the problem.

Otherwise, if

$$\rho \geq \varepsilon$$

pass to (z).

The proof of algorithm convergency is similar to the one in [5].

The specific character of the problem and its solution by the method should be outlined.

When integrating the system (10) (i) the following linear programming problem (LPP) is to be solved at every time-step:

$$\text{Max } H(u, z^{(k)}, p^{(k)})$$

s.t. (17)

(β) constraints

The coefficient attached to u^j in the Hamiltonian function is zero for those jobs which do not satisfy (α) constraints or have terminated. Hence the corresponding u^j may be made equal to zero without changing the values of the Hamiltonian. Thus, the maximum can be sought only with respect to the u^j for which the jobs have not been performed and which are admissable by the network logic. This essentially reduces the dimension of the LPP. In the problem the number of variables is equal to the number of logically admissable operations at the instance.

The number of linear constraints at each time-step is equal to a number of different resources, which are consumed by these logically admissable jobs.

The choice of the time step length could be easily automated in the algorithm. Indeed one need not solve LP problem (17) at every time step, but only at the instants, when one of the following events takes place:

(i) resource inflows have changed,

(ii) resource consumption (functions $r_j^i(t)$, $g_j^i(t)$ have changed,

(iii) one (or more jobs) has been completed.

The difference between the time when one of these events occurs and the current instant determines the length of the next time step.

The positive features of the method are as follows:

(i) the usage of the standard procedures (for example, simplex algorithm);

(ii) the simplicity of the computer program;

(iii) a relative small number of computations at every iteration;

(iv) "high speed" work of the algorithm (as a consequence of (i)-(iii), due to the fact that scheduling problems and LP problems are not to be solved with high precision);

(v) the approximate solution obtained at every intermediate iteration always belongs to the feasible control set;

(vi) the algorithm can easily be extended to incorporate nonlinear relationships between resource consumption and the performance intensity of a job.

Some generalizations of the model are discussed below.

The shortcomings of the algorithm are:

(i) in general the algorithm enables the obtainment of a solution which corresponds to a local minimum of the objective function;

(ii) noneconomical usage of computer memory (the algorithm is expected to store program trajectories obtained at the two adjacent iterations).

These shortcomings can be easily removed. The first one will be discussed in section 5 . The second one may be removed by nonessential sophistication of the computer program.

5. Additional Constraints on Program Performance

1. In previous sections we described the algorithm which guarantees obtaining the local optimal solution of the scheduling problem. Note that the problem is a multiextremal one by its nature.

Let us modify the initial problem by introducing the objective function

$$\tilde{I}(u) = I(u) + \varepsilon \|u\|^2 = I + \varepsilon \int_0^T \sum_{j=1}^{N} (u^j(t))^2 \, dt \qquad (18)$$

instead of $I(u)$.

Where $\varepsilon > 0$ is a sufficiently small number.*
In this case the Hamiltonian

$$H(u) = \sum_{j=1}^{N} (c_j u^j - \varepsilon (u^j)^2)$$

is a strictly concave function.

---

*It is easy to verify that ε is subjected to the following constraint

$$\varepsilon \leq \min_{\substack{1 \leq j \leq N \\ t \in [0,T]}} \left\{ \frac{c_j(t)}{h^j(t)}, \min_{1 \leq i \leq M} \frac{1}{R_i(t)} \sum_{j=1}^{N} r_j^i(t) c_j(t) \right\} .$$

Where

$c_j(t) \equiv$ the coefficient at u^j in the Hamiltonian (12),

and

$c_j(t) > 0$, $h^j(t) > 0$, $R^i(t) > 0$, $r_j^i(t) > 0$.

Thus the problem

$$H(u) \to \text{Max}$$

s.t. (β)

has a single (global) solution. Consequently the initial modified problem has a single solution.

To solve the modified problem we apply the same algorithm. However, now one needs to solve a nonlinear (quadratic) programming problem at every time step. The dimension of the problem is the same as in the linear case (see (12)).

2. Storable Resources. In section 2 we considered the case when a program consumes only unstorable resources. The problem may be generalized by including constraints to storable resources. As usual, a resource is called storable if the residue of it can be utilized at subsequent instants.

The (β) constraints on the storable resources can be written as

$$\sum_{j=1}^{N} \int_0^t q_j^k(\tau)\, u^j(\tau)\, d\tau \leq \int_0^t Q^k(\tau)\, d\tau \quad , \qquad k = 1,2,\ldots,N_2 \tag{19}$$

where

$g_j^k(t) \equiv$ the intensity with which the type k storable resource is consumed at the instance t when performing the j-th operation with unit intensity;

$Q^i(t) \equiv$ the intensity of inflow of the type i storable resource, released for performance of the program at the instant t;

$N_2 \equiv$ the number of different storable resources.

Let us extend the phase vector by introducing additional phase variables $z^{N+i}(i = 1,2,\ldots,N_2)$. The equations for these variables are written in the following form:

$$\frac{dz^{N+i}}{dt} = \sum_{j=1}^{N_2} q_j^i \, u^j \quad , \qquad z^{N+i}(0) = 0 \quad , \tag{20}$$

$$i = 1,2,\ldots,N_2 \quad .$$

Denote functions $F^i(t)$ as

$$F^i(t) = \int_0^t Q^i(\tau) \, d\tau \tag{21}$$

Then, in accordance with (20), (21) the constraints (19) may be written as

$$z^{N+i}(t) \leq F^i(t) \quad , \qquad i = 1,2,\ldots,N_2 \quad . \tag{22}$$

Thus we get the control problem with phase constraints.

Consider one simple approach for its solving. Again, we modify equations (10) by introducing additional discontinuous terms to its right-hand sides. Instead of equations (10) and constraints (22), consider the system

$$\frac{dz^j}{dt} = u^j\left(\theta_-(1 - z^j) \prod_{\ell \in \Gamma_j^-} \theta_+(z^\ell - 1) - \sum_{i=1}^{N_2} Y_{ij} \, \theta_-(z^{N+i}(t) - F^i(t))\right) \quad ,$$

$$z^j(0) = 0 \quad , \tag{23}$$

$$j = 1,2,\ldots,N \quad .$$

Where

$$y_{ij} = \begin{cases} 0 \ , & \text{if } q_j^i \equiv 0 \\ 1 \ , & \text{if } q_j^i \not\equiv 0 \end{cases}$$

Similar to 3 it may be shown that maximum principle conditions are necessary and sufficient for control optimality in the problem B, when phase equations are (20) and (23). In this case the Hamiltonian is as follows

$$H(u,z,p) = \sum_{j=1}^{N} \Bigg(p_j \Bigg(\theta_-(1 - z^j) \prod_{\ell \varepsilon \Gamma_j^-} \theta_+(z - 1) - \sum_{i=1}^{N} y_{ij} \theta(z^{N+i} - F^i) \Bigg) \Bigg) + \sum_{i=1}^{N_2} q_j^i(t) \ \sigma_i(t) \Bigg) u^j \ .$$

Where

$p_j \equiv$ conjugate to z^j variable, which satisfies the conditions (13) - (15);

$\sigma_i \equiv$ conjugate to z^{N+i} variable, which satisfies equations

$$\frac{d\sigma_i}{dt} = 0 \ ,$$

boundary conditions

$$\sigma_i(T) = 0$$

and jumps conditions

$$\sigma_i(t_H^i - 0) - \sigma_i(t_H^i + 0) = \frac{\sum_{j=1}^{N} y_{ij} \ p_j(t_H^j + 0) \ u^j(t_H^j + 0)}{\sum_{j=1}^{N} q_j^i(t_H - 0) \ u^j(t_H^i - 0)} \ . \qquad (24)$$

where we denote t^j_H as the moment when

$$z^{N+1}(t) = F^i(t)$$

and

$$\frac{dz^{N+i}}{dt} > 0 \quad ,$$

i.e. the moment when the phase trajectory intersects outward with the surface $F(t) = (F_1(t), \ldots, F_{N2}(t))$. Whenever this occurs, conjugate variable σ_i is subjected to jumps (24).

The algorithm does not change, but one should take into account the modifications mentioned above.

Similar to p_j the conjugate variables σ_i are piecewise constant over time. It allows us to use the computer memory economically, because to construct the conjugate trajectory we need to know the values of the jumps and the corresponding instants only.

3. Constraints on Minimal Intensity of Job Performance

Consider the case where constraints are imposed upon the minimal performance intensity for all or some jobs of the program. In particular, one of these constraints is that the job is to be carried out without interruptions (for example, technological processes in the chemical industry cannot be interrupted).

Constraints of this kind can be taken into account in the model in the following way

$$u^j(t) \geq s^j(t) \; \theta_-(z^j(t)) \; \theta_-(1 - z^j(t)) \qquad (25)$$

where

$s^j(t) \equiv$ minimal admissible intensity of carrying out the job j at the instant t.

This means that if the job performance has begun and is not completed ($0 < z^j(t) < 1$) its intensity should be no less than $s^j(t)$. If the job has not begun ($z^j(t) = 0$) or has been completed, formula (25) reduces to

$$u^j(t) \geq 0 \quad ,$$

i.e. the job may remain in one of these states for an indefinite time.

Multiplying both sides of (25) by $\theta_-(1 - z^j) \prod_{\ell \varepsilon \Gamma_j^-} \theta_{\pm}(z^\ell - 1)$ and integrating them from $\tau = 0$ up to $\tau = t$ we get

$$z^j(t) \geq \int_0^t s^j(t)\ \theta_-\ (z^j)\ \theta_-(1 - z^j) \prod_{\ell \varepsilon \Gamma_j^-} \theta(z^\ell - 1)dt \qquad (26)$$

Here we used equation (10).

It is convenient to introduce auxiliary phase variables z^{N+j}, which satisfy the system

$$\frac{dz^{N+j}}{dt} = s^j(t)\ \theta_-(z^j)\ \theta_-(1 - z^j) \prod_{\ell \varepsilon \Gamma_j^-} \theta(z^\ell - 1) \qquad (27)$$

$$z^{N+j}(0) = 0 \quad ,$$

$$j = 1,2,\ldots,N \quad .$$

Note that constraints (26) are equivalent to

$$z^j(t) - z^{N+j} \geq 0 \qquad (28)$$

Instead of (10) and (28) we consider the system of equations

$$\frac{dz^j}{dt} = u^j \; \theta_-(1 - z^j) \; \theta_-(z^j - z^{N+j}) \prod_{\ell \varepsilon \Gamma_j^-} \theta_+(z^\ell - 1) \qquad (29)$$

The maximum principle conditions for this (modified) problem are as follows

$$\text{Max } H(u, z^*(t), p^*(t)) = H(u^*(t), z^*(t), p^*(t)) \qquad (30)$$

where

$$H(u, z, p) = \sum_{j=1}^{N} u^j p_j \; \theta_-(1 - z^j) \; \theta_-(z^j - z^{N+j}) \prod_{\ell \varepsilon \Gamma_j^-} \theta_+(z \quad - 1)$$

$p_j(t) \equiv$ Lagrange multipliers $(j=1,\ldots,N)$

which satisfy equations (13), boundary conditions (15) and jump conditions

$$p_j(t_f^j - 0) - p_j(t_f^j + 0) = \frac{1}{u^j(t_f^j - 0)} \Big(\sum_{i \varepsilon \Gamma_j^+} p_i(t_f^j + 0) u^i(t_f^j + 0) - \sigma_j(t_f^j + 0) s^j(t_f^j + 0) \quad , \qquad (31)$$

$$p_j(t_q^j - 0) - p_j(t_q^j + 0) = - \frac{1}{u^j(t_q^j - 0)} p_j(t_q^j + 0) u^j(t_q^j + 0) \quad , \qquad (32)$$

$$p_j(t_0^j - 0) - p_j(t_0^j + 0) = \frac{1}{u^j(t_0^j - 0)} \sigma_j(t_0^j + 0) \; ; \qquad (33)$$

$t_q^j \equiv$ the first moment when the trajectory z^j achieves the boundary of the domain

$z^j \leq z^{N+j}(t) \quad ;$

$\sigma_j \equiv$ conjugate to z^j $(j=N+1,\ldots,2N)$ variables, which satisfy the following equations

$$\frac{d\sigma_j}{dt} = 0 \quad ,$$

boundary condition

$$\sigma_j(T) = 0$$

and jump conditions

$$\sigma_j(t_q^j - 0) - \sigma(t_q^j + 0) = \frac{1}{s^j(t_q^j - 0)} \; p_j(t_q^j + 0) u^j(t_q^j + 0) \qquad (34)$$

We have denoted the values corresponding to optimal control with a star.

The variables σ can be treated as "indirect" penalties for violation of conditions (28). We need not change the algorithm to solve the modified problem. Additional information includs information of auxiliary variable trajectories, of the instants t_q^j and of the jumps (31) - (34). The dimension of the LP problem, which is to be solved at each time step, does not increase.

4. <u>Constraints on Simultaneous Performance of Jobs</u>

If some operations should be performed simultaneously and cannot be shared, one may consider them as one operation with an extended vector of resource consumption. The elements of the vector are intensities of resource consumption for all jobs which are combined. Note that components corresponding to the same resource type should be added.

Conversely, some jobs may be subjected to the restriction of sharing their performances over time; for instance, job j cannot be performed simultaneously with job k.

These restrictions are also taken into account by introducing appropriate discontinuous multipliers into the right-hand sides of the equations. In our case the modified equations are written in the form

$$\frac{dz^j}{dt} = u^j(\theta_-(1 - z^j) \prod_{\ell \varepsilon \Gamma_j^-} \theta_+(z^\ell - 1) - \theta_-(z^k)\ \theta_-(1 - z^k))$$

$$\frac{dz^k}{dt} = u^k(\theta_-(1 - z^k) \prod_{m \varepsilon \Gamma_k^-} \theta+(z^m - 1) - \theta_-(z^j)\ \theta_-(1 - z^j))$$

The maximum principle holds and the algorithm does not change. In a similar manner many other restrictions could be combined with the observed restrictions and included in the model.

5. Other Approaches to Solving the Problem

Note that the method used to deal with (α) constraints in previous sections is not the only feasible one. In this section we briefly discuss some other techniques for solving the problem which are generalizations and complements of the method under discussion.

The first group of methods introduces penalties (not necessarily of the discontinuous type) on the intensity of performing an operation.

(α) constraints could be written in the form:

$$\begin{aligned} & z^j(t)\ (1 - z^\ell(t)) = 0 \ , && \ell \varepsilon \Gamma_j^- \ , \\ & u^j(t)\ (1 - z^\ell(t)) = 0 \ , && \ell \varepsilon \Gamma_j^- \ , \\ & (1 - z^j)\ z^k = 0 \ , && k \varepsilon \Gamma_j^+ \ , \\ & u^j\ z^k = 0 \ , && k \varepsilon \Gamma_j^+ \ , \end{aligned} \tag{35}$$

This means that the j-th job cannot be performed until all immediately preceeding jobs have been completed, and its performance breaks off after the immediately consecutive jobs are in operation.

Instead of the initial equations (3) and conditions (35), consider

$$\frac{dz^j}{dt} = f^j(z,\ u,\ \mu)u^j \tag{36}$$

$$j = 1,2,\ldots,N \ ,$$

where

$f_j \equiv$ function of phase coordinates, controls and the vector of parameters μ with the following properties.

(i) $f_j \leq 1$;

(ii) if (α) constraints are satisfied, then

$f_j = 1$;

(iii) if (α) constraints are infringed then

$f_j \rightarrow f_j^0 \leq 0$,

as parameters tend to certain limits.

If we consider the problem A we obtain

$$T^*(\mu) \leq T^* \quad .$$

where

$T^*(\mu) \equiv$ minimum time in the modified problem (36) under (β) constraints;

$T^* \equiv$ minimum time in the initial problem under (α) and (β) constraints

Indeed, the set of all solutions of system (36) subjected to (β) constraints includes all solutions of system (3) subjected to (α) and (β) constraints. The penalty functions may be constructed in such a way that

$$T^*(\mu) \rightarrow T^*$$

with a certain variation of the parameters μ.

For example, take the penalty functions

$$f_j(z, \mu) = (1 - \sum_{\ell \varepsilon \Gamma_j^-} \mu_\ell^j (1 - z^\ell) - \sum_{k \varepsilon \Gamma_j^+} \mu_k^j z^k)^{\mu O} \quad ,$$

or

$$f_j(z, \mu) = 1 - \sum_{\ell \varepsilon \Gamma_j^-} \mu_\ell^j \, \theta_-(1 - z^\ell) - \sum_{k \varepsilon \Gamma_j^+} \mu_k^j \, \theta_-(z^k)$$

where $\mu_j^i (i, j = 1,\ldots,N)$ are large positive numbers and μO is and odd positive integer. Violation of (α) constraints will lead to reduction in the job number z^j.

Another group of methods uses a penalty for infringing (α) constraints, introduced into the objective function. Instead of the initial objective function I, a "penalized" function of one of the following typical kinds is minimized:

$$I + \int_0^T \sum_{j=1}^N z^j \sum_{\ell \varepsilon \Gamma_j^-} \mu_\ell^j (1 - z^\ell) dt \quad ,$$

$$I + \int_0^T \sum_{j=1}^N (u^j)^2 \sum_{\ell \varepsilon \Gamma_j^-} \mu_i^j (1 - z^\ell)^2 \, \theta_-(1 - z^\ell) dt \, +$$

$$\int_0^T \sum_{j=1}^N (u^j)^2 \sum_{k \varepsilon \Gamma_j^+} \mu_k^j (z^k)^2 \, \theta_-(z^k) dt$$

$$I + \int_0^T \sum_{j=1}^N u^j \sum_{\ell \varepsilon \Gamma_j^-} \mu_i^j \, \theta_-(1 - z^\ell) dt \, +$$

$$\int_0^T \sum_{j=1}^N u^j \sum_{k \varepsilon \Gamma_j^+} \mu_k^j \, \theta_-(z^k) dt \quad . \tag{37}$$

The utilization of smooth penalties allows us to apply direct optimization methods to solve the problem [4] and rationally combine them with the indirect methods described above.

6. Some Heuristic Approaches to Solve the Problem

It should be emphasized that the conjugate variables in our problem could be termed the objectively stipulated estimate of the operation (or "shadow price" of the job). The intensity u^j of performing a job depends on the value of the coefficient in the hamiltonian. In the case of uncompleted jobs satisfying (α) constraints, it is equal to P_j and characterizes the "weight" or importance of performing the job at a given instant. The weight of the job vanishes at $t = T$, if the job terminates at this instant. If the job is not completed (terminated) its weight is non-zero and equal to $\lambda_j(1 - z^j(T))$. (15)

The zero value of p_j at the instant of terminating the job ($t = t_f^j$) is increased by a jump (14). This increase is larger, the greater the weight of the jobs immediately following the j-th, and the greater the intensity of performing these jobs (at previous iteration of the algorithm). It is smaller, the less intensively the j-th action was performed at the instant t_f^j. We may treat it as the i-th job immediately following the j-th makes a claim for an increase in the intensity of performance of its immediate predecessors, by increasing their weight at the next iteration of the algorithm.

Notice that the job i, immediately following job j, increases the weight of job j only if it is started immediately after action j, i.e. $u^j(t_f^j + 0) \neq 0$, and if all the other preceding jobs i are completed at this instant ($z^k(t) = 1$, $k \varepsilon \Gamma i$).

Though the weight of each job is increased at the expense of the job immediately following it, the increase becomes more marked as the time lag of the following actions becomes greater. For the weight of a job is increased to a greater degree as weights of the immediately subsequent jobs are increased. In turn, the weights of the following jobs become greater, the greater the weights of the jobs following them, etc. Thus all the actions lagging behind a given job accumulate in the weight of the given j.

Everything stated above about conjugate variables (Lagrange multipliers) can serve as a starting-point for various heuristic algorithms, in cases where joint solution of the direct and dual problems is impossible for some reason. The reasons to construct such approaches are, for example, on one hand excessive high dimensions (one hundred thousand variables) of the problem, and on the other hand the necessity to obtain a solution in an extremly short time. The latter takes place in short-term planning for fast proceeding processes. Heuristic procedures could also be used to obtain rough upper bounds for a length of the schedule.

Here we consider the approach, which is based on the utilization of conjugate variables as job priorities.

The most labour-consuming operation in the algorithm is the solution of the LP problem at each time step. If one solves it by using the simplified component-wise descent method the following procedure is used:

(i) Choose the maximal positive coefficient in the hamiltonian (12). Let it be p_j.

(ii) Set the corresponding $u^j(t)$ equal to

$$\min_{r_j^i \neq 0} \left\{\frac{R^1(t)}{r_j^i(t)}, \ldots, \frac{R^{N_1}(t)}{r_j^{N_1}(t)}, h^j(t)\right\} . \qquad (37)$$

Pass to the next choice within the other positive coefficients, (i) - (ii). Note that if the coefficient is equal to zero (that means the performance of the corresponding job is not admissible by (α) constraints or is completed) we assume the corresponding u^j is equal to zero.

Thus we have obtained a well-known priority method. The idea of the method is to assign each job some number (priority) which defines the relative weight of the job. Then at every instant one appoints the performance of the job which has the maximal priority. If resources are available to perform the job, the intensity is set equal to (37), maximal admissible intensity otherwise - zero. Then pass to the next job and so on. Regarding Lagrange multipliers as priorities one has the following rule to calculate them.

Let us consider the problem A. We assume the problem has a solution. That is, there exists $T < +\infty$, for which

$$z(T) \geqq e \quad .$$

where $e = (1,...,1)$ is a vector with N components.
Then for all j

$$p_j(T) = 0$$

except for final fictitious job N+1.

According to (14) the jumps for final jobs, which complete at the instant T are as follows

$$\Delta p_j = p_{N+1}\ (T + 0)\ u^{N+1}\ (T + 0)/u^j(T - 0) \quad .$$

We may assume the intensity and the weight of final fictitious job to be arbitrarily positive numbers (due to homogenity of the conjugate system (13) - (15)). Consequently, without loss of generality one may let

$$p_{N+1}(T + 0)\ u^{N+1}(T + 0) = 1 \quad .$$

Thus we get

$$p_j = \Delta p_j = \begin{cases} 1/u^j(T - 0) \quad , & \text{if } t_f^j = T \\ 0, & \text{if } t_f^j < T \quad . \end{cases}$$

where j is a number of final jobs in the program.

Then considering the jobs of the next job layer in the graph of the program (beginning from the end), we calculate priorities for these jobs as

$$p_j = \begin{cases} 0 \quad , & \text{if } t_f^j < t_0^i \quad , \quad i \varepsilon \Gamma_j^+ \quad , \\ \\ \dfrac{1}{u^j(t_f^j)} \displaystyle\sum_{i \varepsilon \Gamma_j^+} p_i\ u_i(t_0^i) \quad , & \text{if } t_f^j = t_0^i \quad . \end{cases} \tag{38}$$

and so on.

(Where t_0^i is the starting time for job i).

Note that priorities are recalculated at every iteration in accordance with the "new" u(t), t_f and t_0.

Similarly one may construct priorities, which take into account the distance between t_0^i and t_f^j, $i \varepsilon \Gamma_j^+$. Then instead of (38) we get

$$p_j = \begin{cases} 0 \quad , \quad \text{if } t_f^j < t_0^i - x \quad , \quad i \epsilon \Gamma_j^+ \\ \\ \dfrac{1}{u^j(t_f^j)} \sum_{i \epsilon \Gamma_j^+} p_i \, u^i(t_0^i), \text{ if } t_0^i - x \leq t_f^j \leq t_0^i \end{cases} .$$

where x is a fixed parameter $(x > 0)$. Thus we take into account all the so called "subcritical" jobs.

In particular, for constant intensities $u^j(t) = u_0^j$ we have

$$\int_{t_0^j}^{t_f^j} u_0^j dt = 1$$

and

$$\tau_j^0 \overset{\Delta}{=} t_f^j - t_0^j = \frac{1}{u_0^j}$$

τ_j^0 is the duration of job performance.

Note that if none of the jobs of Γ_j^+ begins its performance directly after completion of j-th job $(u^i(t_f^j + 0) = 0, i \epsilon \Gamma_j^+)$, the priority of the j-th job equals zero. In other words the job has zero priority if it does not delay performance of its successors. In this way one may evaluate how critical the job is. From (38) we obtain the following priority

$$p_j = \tau_j^0 \left[\Gamma_j^+\right]^1 .$$

where $\left[\Gamma_j^+\right]^1$ is the number of immediately subsequent jobs for j-th job. This priority is a generalization of a well known priority "the longest operation"

Using conjugate variables in the problem when penalties are given in form (37) we get the following priority rules

$$p_j = \tau_j^0 \sum_{i \varepsilon \Gamma_j^+} \frac{1}{\tau_i^0}$$

That is the most preferable job of the set which is admissible with respect to (α) constraints is the one which has the longest duration and the largest number of successors. Moreover, the priority of the job is greater, the shorter the duration of each of its successors.

If we use penalties for violation (α) constraints in form

$$\frac{dz^j}{dt} = u^j - \sum_{i \varepsilon \Gamma_j^-} \theta_-(1 - z^i)\ \theta_+(z^j) \quad .$$

we immediately get the following priority rule for our particular case:

$$p_j = \tau_j^0\ \Pi\ \tau_i^0 \quad ,$$

where $\Pi\,\tau_i$ is the production of durations of all successor jobs.

In a similar way we may obtain a number of other various priority rules.

It should be emphasized that the "price" for such simplification of the algoritjhm is the solution quality change for the worse. Despite this fact the heuristics developed (as it follows from a preliminary testing) allows one to obtain much better solutions than well known rules-of-thumb algorithms (for example CPM technique).

7. Example

In this section we consider a simple example to illustrate the algorithm.

Let the program consist of seven jobs. The 7-th job is a fictitious one. The graph of the program ((α) constraints) is shown in figure 1.

(β) constraints are as follows:

$$3u_1(t) + 4u_2(t) + 6u_3(t) + 8u_4(t) + 6u_5(t) + 6u_6(t) \leq R(t) \quad .$$

where (see also figure 2)

$$R(t) = \begin{cases} 4, \text{ if } t \leq 1 \quad , \\ 2, \text{ if } 1 < t \leq 3 \quad , \\ 3.5, \text{ if } 3 < t \leq 7 \quad , \\ 5, \text{ if } t > 7 \quad . \end{cases}$$

The constraints on maximal performance intensities are given in the form:

$$0 \leq u^1(t) \leq 0.33 \quad ,$$
$$0 \leq u^2(t) \leq 0.50 \quad ,$$
$$0 \leq u^3(t) \leq 0.50 \quad ,$$
$$0 \leq u^4(t) \leq 0.25 \quad ,$$
$$0 \leq u^5(t) \leq 0.33 \quad ,$$
$$0 \leq u^6(t) \leq 0.50 \quad ,$$
$$0 \leq u^7(t) \leq 0.10 \quad ,$$

In this case equations for phase variables are written as

$$\frac{dz^1}{dt} = u^1\ \theta_-(1 - z^1) \quad ,$$

$$\frac{dz^2}{dt} = u^2\ \theta_-(1 - z^2) \quad ,$$

$$\frac{dz^3}{dt} = u^3\ \theta_-(1 - z^3)\ \theta_+(z^1 - 1) \quad ,$$

$$\frac{dz^4}{dt} = u^4\ \theta_-(1 - z^4)\ \theta_+(z^2 - 1) \quad ,$$

$$\frac{dz^5}{dt} = u^5\ \theta_-(1 - z^5)\ \theta_+(z^3 - 1)\ \theta_+(z^4 - 1) \quad ,$$

$$\frac{dz^6}{dt} = u^6\ \theta_-(1 - z^6)\ \theta_+(z^4 - 1) \quad ,$$

$$\frac{dz^7}{dt} = u^7\ \theta_+(z^5 - 1)\ \theta_+(z^6 - 1) \quad .$$

We have the following expressions for the jump conditions of the conjugate variables

$$\Delta p_1 = \frac{1}{u^1(t_f^1)}\ (p_3(t_f^1 + 0)\ u^3(t_f^1 + 0) + p_4(t_f^1 + 0)\ u^4(t_f^1 + 0)) \quad ,$$

$$\Delta p_2 = \frac{1}{u^2(t_f^2)}\ p_4(t_f^2 + 0)\ u^4(t_f^2 + 0) \quad ,$$

$$\Delta p_3 = \frac{1}{u^3(t_f^3)}\ p_5(t_f^3 + 0)\ u^5(t_f^3 + 0) \quad ,$$

$$\Delta p_4 = \frac{1}{u^4(t_f^4)}\ (p_5(t_f^4 + 0)\ u^4(t_f^4 + 0) + p_6(t_f^4 + 0)\ u^6(t_f^4 + 0)) \quad ,$$

$$\Delta p_5 = \frac{1}{u^5(t_f^5)}\ p_7(t_f^5 + 0)\ u^7(t_f^5 + 0) \quad ,$$

$$\Delta p_6 = \frac{1}{u^6(t_f^6)}\ p_7(t_f^6 + 0)\ u^7(t_f^6 + 0) \quad ,$$

$$\Delta p_7 = 0 \quad .$$

We consider T = 11 and objective function

$$I(z(T)) = \frac{1}{2} \sum_{j=1}^{7} (1 - z^j(11))^2 \quad .$$

The interval [0, 11] is divided into 11 equal parts. Let $u_k^j = 0.1$ ($j=1,\dots,7$; $k=1,\dots,11$) is the starting point.

Below we exhibit the results of calculations at every iteration.

1-st Iteration

Intensities:

time	u^1	u^2	u^3	u^4	u^5	u^6	u^7
1	0.10	0.10	0.10	0.10	0.10	0.10	0.10
2	0.10	0.10	0.10	0.10	0.10	0.10	0.10
3	0.10	0.10	0.10	0.10	0.10	0.10	0.10
4	0.10	0.10	0.10	0.10	0.10	0.10	0.10
5	0.10	0.10	0.10	0.10	0.10	0.10	0.10
6	0.10	0.10	0.10	0.10	0.10	0.10	0.10
7	0.10	0.10	0.10	0.10	0.10	0.10	0.10
8	0.10	0.10	0.10	0.10	0.10	0.10	0.10
9	0.10	0.10	0.10	0.10	0.10	0.10	0.10
10	0.10	0.10	0.10	0.10	0.10	0.10	0.10
11	0.10	0.10	0.10	0.10	0.10	0.10	0.10

Job marks:

time	z^1	z^2	z^3	z^4	z^5	z^6	z^7
1	0.10	0.10	0.00	0.00	0.00	0.00	0.00
2	0.20	0.20	0.00	0.00	0.00	0.00	0.00
3	0.30	0.30	0.00	0.00	0.00	0.00	0.00
4	0.40	0.40	0.00	0.00	0.00	0.00	0.00
5	0.50	0.50	0.00	0.00	0.00	0.00	0.00
6	0.60	0.60	0.00	0.00	0.00	0.00	0.00
7	0.70	0.70	0.00	0.00	0.00	0.00	0.00
8	0.80	0.80	0.00	0.00	0.00	0.00	0.00
9	0.90	0.90	0.00	0.00	0.00	0.00	0.00
10	1.00	1.00	0.00	0.00	0.00	0.00	0.00
11	1.00	1.00	0.10	0.10	0.00	0.00	0.00

Objective Function: I = 2.31

2-nd Iteration

Intensities

time	u_1	u_2	u_3	u_4	u_5	u_6	u_7
1	0.33	0.50	0.00	0.00	0.00	0.00	0.00
2	0.33	0.25	0.00	0.00	0.00	0.00	0.00
3	0.33	0.25	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.50	0.06	0.00	0.00	0.00
5	0.00	0.00	0.50	0.06	0.00	0.00	0.00
6	0.00	0.00	0.00	0.25	0.00	0.00	0.00
7	0.00	0.00	0.00	0.25	0.00	0.00	0.00
8	0.00	0.00	0.00	0.25	0.00	0.00	0.00
9	0.00	0.00	0.00	0.12	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.33	0.50	0.00
11	0.00	0.00	0.00	0.00	0.33	0.50	0.00

Job Marks:

time	u_1	u_2	u_3	u_4	u_5	u_6	u_7
1	0.33	0.50	0.00	0.00	0.00	0.00	0.00
2	0.67	0.75	0.00	0.00	0.00	0.00	0.00
3	1.00	1.00	0.00	0.00	0.00	0.00	0.00
4	1.00	1.00	0.50	0.06	0.00	0.00	0.00
5	1.00	1.00	1.00	0.12	0.00	0.00	0.00
6	1.00	1.00	1.00	0.37	0.00	0.00	0.00
7	1.00	1.00	1.00	0.62	0.00	0.00	0.00
8	1.00	1.00	1.00	0.87	0.00	0.00	0.00
9	1.00	1.00	1.00	1.00	0.00	0.00	0.00
10	1.00	1.00	1.00	1.00	0.33	0.50	0.00
11	1.00	1.00	1.00	1.00	0.67	1.00	0.00

Objective function: I = 0.56

3-rd Iteration (Optimal Solution)

Intensities

time	u_1	u_2	u_3	u_4	u_5	u_6	u_7
1	0.33	0.50	0.00	0.00	0.00	0.00	0.00
2	0.33	0.25	0.00	0.00	0.00	0.00	0.00
3	0.33	0.25	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.25	0.25	0.00	0.00	0.00
5	0.00	0.00	0.25	0.25	0.00	0.00	0.00
6	0.00	0.00	0.25	0.25	0.00	0.00	0.00
7	0.00	0.00	0.25	0.25	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.33	0.50	0.00
9	0.00	0.00	0.00	0.00	0.33	0.50	0.00
10	0.00	0.00	0.00	0.00	0.33	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.10

Job Marks:

time	z_1	z_2	z_3	z_4	z_5	z_6	z_7
1	0.33	0.50	0.00	0.00	0.00	0.00	0.00
2	0.67	0.75	0.00	0.00	0.00	0.00	0.00
3	1.00	1.00	0.00	0.00	0.00	0.00	0.00
4	1.00	1.00	0.25	0.25	0.00	0.00	0.00
5	1.00	1.00	0.50	0.50	0.00	0.00	0.00
6	1.00	1.00	0.75	0.75	0.00	0.00	0.00
7	1.00	1.00	1.00	1.00	0.00	0.00	0.00
8	1.00	1.00	1.00	1.00	0.33	0.50	0.00
9	1.00	1.00	1.00	1.00	0.67	1.00	0.00
10	1.00	1.00	1.00	1.00	1.00	1.00	0.00
11	1.00	1.00	1.00	1.00	1.00	1.00	0.10

Objective function: $I = 0.40$

Gant diagram corresponding to the optimal solution is presented in figure 3. A computer program of the algorithm has been written in FORTRAN.

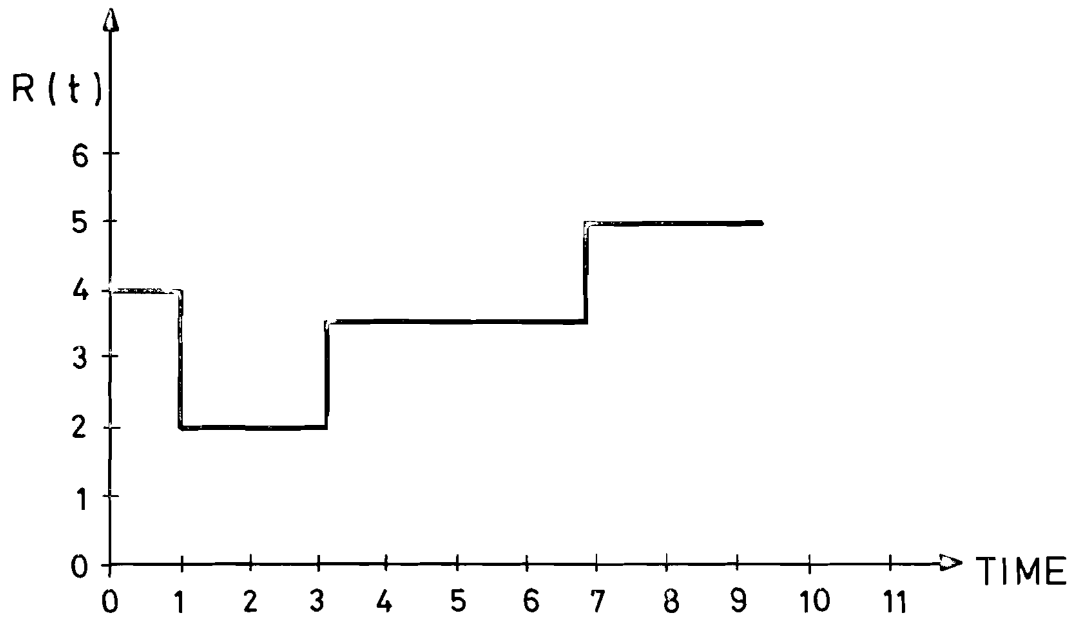
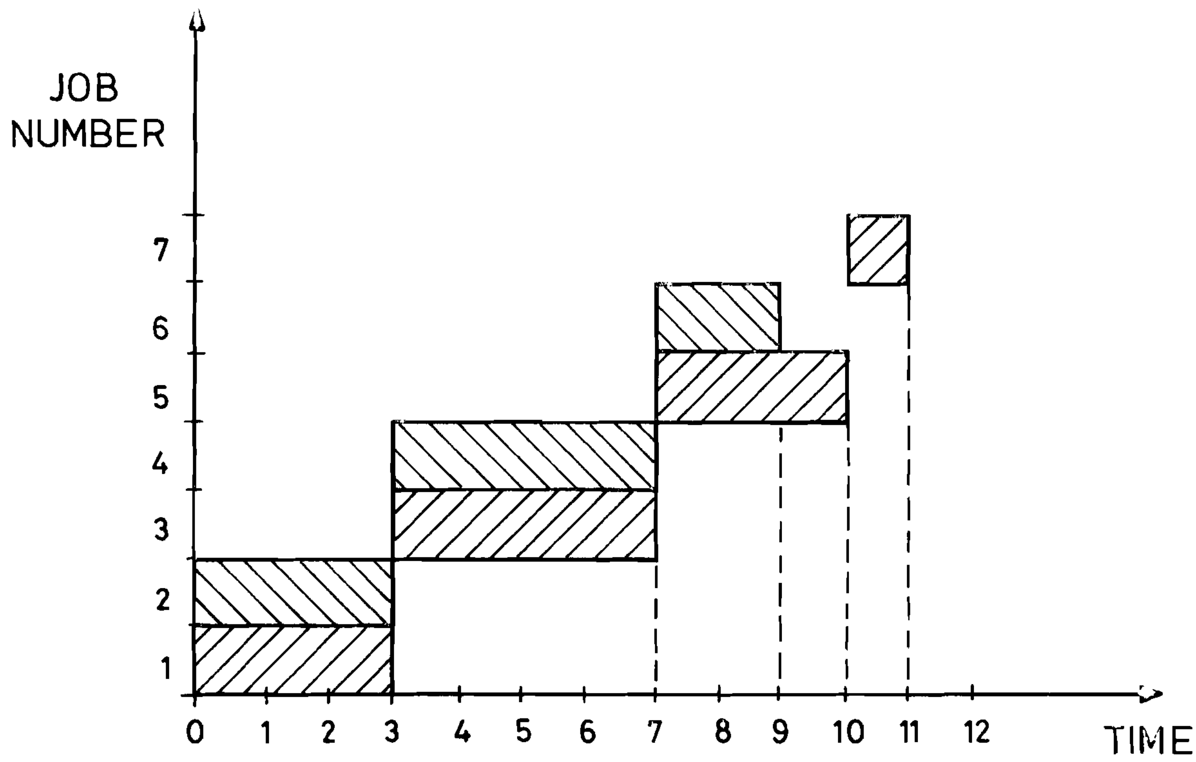

R(t)
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11
TIME

FIGURE 2


JOB NUMBER
7
6
5
4
3
2
1
0 1 2 3 4 5 6 7 8 9 10 11 12
TIME

REFERENCES

[1] Zimin, I.N., "On Some Optimal Control Problems Arising from Project Management", IIASA, WP-75-6, January 1975.

[2] Velitchenko, V.V., "On Some Optimal Control Problems with Discontinues Right-Hand Sides in Dynamics Equations", Vol. 27, No. 7, 1966, (in Russian).

[3] Zimin, I.N. and Ivanilov, Yu. P., "Solution of Network Planning Problems by Reducing them to Optimal Control Problems", Zh. vychisl. Math. and mat. Fiz., Vol. 11, No. 3, 1971 (in Russian).

[4] Moiseev, N.N., "Computational Methods in Optimal Control Problems, "Science", Moscow, 1971.

[5] Beiko, I.V. and Karpenko, M.R., "Solution of Nonlinear Optimal Problems by Successive Approximations Method", Doklady A.N., UK SSR, No. 12, 1964. (in Russian).