RM-76-13

OPTIMAL CONTROL THEORY PROBLEMS WITH NETWORK CONSTRAINTS AND THEIR APPLICATION

Igor Zimin

March 1976

Preface

Scheduling problems are related to combinatorial ones and are usually formulated as integer programming problems. Solvability of the latter problems of realistic size is still an open question. In this paper principles of a new approach to solve scheduling problems which is based on implementation of indirect optimal control theory methods are described.

Abstract

Optimization algorithms for solving of a certain class of large-scale optimal control theory problems are described in the paper. The problems could be stated as problems of dynamic balanced allocation of limited resources among activities of a complex project. The problems are attacked by indirect optimal control theory methods based on Pontzyagin's maximum principle. Extensions and applications are also described.

Optimal Control Theory Problems with Network Constraints and their Application

Igor Zimin

I. Introduction

The impact of activity network models on development planning of complex projects, both in practice and research, is well known [5,10,11,17,21,23]. These project planning models are based on network structure. At the beginning of their development they assumed unlimited resources (CPM, PERT [4,6,9,12,14,16]). As a rule, however, this assumption is not valid in real problems. Many researchers were thus forced to deal with problems involving limited resources.

In general the problem of scheduling interacting activities subject to limited resources can be stated as follows.

Consider a graph $\Gamma = (N,G)$ where N, the set of nodes, corresponds to the activities, and G, the set of arcs, corresponds to the precedence relations. N and G could also be used as the number of nodes and precedence, respectively. Any activity $i \varepsilon N$ is characterized by the following information.

t_i^{nom}, t_i = nominal time and shortest time of completion of the i-th activity ($t_i^{nom} \geq t_i$).

The values of completion time are dependent on the vector of resources required (denoted r) and resources available for activity performance (denoted R). Some components of the vector could be equal to zero. The activity could be completed within time t_i if all resources it consumes are totally available. The problem is to find such a schedule (start and finish times of all activities) which minimizes a certain given objective function $I(t_i,\ldots,t_N)$. To solve the problem, we have to determine when the activities of the N set will be or can be performed if the supply of resources (vector R(t)) is given for all t within the planning period [O,T]. Thus, the following inequality should hold:

$$\sum_{i=1}^{N} r_i(t) \leq R(t) \quad ,$$

where $r_i(t)$, $R(t)$ are the vector of resource consumption for the i-th activity of the vector of resource supply corresponding to moment t.

To solve this problem, past researchers pursued the following approaches.

First, the problem was formulated as an integer or general linear program and solved by standard linear programming techniques [15,22,25]. Other approaches were the direct use of some enumerative schemes for constructing an optimal schedule, [7,8,13], and problem formulation in terms of minimaximal paths in a disjunctive graph and solution by network flow methods and implicit enumeration. Finally the problem was formulated as an optimal control theory problem and solved by penalty function methods [20]. The first three approaches have been compared and the essential difficulties of the several methods have been identified in [3]. The results obtained indicate that the procedures have not been used successfully for any problem of realistic size due to the fact that solvability of an integer programming problem decreases fairly rapidly as the number of constraints increases. (The number of constraints is dependent on the number of activities N, number of resources M, and number of steps during the planning interval). The failure of the fourth approach stems from the necessity of solving the M transcedental equation at every time step. As computation time increases exponentially with the growth of the number of resources, the problem becomes unsolvable for realistic sizes of M.

In this paper we present a new approach to the problem, based on the implementation of indirect methods of optimal control theory. It allows one, on the one hand, to deal with quite general scheduling problems of a dynamic kind, and on the other hand to solve large-scale problems arising in practice.

A few applications to water economy development, regional development, an industrial integrated complex operation and computer network optimizations are considered.

2. The Dynamic Scheduling Problem Statement

Given a list of activities P that must be performed, let these activities be numbered, and call P a program and activities jobs. The state of activity i at a given moment t we characterize by the number $z^i(t)$. We assume that $z^i(t)$ is a portion of the completed section of the activity at instant t. The activity is terminated if

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

where N is the total number of activities. We designate the moment t when condition (1) holds t_f^j. The initial state of the i-th activity could be assumed to be equal to zero:

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

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

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

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

Group (α) is a group of precedence network constraints. The predecessor Γ_i^- and successor Γ_i^+ sets for the i-th activity are given. This means that all activities of Γ_i^- must be completed before i is started and all activities of Γ_i^+ can start after the i-th activity is completed. Group (β) includes resource constraints and some others. For instance, various conditions may be imposed on maximum and minimum intensity of the activity performance.

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

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

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

$$0 \leq u^i(t) \leq h^i(t) \quad , \qquad i = 1,\ldots,N \quad , \qquad (5)$$

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

In the model we consider $z(t) = (z^1(t),\ldots,z^N(t)$ as a phase vector and the vector $u(t) = (u^1(t),\ldots,u^N(t))$ 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.

Note that the model presented is a general one and overlaps a wide range of scheduling problems. Below we will consider some of these. All results obtained with the model can be easily interpreted in terms of "classical CPM and PERT language." The model presented is a dynamic model because activity performance is considered over time and space, all activities may change their intensities while being performed, and the inflow intensity is an arbitrary function of time.

3. Examples of Dynamic Scheduling Problems

A. Duration Minimization Problem

$$\begin{aligned} &\text{Min } T \\ \text{s.t.} \quad &(2),\ (3), \\ &(\alpha),\ (\beta), \\ &z(T) \geqq e, \qquad (6) \end{aligned}$$

where e = N-dimensional vector with all components equal to one (e=(1,1,...,1)).

The problem is to carry out the program for a minimum time. In this case $I(u) = T$. T is the time of completing the program. We define the program as completed when all its operations are completed.

B. Deviation Minimization Problem

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

s.t. (2), (3),
(α), (β),

where

T = given period of time (the period of planning);
$||x-y||$ = the distance between two points x and y in N-dimensional space (in a certain given metric).

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

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

where

λ_j = relative "weight" of the j-th job in the program.

C. Cost Minimization Problem

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

s.t. (2), (3)
(α), (β),
(6).

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

D. Due Date Minimization Problem

$$\min_{} \max_{j \in P} (t_f^j - t_D^j)_+$$

s.t. (2), (3)

(α), (β)

(6).

The problem is to minimize the maximum deviation between time t_f^j, when the corresponding job is actually completed, and the due date of completion which is designated by t_D^j. The function $(x)_+$ is defined as follows:

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

E. Plan Deviation Minimization Problem

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

s.t. (2), (3)

(α), (β)

(6).

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

F. Consumption Deviation Minimization Problem

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

s.t. (2), (3),

(α), (β),

(6).

where

F*(t) = a given (desirable) resource consumption by the program;

F(u(t)) = resource consumption under given constraints.

For example, the objective function may be assumed as

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

Many other problems may be stated, in a similar way and all objective functions described may be combined.

To simplify the discussion we will 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 to one may be obtained on the basis of the solution to the other. 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 is degenerate. But this difficulty is easily overcome by the introduction of a dummy job[1], which can start when all activities of the program have terminated. The intensity of this job should be assumed to be constant and less than or 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 minimizing (8) for the initial program, a similar objective function for the "lengthened" program (that is for extended vector $\hat{z} = (z, z_{N+1})$) has to be minimized.

If the final dummy job has not started its performance at a given T, then $T < T^*$. On the other hand, if dummy job number z^{N+1} becomes non-zero, then $T > T^*$. Thus, during

[1]We define a dummy job as one which has non-zero duration and does not consume resources.

computation, a dual upper and lower estimate is obtained for the optimal time of performing the program.

4. Solution of the Problem

Thus we consider the solution of the following problem:

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

s.t. (2), (3)
(α), (β),
(6).

Note that most computational methods for obtaining optimal solutions can be considered as a utilization of explicit or implicit penalties for constraint violation. In some cases, there is a feedback between deviation size from the optimal solution and size of the penalty (implicit case). This feedback 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 precedence network constraints (α), introduce the modified system.

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

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

Now every job can be performed (the corresponding u^j may be positive) before the previous activities 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 a job which has terminated or which is not feasible for the (α) constraints can take in an interval $[0,h(t)]$ ($t \in [0,T]$). To avoid this lack of uniqueness, at such instants we will choose $u^j = 0$ from the set of feasible values.

Under these conditions the problem B is equivalent to

---

[2] $$\theta\pm(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x < 0 \end{cases}, \qquad \theta_+(0) = 1, \quad \theta_-(0) = 0$$

the modified problem:

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

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

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

$$0 \leq u^i(t) \leq h^i(t) \quad .$$

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 V.V. Velitchenko [24]. Moreover the maximum principle conditions are (locally) sufficient for this problem. The proof can be found in [26].

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 feasible controls. We define the schedule $u^*(t)$ as a feasible control if it satisfies all (β) constraints. Here $p(t)$, corresponding to the $u^*(t)$ vector of Lagrange multipliers is a solution of the conjugate system:

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

with jump conditions for instants coinciding with the instants at which the activities terminate:

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

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

and boundary conditions

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

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

It is clear that all p_j are piecewise constant functions.

The main purpose of the method is to find controls $u(t)$ and corresponding $p(t)$ which satisfy conditions (13)-(15) and maximize functions $H(u,z,p)$. Such methods are usually called indirect optimal control methods [18].

The following algorithm, based on the method of succesive approximations, will be used for solving the problem. (The flow chart is given in Figure 1).

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

$$H(u(t)^{(k+1)}, z^{(k)}(t), p^{(k)}(t)) = \text{Max } H\ (u, z^{(k)}, p^{(k)}) \quad , \tag{16}$$

where the maximum is taken under (β) constraints.

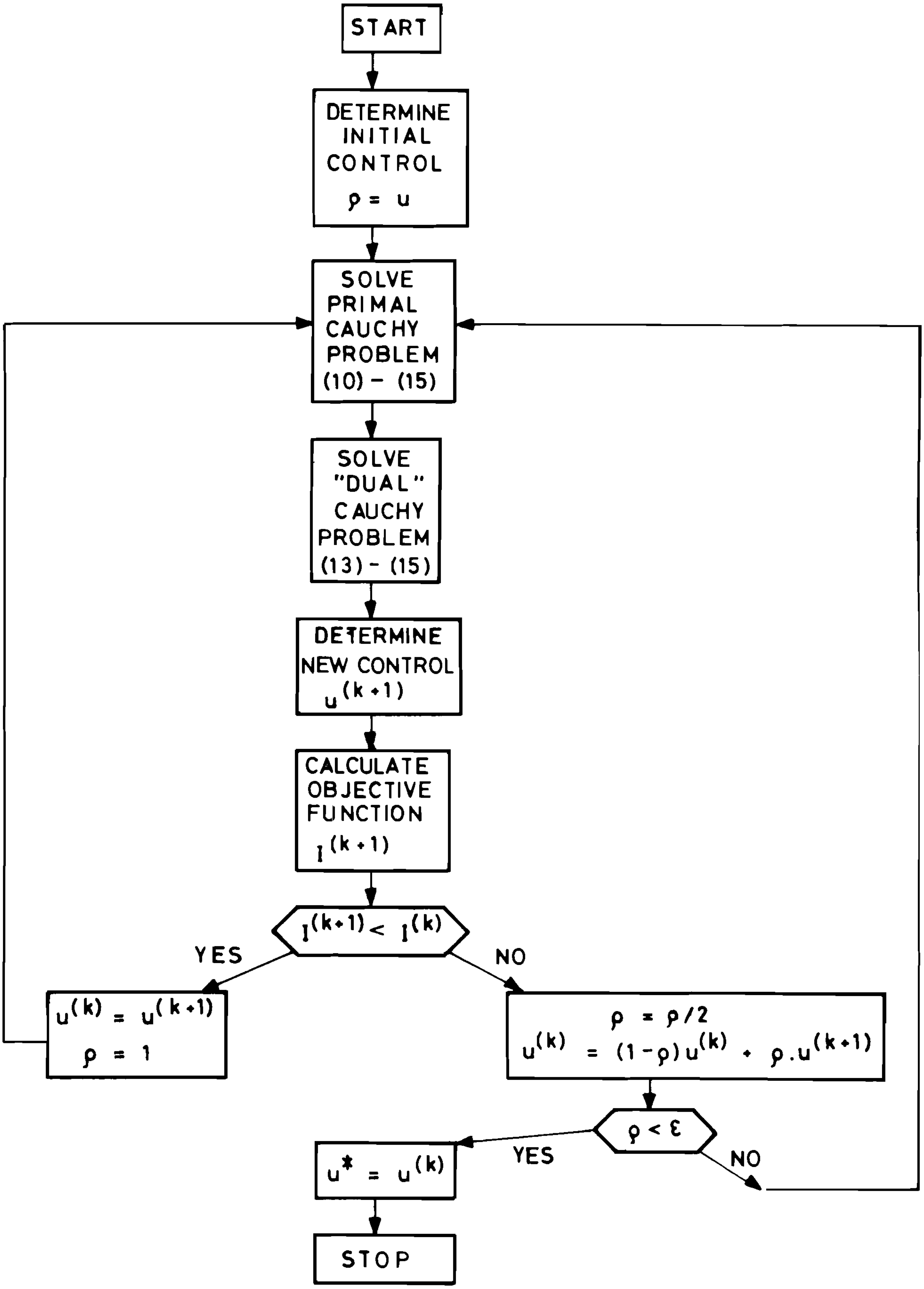


FIGURE 1. COMPUTATIONAL FLOW CHART OF THE ALGORITHM.

(iv) Compare $I^{(k)}$ and $I^{(k+1)}$.[3]
If

$$I^{(k)} < I^{(k+1)} \quad ,$$

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

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

pass to (v).

(v) Calculate new controls as follows:

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

$$0 \leq \rho \leq 1 \quad .$$

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

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

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

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

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

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

---

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

If

$$\rho \geq \varepsilon$$

pass to (v).

The proof of algorithm convergence is similar to that in [1]. The specific character of the problem and its solution by the method should be outlined. When integrating the system (10) in (i), the following linear programming problem (LPP) is to be solved at every time step:

$$\text{Max } H(u, z^{(k)}, p^{(k)}) \qquad (17)$$

s.t. (β) constraints.

The coefficient attached to u^j in the Hamiltonian function is zero for those jobs which do not satisfy (α) constraints or which 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 admissible 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 technologically feasible jobs at the instants.

The number of linear constraints at each time step is equal to the number of different resources consumed by these 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 have been completed.

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

The positive features of the method are:

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

(ii) the simplicity of the computer program;

(iii) a relatively 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 need not be solved with great 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.

The shortcomings of the algorithm are:

(i) in general the algorithm makes it possible to obtain a solution which corresponds to a local minimum of the objective function;

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

These shortcomings can be overcome. The first will be discussed in Section 5; the second may be removed by somewhat sophisticating the computer program.

Some generalizations of the model are discussed below.

5. Additional Constraints on Program Performance

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

Let us modify the initial problem by introducing, instead of I(u), the objective function

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

where $\varepsilon > 0$ is a sufficiently small number.[4] In this case the Hamiltonian

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

is a strictly concave function.

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, one now 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)).

B. Storable Resources

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

---

[4] It is easy to verify that ε is subjected to the following constraints:

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

where

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

and

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

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

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

where

$g_j^k(t)$ = the intensity with which the k-th storable resource is consumed at instant t when performing the j-th activity with unit intensity;

$Q^i(t)$ = the intensity of inflow of the i-th storable resource, released for performance of the program at instant t;

N_2 = 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:

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

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

Denote functions $F^i(t)$ as

$$F^i(t) = \int_0^t Q^i(\tau)d\tau \quad . \qquad (21)$$

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

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

Thus we get the control problem with phase constraints.

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

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

$$z^j(0) = 0 \quad , \qquad (23)$$

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

where

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

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

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

where

p_j ≡ the Lagrange multipliers that corresponds to the variable z^j and satisfies the conditions (13)-(15);

σ_i ≡ the Lagrange multiplier that corresponds to the variable z^{N+i} and satisfies equation

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

boundary condition

$$\sigma_i(T) = 0 \quad ,$$

and jump condition

$$\sigma_i(t_H^i - 0) - \sigma_i(t_H^i + 0) = \frac{\sum_{j=1}^{N} \gamma_{ij} p_j (t_H^j + 0) u^j (t_H^j + 0)}{\sum_{j=1}^{N} q_j^i (t_H - 0) u^j (t_H^i - 0)} \quad , \tag{24}$$

where we denote t_H^j 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 $F(t) = (F_1(t),\ldots,F_{N_2}(t))$. Whenever this occurs, conjugate variable σ_i is subjected to jumps (24).

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

C. 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 with no preemption (for example, certain 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:

$$u^j(t) \geq s^j(t)\theta_-(z^j(t))\theta_-(1 - z^j(t)) \quad , \tag{25}$$

where

$s^j(t) \equiv$ minimal admissible intensity of carrying out the job j at 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 $\theta_-(1 - z^j) \prod_{\ell \in \Gamma_j^-} \theta_{\pm}(z^\ell - 1)$ and integrating from $\tau = 0$ up to $\tau = t$, we get

$$z^j(t) \geq \int_0^t s^j(\tau)\theta_-(z^j) \; \theta_-(1 - z^j) \prod_{\ell \in \Gamma_j^-} \theta(z^\ell - 1)d\tau \quad . \tag{26}$$

Here we used equations (10).

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

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

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

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

Note that constraints (26) are equivalent to

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

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

$$\frac{dz^j}{dt} = u^j\theta_-(1 - z^j)\theta_-(z^j - z^{N+j}) \prod_{\ell \in \Gamma_j^-} \theta_+(z^\ell - 1) \quad . \tag{29}$$

The maximum principle conditions for this (modified) problem are

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

where

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

and

$p_j(t)$ = Lagrange multiplier ($j = 1,\ldots,N$) which satisfies equations (13), boundary conditions (15) and jump conditions

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

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

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

Here

t_q^j = the first moment when the trajectory z^j achieves the boundary of the domain

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

σ_j = conjugate variable to z^j ($j = N+1,\ldots,2N$) which satisfies the equation

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

boundary condition

$$\sigma_j(T) = 0$$

and jump conditions

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

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 includes information on auxiliary variable trajectories, instants t_q^j, and jumps (31)-(34). The dimension of the LP problem to be solved at each time step does not increase.

D. Constraints on Simultaneous Performance of Jobs

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

Conversely, certain jobs may be subject to the restriction that their performance cannot be shared over time; for instance, job j cannot be performed simultaneously with job k. These constraints 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

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

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

The maximum principle holds and the algorithm does not change.

E. 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 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 can be written in the form:

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

This means that the j-th job cannot be performed until all preceding jobs have been completed, and its performance breaks off when the succeeding jobs begin operation.

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

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

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

where

f_j = function of phase variables, control variables and vector of parameters μ; the function has 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

$f_j \to f_j^0 \leq 0$,

as parameters tend to certain limits.

If we consider problem A we obtain

$$T^*(\mu) \leq T^* ,$$

where

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

T^* = 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 function may be constructed in such a way that

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

with a certain variation of the parameter μ.

For example, take the penalty functions

$$f_j(z,\mu) = (1 - \sum_{\ell \varepsilon \Gamma_j^-} \mu_\ell^j (1 - z^\ell) - \sum_{k \varepsilon \Gamma_j^+} \mu_k^j z^k)^{\mu 0} ,$$

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 $\mu_j^i (i,j = 1,...,N)$ are large positive numbers and $\mu 0$ is an 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:

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

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

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

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

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

The utilization of smooth penalties allows us to apply direct optimization methods to solve the problem [18] and combine them rationally with the indirect methods described above.

F. Some Heuristic Approaches to Solving 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. It 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 terminated its weight is non-zero and equal to $\lambda_j(1 - z^j(T))$.

The zero value of p_j at the instant of terminating job ($t = t_f^j$) is increased by a jump (14). This increase is the larger, the greater the weight of the jobs immediately following the j-th, and the greater the intensity of performing these jobs (at previous iterations of the algorithm). It is the smaller, the less intensively the j-th action was performed at instant t_f^j. We may consider that 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 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, since 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 the 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 reason for constructing such approaches are, for example, excessively high dimensions (1000,000 variables) of the problem on the one hand, and on the other hand the necessity to obtain a solution in an extremely 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 schedule.

Here we consider the approach based on the utilization of conjugate variables as job priorities. The most labor-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

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

Pass to the next choice within the other positive coefficients (i)-(ii). Note that if the coefficient is equal to zero which means the performance of the corresponding job is not admissible by (α) constraints or is completed) we assume that 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 indicates the performance of the job which has the maximal priority. If the resources are available to perform the job, the intensity is set equal to (37)(maximal admissible intensity), and otherwise equal to 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 problem A. We assume that the problem has a solution; that is that there exists $T < +\infty$, for which

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

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

$$p_j(T) = 0 \quad ,$$

except for the final dummy job N+1.
According to (14), the jumps for final jobs which are completed at 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 the final dummy job to be arbitrarily positive numbers (due to the homogeneity 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

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

where j is the number of the final job 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

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

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, $i \in \Gamma_j^+$. Then instead of (38) we get

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

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

$$\tau_j^0 = t_f^j - t_0^j = \frac{1}{u_0^j} \quad .$$

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 the j-th job ($u^i(t_f^j + 0) = 0$, $i \in \Gamma_j^+$), the priority of the j-th job equals zero. In other words the job has zero priority if this 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,

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

where $\left[\Gamma_j^+\right]^1$ is the number of jobs immediately succeeding the 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 rule:

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

This means that 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 the greater, the shorter the duration of each of its successors.

If we use penalties for violation of (α) constraints in the 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 algorithm is a change for the worse in the solution quality. Despite this fact the heuristic developed (as follows from a preliminary testing) allows one to obtain much better solutions than well-known rule-of-thumb algorithms, for example the CPM technique.

5. 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 dummy one. The graph of the program ((α) constraints) is shown in Figure 2.

(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 3)

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

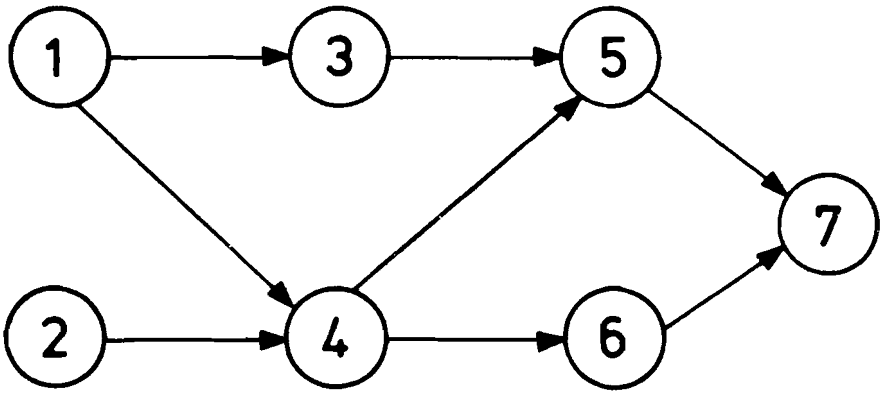


FIGURE 2. NUMERICAL EXAMPLE:
ACTIVITY NETWORK.

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

$$0 \le u^1(t) \le 0.33 \quad ,$$
$$0 \le u^2(t) \le 0.50 \quad ,$$
$$0 \le u^3(t) \le 0.50 \quad ,$$
$$0 \le u^4(t) \le 0.25 \quad ,$$
$$0 \le u^5(t) \le 0.33 \quad ,$$
$$0 \le u^6(t) \le 0.50 \quad ,$$
$$0 \le u^7(t) \le 0.1 \quad .$$

In this case equations for phase variables are written as

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

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

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

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

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

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

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

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

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

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

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

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

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

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

We consider $T = 11$, and the objective function as

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

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

Below we exhibit the results of calculations at every iteration. (A computer program of the algorithm has been written in FORTRAN).

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: $I = 2.31$

<u>2-nd Iteration</u>:

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

<u>3-rd Iteration (Optimal Solution)</u>

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 = 0.4)

The Gantt diagram corresponding to the optimal solution is presented in Figure 4.

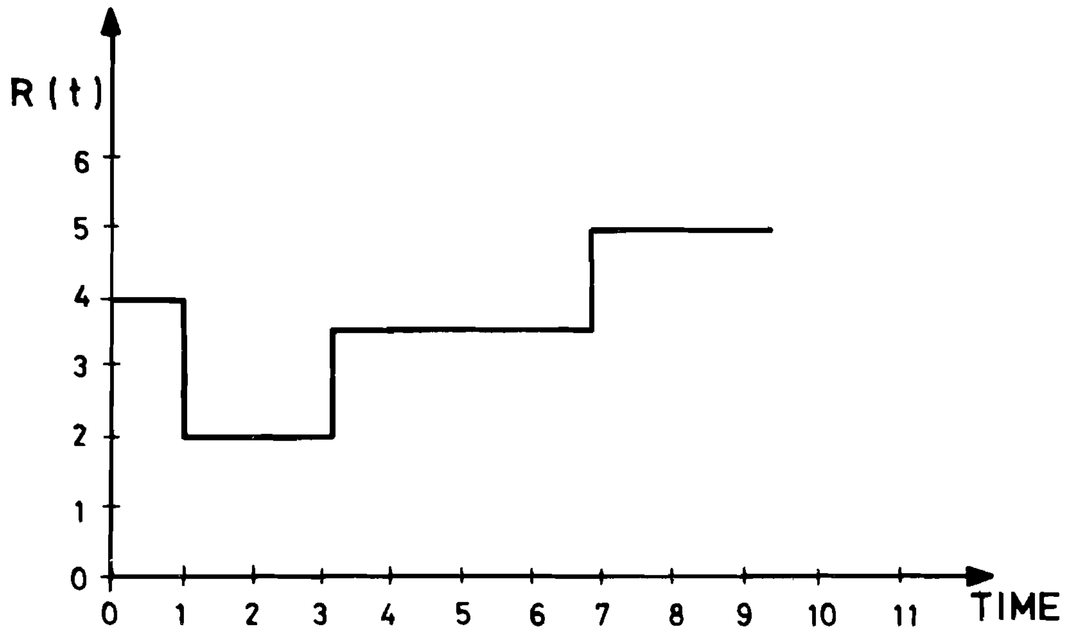


FIGURE 3. NUMERICAL EXAMPLE :
RESOURCE SUPPLY.

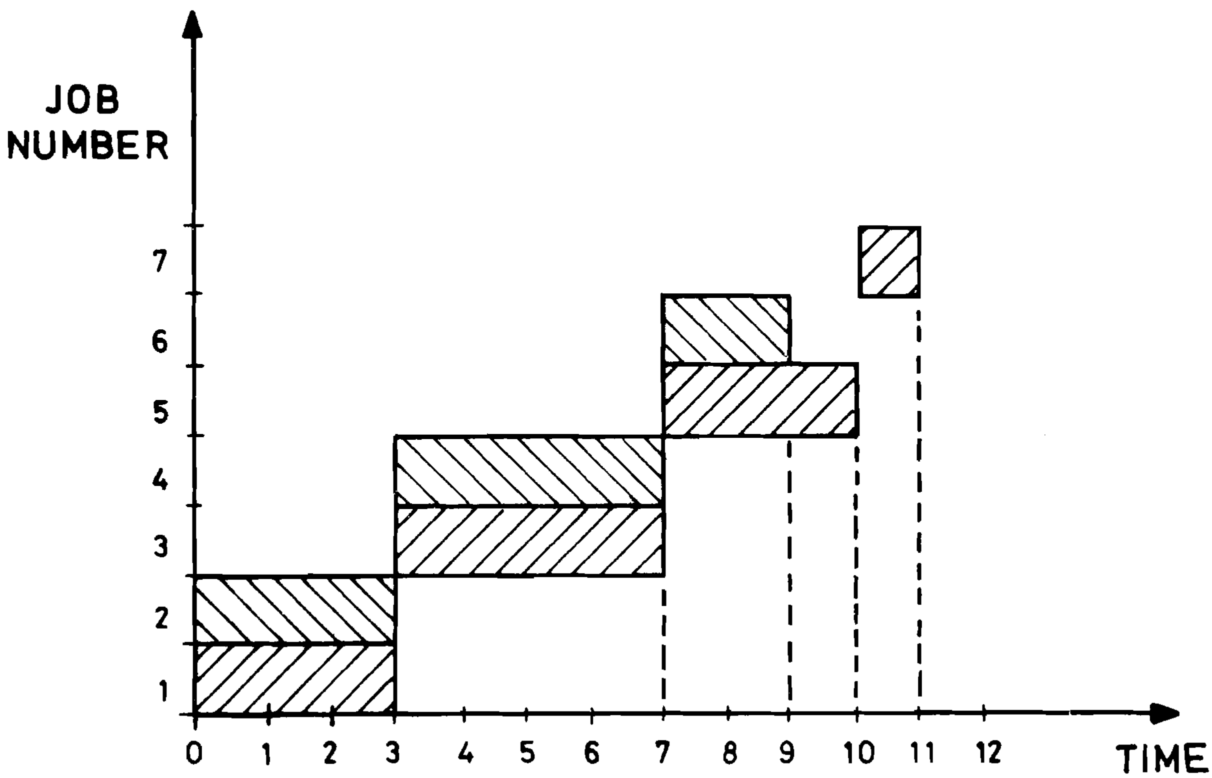


FIGURE 4. OPTIMAL SCHEDULE
(GANTT DIAGRAM)

7. Applications

A. Water Economy: Long- and Short-Term Planning

Here we consider application of scheduling methods developed for long- and short-term planning in the water economy of a region or state.

We assume the water economy to consist of a river basin and a set of water economy structures (reservoirs, canals, conduits, dams, etc.). The problem is to construct a plan for the short-term operation of water economy structures during their expansion, building and reconstruction in order to minimize cumulative losses over the planning period. The losses are due to an insufficient water supply for water consumers and to floods; and the water economy directive body seeks to avoid or at least to decrease the losses as far as possible. Thus the problem is how to distribute limited resources (capital, skills and raw materials) over time and among water structures for their expansion, building and reconstruction, and how to operate them during the planning period in order to minimize the losses. We assume that resource inflows are given for the whole period.

The model of water economy development consists of two sub-models, a river basin model, and a program performance model as described in Section 1.

First consider the river basin model. Here we use the model described in [19] with a few simplifications. Let the river system be represented as a cycle-free precedence network. The nodes of the network are separate cross-sections of the river and its tributaries where intake units (cities, irrigation systems, canals, etc.) are situated. The arcs connecting the nodes are marked by arrows which show the direction of water flow. The water balance equation for the i-th element is

$$W_i(t) = \sum_{l\varepsilon\gamma_i^-} F_{li} - \sum_{k\varepsilon\gamma_i^+} F_{ik}(t) + F_{oi}(t) - F_{io}(t) + \sum_{m\varepsilon\tilde{\gamma}_i^-} E_{mi}(t) - \sum_{n\varepsilon\tilde{\gamma}_i^+} E_{in}(t) - E_{io}(t) \quad , \tag{39}$$

where

$w_i(t)$ = total amount of water in the i-th element at instant t;

$F_{ij}(t)$ = intensity (or rate) of water running from element i (the network node) to element j at instant t;

$F_{io}(t)$ = intensity of water withdrawal in element i at the instant t (intensity of water flow to another sector of the economy;

$F_{oi}(t)$ = intensity of water inflow into element i from without (surface and underground inflow, precipitations);

γ_i^- = set of preceding (upstream) elements (cross-sections, reservoirs, canals);

γ_i^+ = set of elements into which water flows from element i;

$E_{ji}(t)$ = intensity of water inflow into the i-th from the j-th element due to floods;

$E_{ij}(t)$ = intensity of water outflow from the i-th to the j-th element due to floods;

$E_{io}(t)$ = intensity of irreparable losses of water caused by the flood at moment t;

$\tilde{\gamma}_i^+$ = set of elements into which water flows from element i if a flood occurs;

$\tilde{\gamma}_i^-$ = set of elements from which water flows to the i-th element if a flood occurs;

M = total number of elements in the water system i,j, = 1,...,M.

The initial conditions are determined by the state of the basin at the initial moment of the planning period:

$$w_i(t_0) = w_i \quad , \quad 0 \quad . \tag{40}$$

The values of $w_i(t)$, F_{ij}, are limited by the maximum $(w_i(t), \bar{F}_{ij})$ and minimum $(\underline{w}_i(t), 0)$ feasible capacity of water reservoirs and canals:

$$\underline{w}_i(t) \leq w_i(t) \leq \bar{w}_i(t) \quad . \tag{41}$$

$$0 \leq F_{ij}(t) \leq \bar{F}_{ij}(t) \quad . \tag{42}$$

If element i corresponds to a reach of a river or canal, then

$$\bar{w}_i(t) = 0 \quad , \tag{43}$$

as there is no accumulation of water in such reaches. $F_{oi}(t)$ are given functions of time. $E_{ij}(t)$, $E_{io}(t)$ are certain given functions of all other variables and are determined by the amount of water and by the relief in the vicinity of a given river reach.

The second sub-model is the program performance model described in Section 2. The connection between the sub-models is accomplished through the values of maximal capacities of the elements $\bar{W}_i(t)$ and $\bar{F}_{ij}(t)$ as follows:

$$\bar{F}_{ij}(t) = \bar{F}_{ij}(0) + \sum_{k \in K_{ij}} f^k_{ij} z^k(t)$$

$$\bar{W}_i(t) = \bar{W}_i(0) + \sum_{l \in L_i} f^l z^l(t) \quad , \tag{44}$$

where

$\bar{F}_{ij}(0), \bar{W}_i(0)$ = maximal capacities of the i-th element at the beginning of the planning period;

K_{ij}, L_i = sets of activities directed to the expansion, reconstruction and creation of the i-th reservoir and river reach (or canal);

$z^k(t)$ = portion of the k-th job completed by moment t;

f_{ij}^k, f_i^l = additional capacities which come into operation after completing the k-th and the l-th jobs.

The values of E_{ij}, E_{io} are non-negative:

$$E_{ij}(t) \geq 0 \quad ,$$

$$E_{io}(t) \geq 0 \quad .$$

We consider the objective function to be accumulated losses (expressed in monetary units) due to an insufficient water supply for consumers and to damage caused by floods. It is assumed to be given in the following form:

$$I = \int_0^t \sum_{i=1}^{M} \{\lambda_i^W(d_i^W(t) - w_i(t)) + \lambda_i^F(d_i^F(t) - F_{io}(t))$$

$$+ \sum_{n\varepsilon\gamma_i^+} \lambda^F{}_{in}(d_{in}^F(t) - F_{in}(t))$$

$$+ \lambda_i^E(\sum_{m\varepsilon\tilde{\gamma}_i^+} E_{im}(t)) + \lambda_i^o(E_{io}(t))\} \; dt \quad , \qquad (45)$$

where

$\lambda_i^W, \lambda_{in}^F, \lambda_i^F$ = convex penalties for an insufficient water supply for consumers in the i-th element;

λ_i^E, λ_i^o = convex penalties corresponding to the damage caused by floods (destruction of buildings, water economy units, swamping of agricultural areas, etc.);

$d_i^W(t), d_{in}^F, d_i^F$ = water demands corresponding to water consumption in reservoirs, canals, cities and agriculture.

$w_i (i = 1,\ldots,M)$ are phase variables and F_{io}, F_{ij} are controls in the model. Thus in the whole model of river system operation, $w_i(t)$, $z^k(t)$ are phase variables and $F_{io}(t)$, $F_{ij}(t)$, $u^k(t)$ are controls $(i = 1,\ldots,M;\ k = 1,\ldots,N)$. Variables $z^k(t)$, $u^K(t)$ describe long-term development and $w_i(t)$, $F_{io}(t)$,

$F_{ij}(t)$ described short-term operation of the river basin.

Solving the problem

$$I \rightarrow \text{Min} \tag{46}$$

s.t. (39)-(44),
one simultaneously obtains long-term and short-term operational plans for river basin development.

Computationally the problem (46) is no more complex than the problem described in Section 2 because additional constraints can easily be taken into account. The details of the model presented and the numerical algorithm can be found in [2].

B. Short-Term Planning in Industry

A modern industrial complex usually consists of separate production sectors or integrated units which are destined to perform successive-parallel operations. The main characteristics of such complexes are huge flows of raw material, energy, final products and information. Thus, the effectiveness of the work of these complexes is dependent, to a great extent, not only on the effectiveness of the seperate production divisions but also on interaction.
Here we consider one particular set of problems arising in optimization and control of the complex. They are so-called machine sequencing or assembly-line balancing problems and are closely related to the project scheduling problems considered above, since they can be represented on a similar network, although the form of the resource constraints may be quite different.

The problem statement is as follows. Consider a set of S jobs which must be performed. The j-th job consists of n_j tasks numbered from 1 to n_j (see Figure 5). The dynamic equation we represent as

$$\dot{x}^{ij} = u^{ij} \; (1 - x^{i-1,j}(t)) \quad , \tag{47}$$

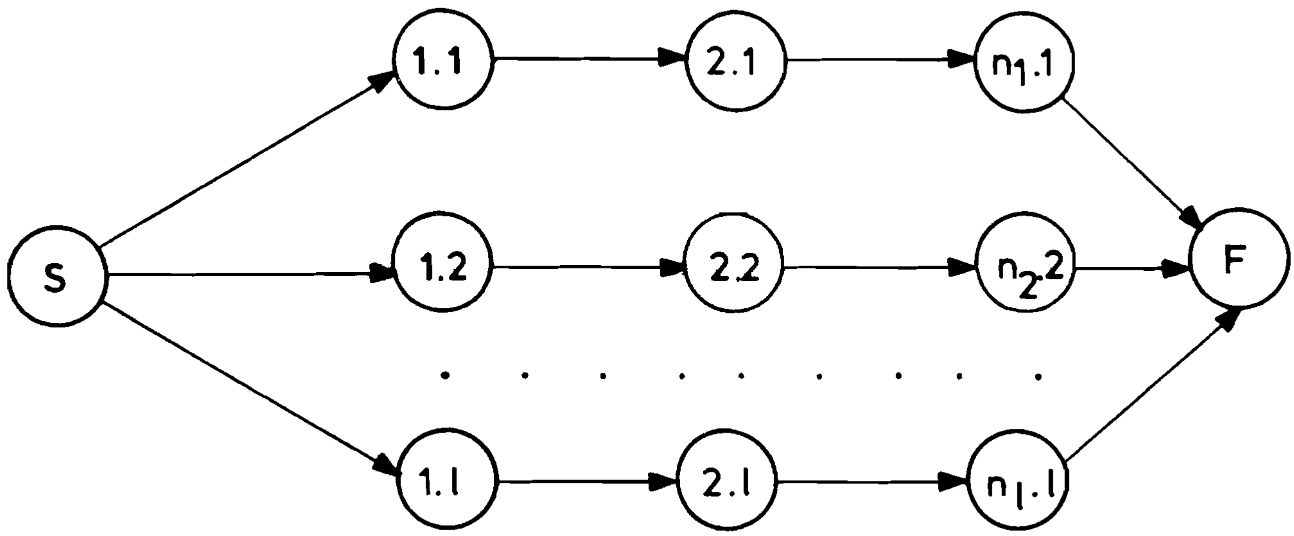


FIGURE 5. ACTIVITY NETWORK IN THE ASSEMBLY-LINE BALANCING PROBLEM.

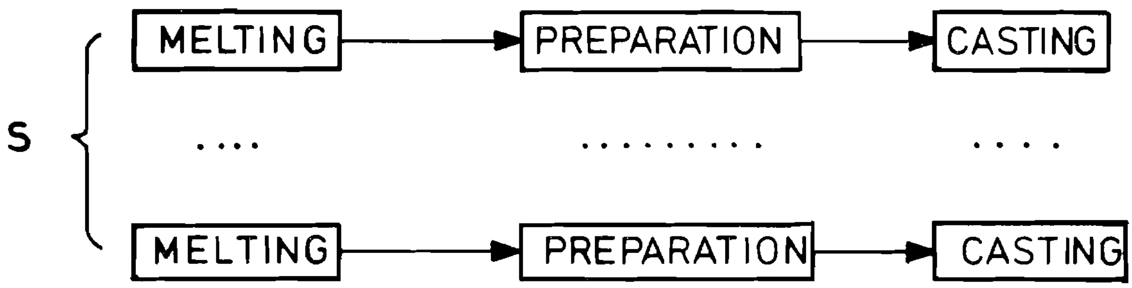


FIGURE 6. NETWORK DIAGRAM FOR MELTING AND CONTINUOUS CASTING PROCESS.

where x^{ij} = portion of the ij-th task performed by moment t. It could be interpreted as a percentage of the total time T_{ij} the task requires for its performance until moment t.
$u^{ij}(t)$ = performance intensity of the ij-th task at instant t.

The initial conditions are:

$$x^{ij}(0) = 0 \quad . \tag{48}$$

We assume that the ij-th task can be completed if $x^{ij}(t) = 1$. Thus, we have the following constraints for every t:

$$x^{ij}(t) \leq 1 \quad , \tag{49}$$

and natural constraints for $u^{ij}(t)$

$$u^{ij}(t) \geq 0 \quad . \tag{50}$$

All relations formulated are valid for:

$$i = 1,2,\ldots,n_j \quad ; \quad j = 1,2,\ldots,S \quad ; \quad t \in [0,T] \quad ; \tag{51}$$

T is the length of the planning period.

In addition, for the final (dummy) task F we have:

$$\begin{aligned} \dot{x}^F &= u^F \prod_{j=1}^{S} \theta(1 - x^{n_j j}) \quad , \\ x^F(0) &= 0 \quad , \\ x^F(t) &\leq 1 \quad , \\ u^F(t) &\geq 0 \quad . \end{aligned} \tag{52}$$

Note that in this case the precedence network has a special structure so that all activities (other than the first and last dummy activities) have exactly one predecessor and one successor each. The nominal time to perform each task is a known integer represented by T_{ij} for the i-th task of the j-th job. Given a set of K different resources to perform the jobs, R_k is the amount of the k-th resource which is available at any time. The amount of the k-th resource required by task ij during its processing is r^k_{ij}. For example, if resources correspond to the machines in a job shop and each task requires only a single machine during the interval of its processing, then k=1 and $r^1_{ij}=1$. Thus resource constraints could be written as:

$$\sum_{i,j} r^k_{ij} u^{ij}(t) \leq R^k \quad , \quad i = 1,\dots,n_j \quad , \quad j = 1,\dots,l$$
$$k = 1,2,\dots,K \quad . \qquad (53)$$

We assume that no preemption of task performance is allowed. Once task ij is started, it must be processed until completed in no more than $\bar{T}_{ij}$ time units and no fewer than $\underline{T}_{ij}$ time units.

The corresponding constraints are written in the following form:

$$u^{ij}(t) \leq \bar{T}_{ij} \qquad (54)$$

$$u^{ij}(t) \geq \frac{1}{\underline{T}_{ij}} \theta(x^{ij}(t))\theta(1 - x^{ij}(t)) \quad .$$

We are required to find intensities for all task performances (vector u) which satisfy all conditions mentioned and for which the total number of jobs (or tasks) completed during the given planning period [O,T] is maximal. Thus the objective function is

$$I(u) = \sum_{j=1}^{S} x^{n_j j}(T) \quad ,$$

and the problem statement is as follows:

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

$$\text{s.t.} \quad (47)-(54) \quad .$$

As an example, let us consider the industrial system consisting of two complexes: oxygen-converters, and continuous casting machines for steel production. (Details can be found in [27]).

The purpose of the oxygen-converter complex is the production of steel of a given composition and temperature. The purpose of the continuous casting machine complex is the continuous casting of steel in slabs of given dimensions.

When scheduling the processing of the complexes, there is the problem of choosing the rythm for all activities in which output steel production is maximal. In other words, the frequency of heat preparation in the oxygen-converter complex should correspond to the productivity of the continuous casting machine complex.

In accordance with the given steel standard, the output, energy demand and certain other characteristics of each complex are given (i.e., in a melting and continuous casting process: melting, preparation for casting, and casting itself). The corresponding network diagram is shown in Figure 6, and the numerical example in Figure 7.

Converters for melting and casts for casting are considered as resources. Certain results concerning computational experiments with the model are shown in Figure 8. The input data for the model are as follows: total number of jobs S = 5; tasks (1.1)-(1.5) correspond to the meltings; tasks (2.1)-(2.5) are preparations for castings; and tasks (3.1)-(3.5) correspond to the casting. Table 1 gives time durations for every task.

TABLE 1

Job Number	Task Number	Duration (in min)	Task Number	Duration (in min)	Task Number	Duration (in min)
1	1	40	2	25	3	40
2	1	35	2	25	3	40
3	1	30	2	25	3	40
4	1	45	2	25.	3	40
5	1	50	2	25	3	40

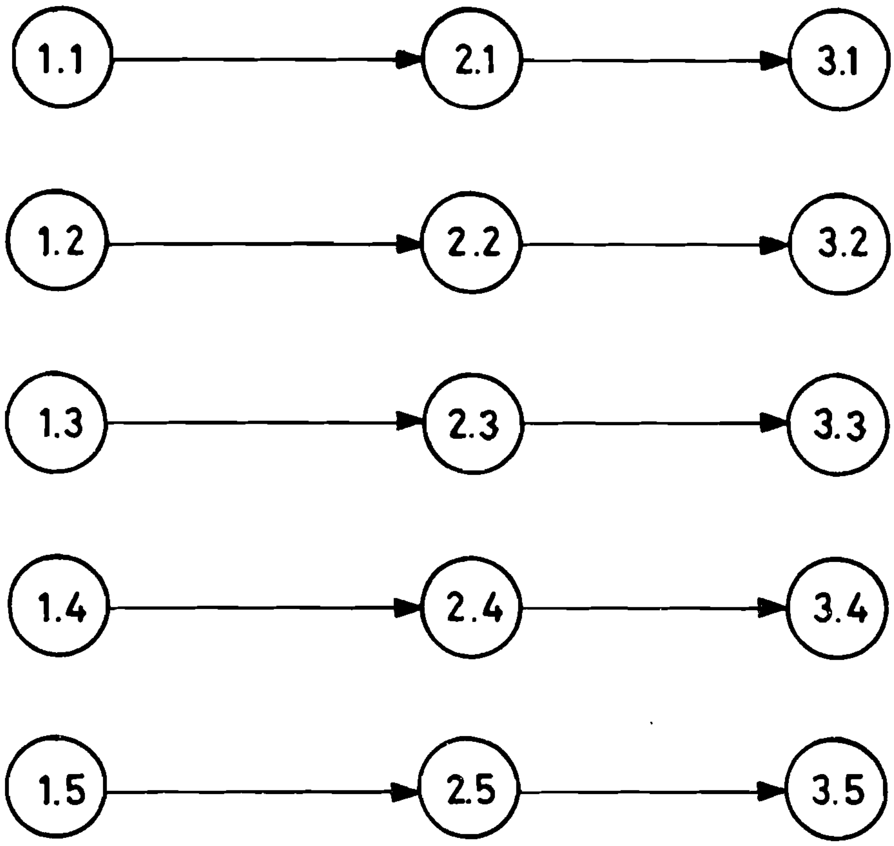


FIGURE 7. NUMERICAL EXAMPLE : NETWORK DIAGRAM FOR MELTING AND CASTING.

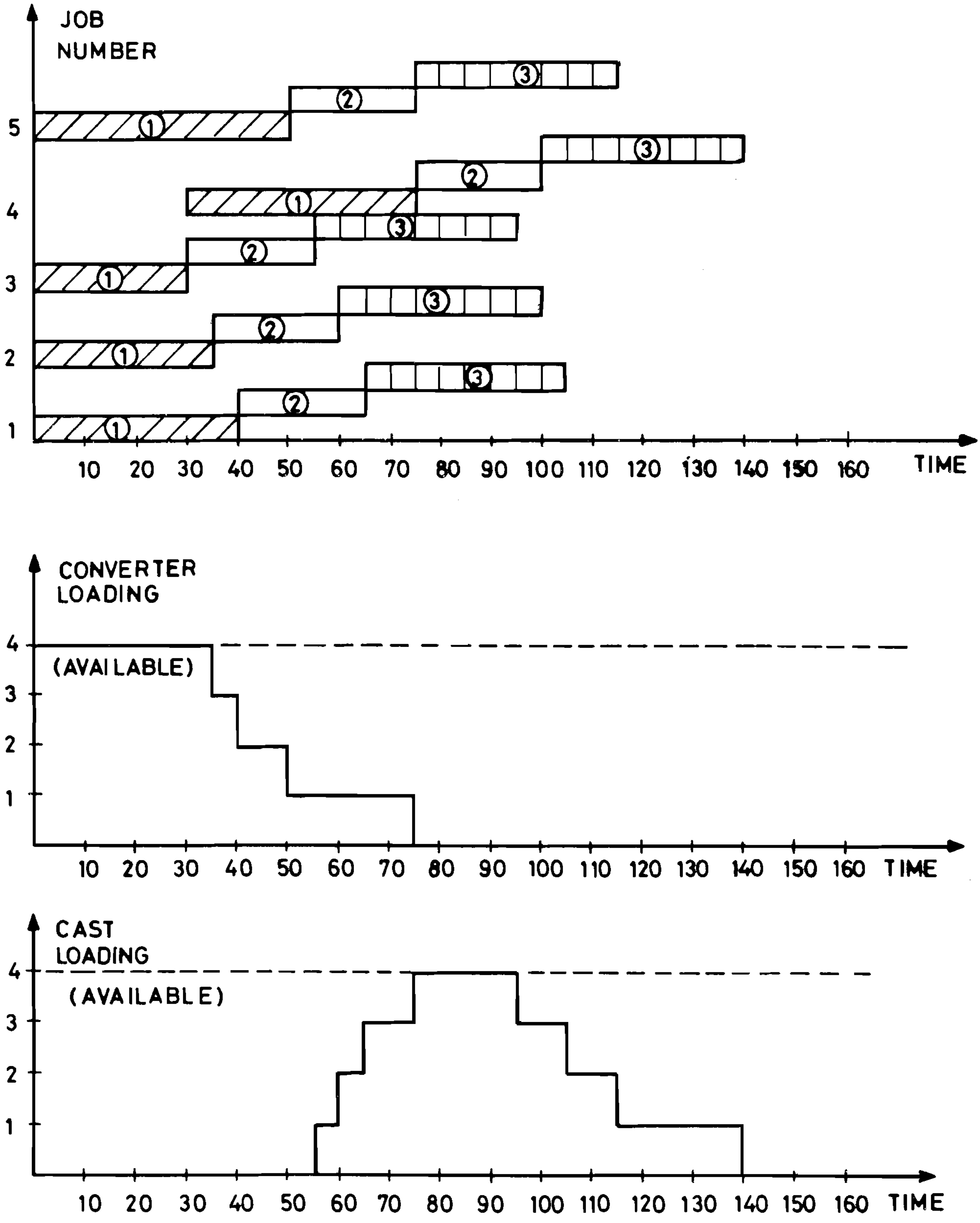


FIGURE 8. NUMERICAL EXAMPLE:
GANTT AND RESOURCE LOADING DIAGRAMS.

Gantt and resource loading diagrams corresponding to the optimal solution of the problem (when T = 140 and the number of converters and casts is 4 each) are given in Figures 7, 8.

C. Joint Calculation of a General Program Performance Model and Dynamic Multi-Branch Industrial Models

The methods suggested in previous sections could be applied in calculating the optimal long-term development plan of a certain complex system such as a state or region. Here we will not give concrete expression to the system under consideration (it could be easily done for every particular case) but assume that the directive body of the system has certain goals, a certain program of actions directed to attain these goals and a multi-branch industry which is to supply the program with the resources required. We refer to the system as the region.

The problem is how to construct the long-term plan of program performance which would allow one to attain given goals in the best way by taking into account limited-resource production rates of the industry.

Thus we consider a general purpose program to be a given set of activities or jobs $\{a_1,...,a_{N_1}\}$ to be carried out to achieve the system goals. The model of program performance is the same as in Section 2. The performance of the program can be controlled because usually there are various ways to carry out its jobs. Indeed there are a number of schedules which satisfy (α) and (β) constraints. Here we assume that the states of the program (vector Z(T)) at the end of the planning period are ranked according to the preferences of the directive body. Thus the calculation of the program is reduced to the following problem: find a schedule which transfers the program from the initial state

$$Z(0) = 0$$

to the point (vector Z(T)) which gives the maximum to the following objective function:

$$I(z(T)) = \sum_{j=1}^{N_1} c_j z^j(T) \quad , \tag{55}$$

where $c_j (j = 1,\ldots,N)$ = relative weight values which reflect directive-body preferences for different program states at instant T.

Let $R_i(t)$ be the M-space resource vector required to perform job i at instant t with unit intensity. We assume that the i-th job consumes the k-th resource if $R_i^k > 0$. Thus the total amount of resources which the program consumes at instant t is equal to $\sum_{i=1}^{N} R_i(t)u^i(t)$. This amount depends on the state of the program z(t) and the intensity u(t). We will write resource constraints as

$$\sum_{i=1}^{N} R_i^k(t)u^i(t) \le r^k(t) \quad , \quad i = 1,\ldots,N_1; \quad k = 1,\ldots,M \quad , \tag{56}$$

where r(t) = vector of resource inflow to carry out the program.

The peculiarity of the problem is that the inflow of resources (vector r(t)) is unknown in advance and can be determined during optimal schedule calculation. The resource inflow is provided by industry and its amount is dependent on the capacities of industry sectors.

The resource supply plan is regarded as a support program. This program is a set of jobs of production, expansion and reconstruction of enterprises. In the two latter cases the jobs to be carried out are unknown in advance and their magnitudes are to be determined during the process of schedule calculation on the basis of the most appropriate resource supply of the general purpose program. Some details on the problem statement can be found in [17] and [28].

Let the regional industry be subdivided into M producing sectors. The balance equations are

$$\dot{q}(t) = x(t) - A(t)x(t) - w(t) - r(t) \quad , \tag{57}$$

where

$q(t) = (q^1(t),\ldots,q^M(t)) \equiv$ stock of resources accumulated up to instant t;

$x(t) = (x^1(t),\ldots,x^M(t) \equiv$ (gross) output;

$x^i(t)$ = output of the i-th industry at instant t;

$A(t)$ = $M \times M$ matrix with elements $a_{ij}(t)$;

$a_{ij}(t)$ = input coefficient of product of sector i into sector j (the quantity of the output sector i absorbed by sector j per unit of its total output);

$w(t) = (w^1(t),\ldots,w^M(t))$ = vector of consumer goods;

$r(t)$ = portion of the final product which is sent into the general purpose and support programs.

Basic dynamics $\xi(t)$ can be written as

$$\xi^i(t) = \xi_o^i + \sum_{n\varepsilon G_i} \theta_n^i z^n(t) \quad , \quad i = 1,\ldots,M \quad . \tag{58}$$

Here

G_i = set of activities directed to the expansion of the i-th industry sector capacities existing at the beginning of the planning period ($t = 0$) ξ_o^i;

θ_n^i = additional capacity of the i-th sector which goes into operation after completion of the n-th job.

If we assume that all capacities are totally loaded at each instant t, then

$$x^i(t) = \xi^i(t) \quad . \tag{59}$$

Substituting (58) and (59) into the right-hand sides of (57) we have

$$\dot{g}^K = \sum_{i=1}^{M} (E - A)_i^K(\xi_0^i + \sum_{n\varepsilon G_i} \theta_n^i z^n) - W^K - r^K \quad , \tag{60}$$

where B_i^k = the k-th element of matrix B;

E = unit matrix of $M \times M$.

For simplicity we assume that all production resources are storable. This leads to the following constraints:

$$q(t) \geq 0 \quad . \tag{61}$$

The consumption required over the planning period $w_o(t)$ is given. Thus we have at each instant:

$$w(t) \geqq w_o(t) \quad . \tag{62}$$

Now the problem of joint calculation of a general purpose problem and multi-branch industry can be stated as follows:

$$cz(T) \rightarrow \text{Max}$$

s.t. (1)-(3), (56), (60)-(62).

In this problem z(t), q(t) are phase variables and u(t), w(t) are controls

To solve the problem the algorithm described in Section 2 is used. In this case the following linear programming problem has to be solved at each time step:

$$H(u,w,r) \rightarrow \text{Max}$$

subject to

$$\begin{aligned} R(t)u - r &\leqq 0 \quad , \\ w &\geqq w_o(t) \quad , \\ u(t) &\geqq 0 \quad , \\ r(t) &\geqq 0 \quad , \end{aligned} \tag{63}$$

where

$$H(u,w,r) = \sum_{j=1}^{N} p_j u^j \theta(z^j - 1) \prod_{l \varepsilon \Gamma_j^-} \theta(1 - z^l) \prod_{k=1} \theta(q^k)$$

$$- \sum_{k=1}^{M} \lambda_k(t)(r^k + w^k)$$ = corresponding Hamiltonian function;

$p_j(t)$, $\lambda_k(t)$ = Lagrange multipliers which satisfy the following conditions:

$$\dot{p}_j + \sum_{k,i=1} \lambda_k (E - A)_i^k \theta_j^i = 0 \quad ,$$

$$p_j(t_f^j - 0) - p_j(t_f^i + 0) = \frac{1}{u^j(t_f^j - 0)} \sum_{k \varepsilon \Gamma_j^+} p_k(t_f^j + 0) u^k(t_f^j + 0) \quad ,$$

$$p_j(T) = c_j \quad ,$$

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

$$\dot{\lambda}_k = 0 \quad ,$$

$$\lambda_k(t_q^k - 0) - \lambda_k(t_q^k + 0) = \frac{1}{|\dot{q}^k(t_q^k - 0)|} \sum_{j=1}^{N} p_j(t_q^k + 0) u^j(t_q^k + 0) \quad ,$$

$$\lambda_k(T) = 0 \quad ,$$

$$k = 1,\ldots,M$$

t_q^k = the instant when the k-th stock (q^k) becomes equal to zero.

The results of computations are illustrated by the following example.

The activity network is given in Figure 9. Here activities 1-7 and 25 (dummy activity) belong to the general purpose program, and activities 7-24 are included in the support program (building and putting into operation additional industry capacities). For example the 1-st sector can be expanded by putting into operation additional capacities θ_{18} and θ_{110}, which correspond to jobs 8 and 10. 7 and 9 are the building of new capacities.

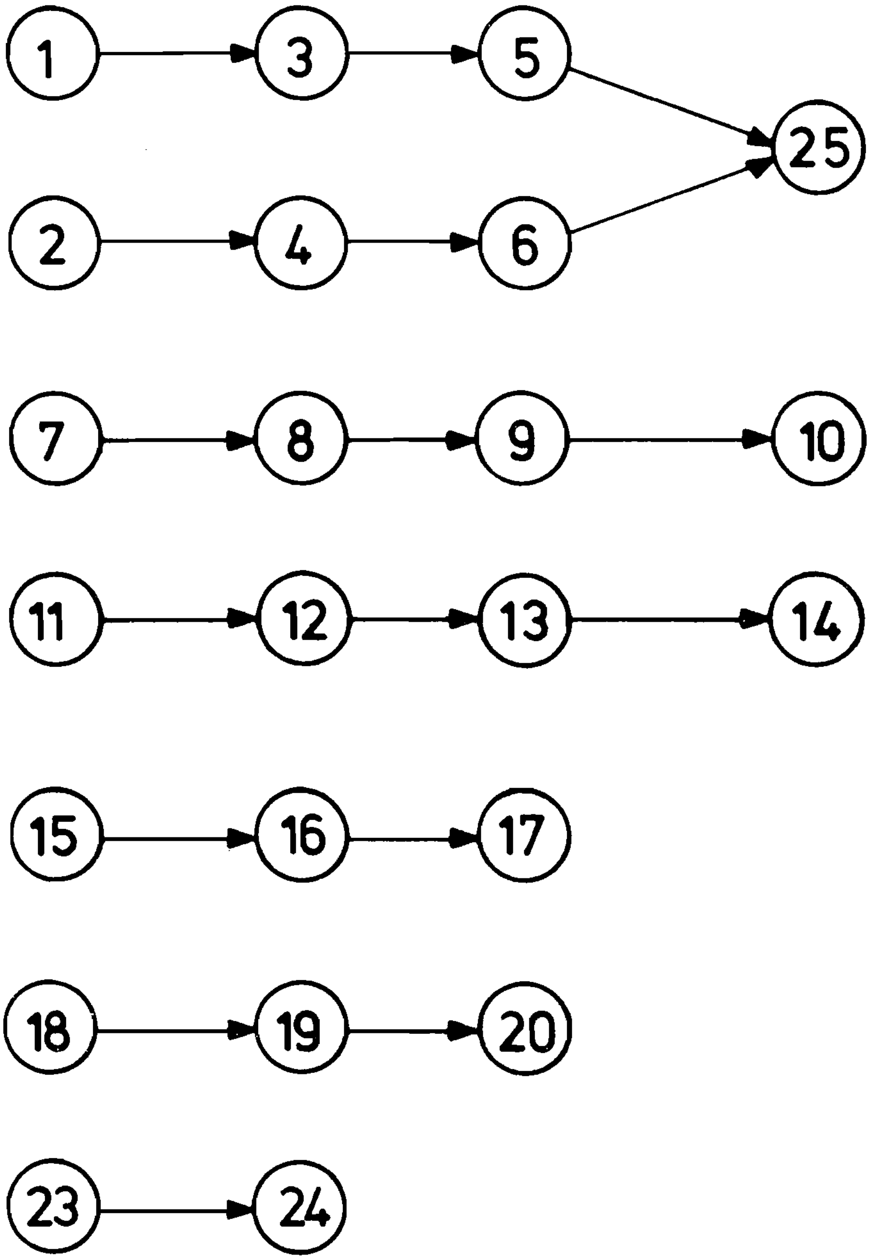


FIGURE 9 : NUMERICAL EXAMPLE:
SIMPLIFIED ACTIVITY NETWORK OF THE GENERAL PURPOSE AND SUPPORT PROGRAMS.

The problem was solved for the following data:
Total number of jobs N = 25,
Number of industry sectors M = 6,
Planning period length T = 9,
Input-output Matrix A is shown in Table 2

TABLE 2

i \ j	1	2	3	4	5	6
1	0.008	0.014	0.012	0.005	0.018	0.008
2	0.038	0.057	0.048	0.023	0.034	0.029
3	0.84	0.45	0.48	0.51	0.39	0.55
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

The initial values of capacities ξ_o^i and stocks q_o^i are:

$$\xi_o^i = 6 \quad ,$$

$$q_o^i = 4 \quad .$$

The additional capacities θ_j^i are equal to 0, except

$$\theta_8^1 = 2, \ \theta_{10}^1 = 4, \ \theta_{12}^2 = 4, \ \theta_{14}^2 = 4, \ \theta_{16}^3 = 14,$$

$$\theta_{17}^3 = 28, \ \theta_{19}^4 = 2, \ \theta_{20}^4 = 4, \ \theta_{22}^5 = 2, \ \theta_{24}^6 = 12 \quad .$$

The required consumption $w_o(t)$ is equal to 0 for all t.
The relative weight values of jobs are

$$c_j = \begin{cases} 1 & \text{for } j = 1,2,\ldots,6: \\ 100 & \text{for } j = 25; \\ 0 & \text{for all others.} \end{cases}$$

The minimal job durations are shown in Table 3.

TABLE 3

Number of Job	Duration
1	2
2	2
3	4
4	2
5	4
6	2
7	2
8	2
9	2
10	2
11	2
12	2
13	2
14	2
15	2
16	2
17	2
18	2
19	2
20	2
21	2
22	2
23	2
24	2
25	100

The resource consumption matrix R is shown as

TABLE 4

Job Number \ Resource Number	1	2	3	4	5	6
1	0	0	0	10	20	40
2	0	0	0	10	20	40
3	0	0	0	10	30	40
4	0	0	0	10	30	40
5	0	0	0	10	40	40
6	0	0	0	10	40	40
7	20	40	0	0	0	0
8	20	40	0	0	0	0
9	20	40	0	0	0	0
10	20	40	0	0	0	0
11	0	10	50	0	0	0
12	0	10	50	0	0	0
13	0	10	50	0	0	0
14	0	10	50	0	0	0
15	40	0	60	0	0	0
16	40	0	60	0	0	0
17	40	0	60	0	0	0
18	10	20	40	0	10	20
19	10	20	40	0	10	20
20	10	20	40	0	10	20
21	10	20	40	0	10	10
22	10	20	40	0	10	10
23	10	20	20	0	10	10
24	10	20	20	0	10	10
25	0	0	0	0	0	0

The solution of the problem is presented in Figures 10-14. It required about two seconds to solve the problem on the CDC-6600 computer. It is easily seen from the figures that the general purpose program consumes resources produced by sectors 4,5 and 6, and that resources produced by sectors 1,2 and 3 are directed to the expansion of the 1-st and 3-rd (industrial sector) capacities. The optimal value of the objective function is equal to 4.15. This means that under given initial conditions the general purpose program could be performed to the extent of 70%. This is caused by the limitation of production of the 4-th, 5-th and 6-th sectors. In spite of this fact the industrial capacities of these sectors were not expanded because their development requires the same kinds of resources as the general purpose program. Thus the performance of activities 18-24 (building and putting into operation additional capacities of the 4-th and 6-th sectors) would decrease the value of the objective function.

At the same time the capacities of the 1-st and 3-rd sectors were expanded because the processing (operation) of these sectors influences the output of all kinds of resources and particularly the output of the 4-th, 5-th and 6-th sectors.

A number of problems could be stated on the basis of the model presented; for example, terms taking into account an increase of consumption over the planning period could be introduced into the objective function, etc.

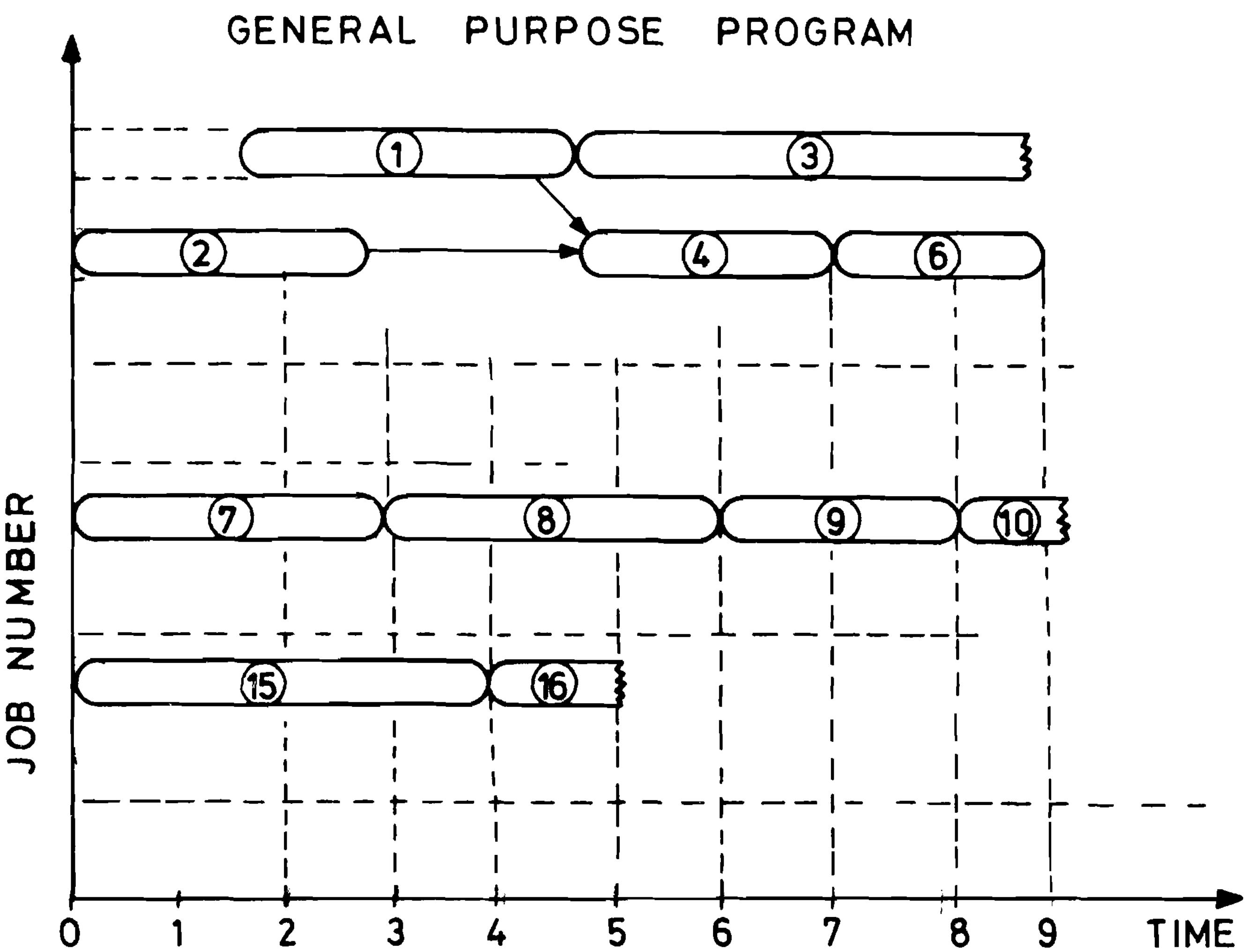


FIGURE 10 : NUMERICAL EXAMPLE:
GANTT DIAGRAM.

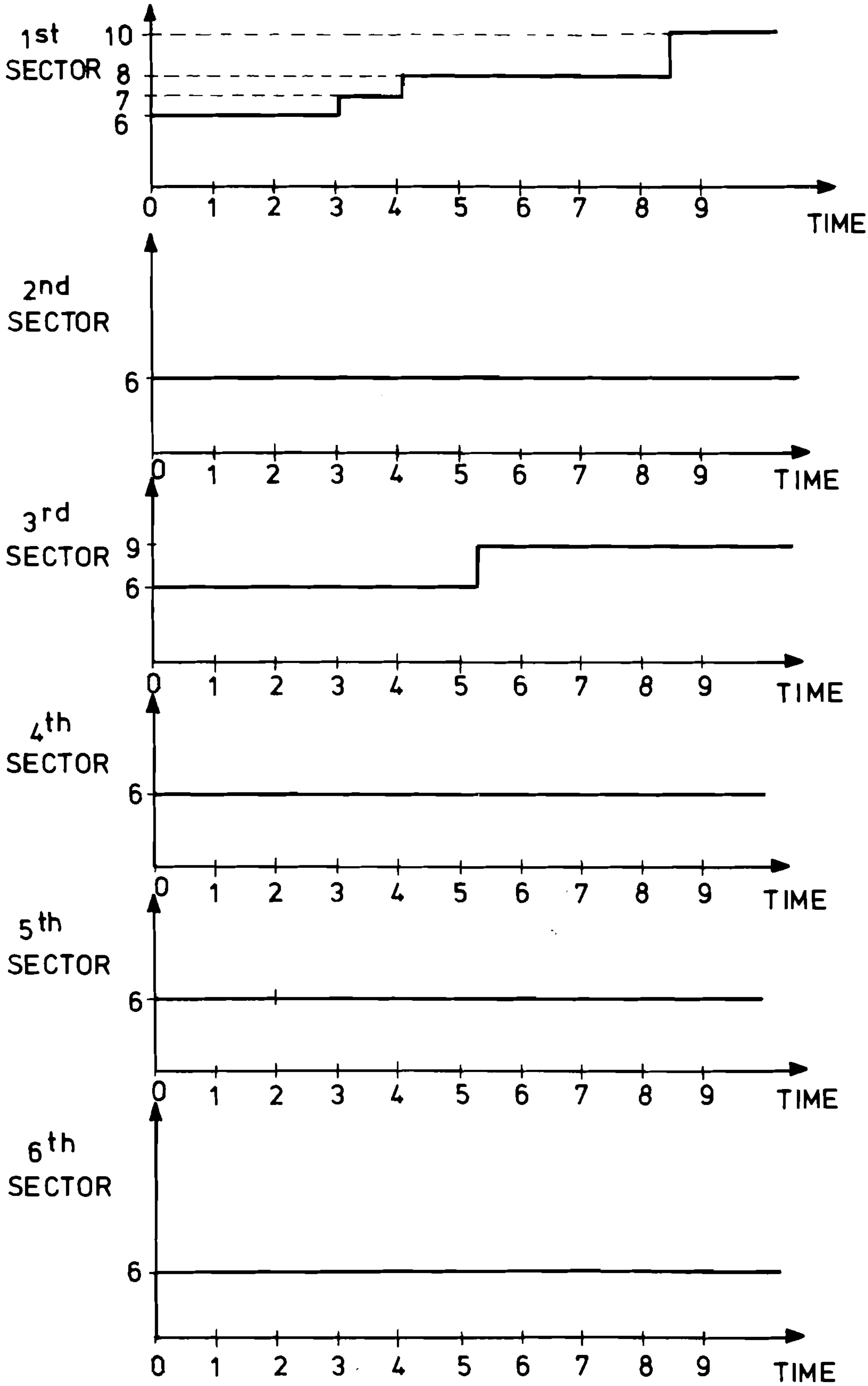


FIGURE 11. NUMERICAL EXAMPLE:
DYNAMICS OF INDUSTRIAL CAPACITIES ($\xi(t)$).

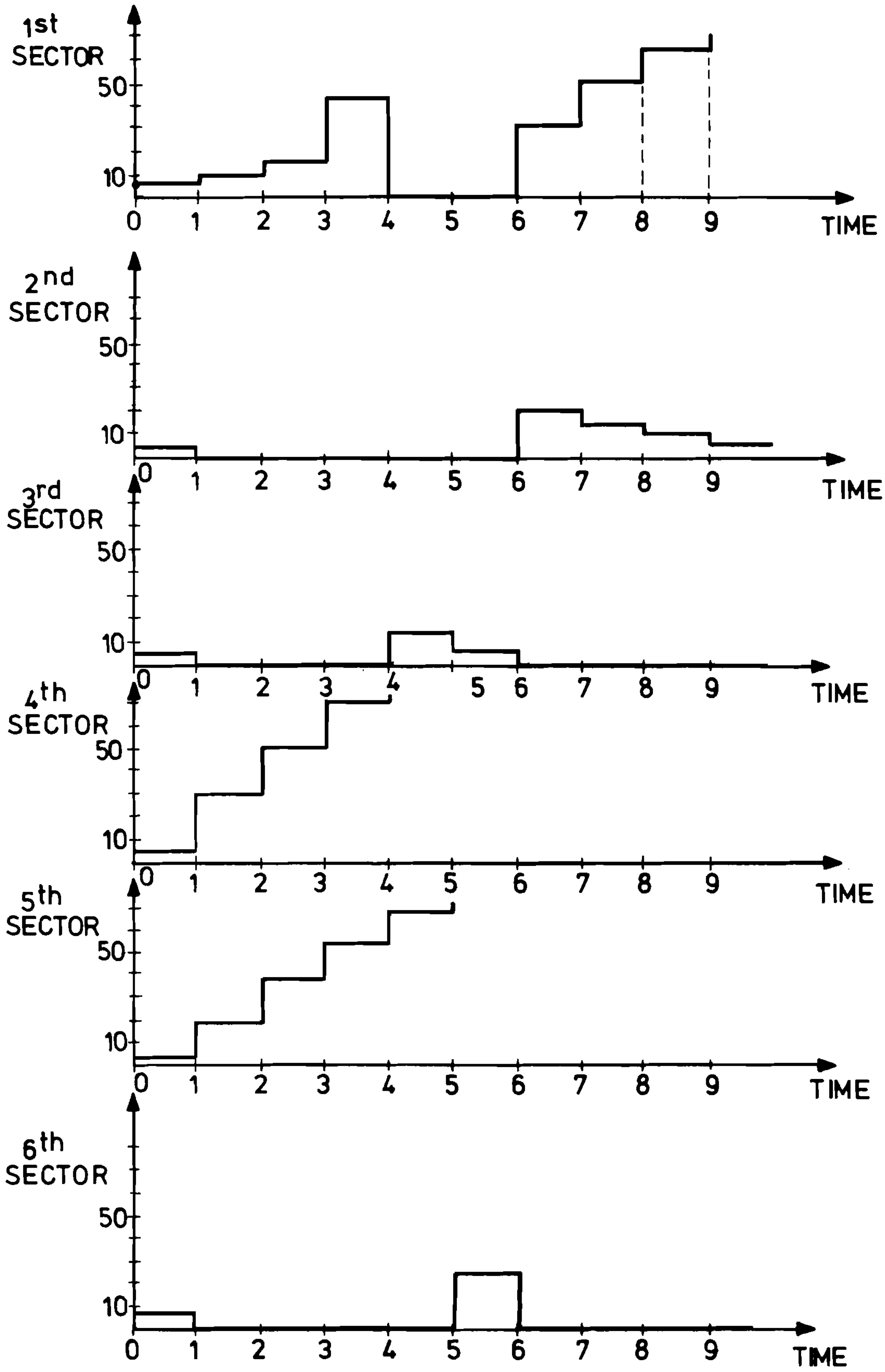


FIGURE 12. NUMERICAL EXAMPLE: DYNAMICS OF STOCKS (q(t)).

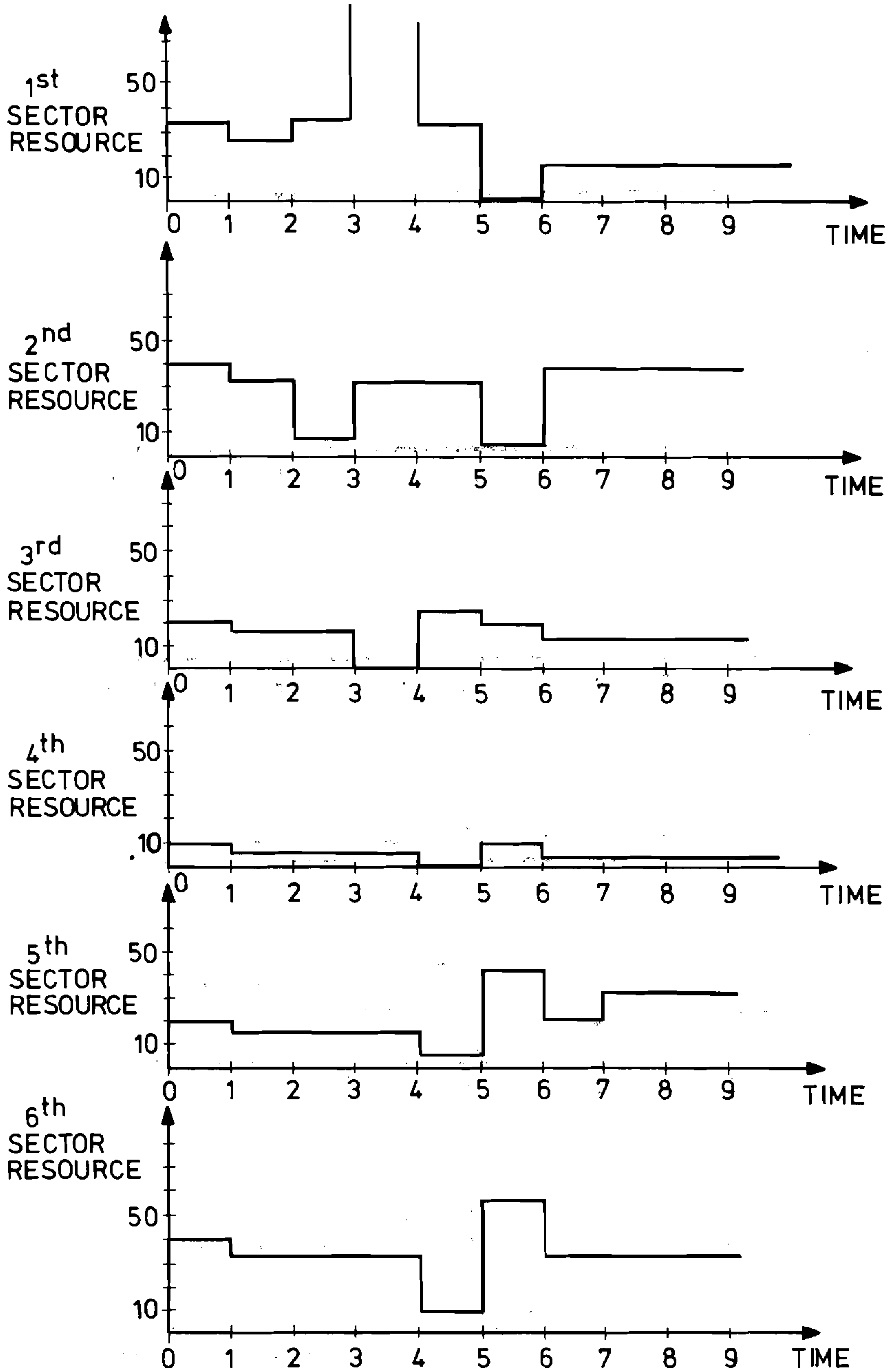


FIGURE 13. NUMERICAL EXAMPLE: RESOURCE SUPPLY OF GENERAL PURPOSE AND SUPPORT PROGRAMS (τ (t))

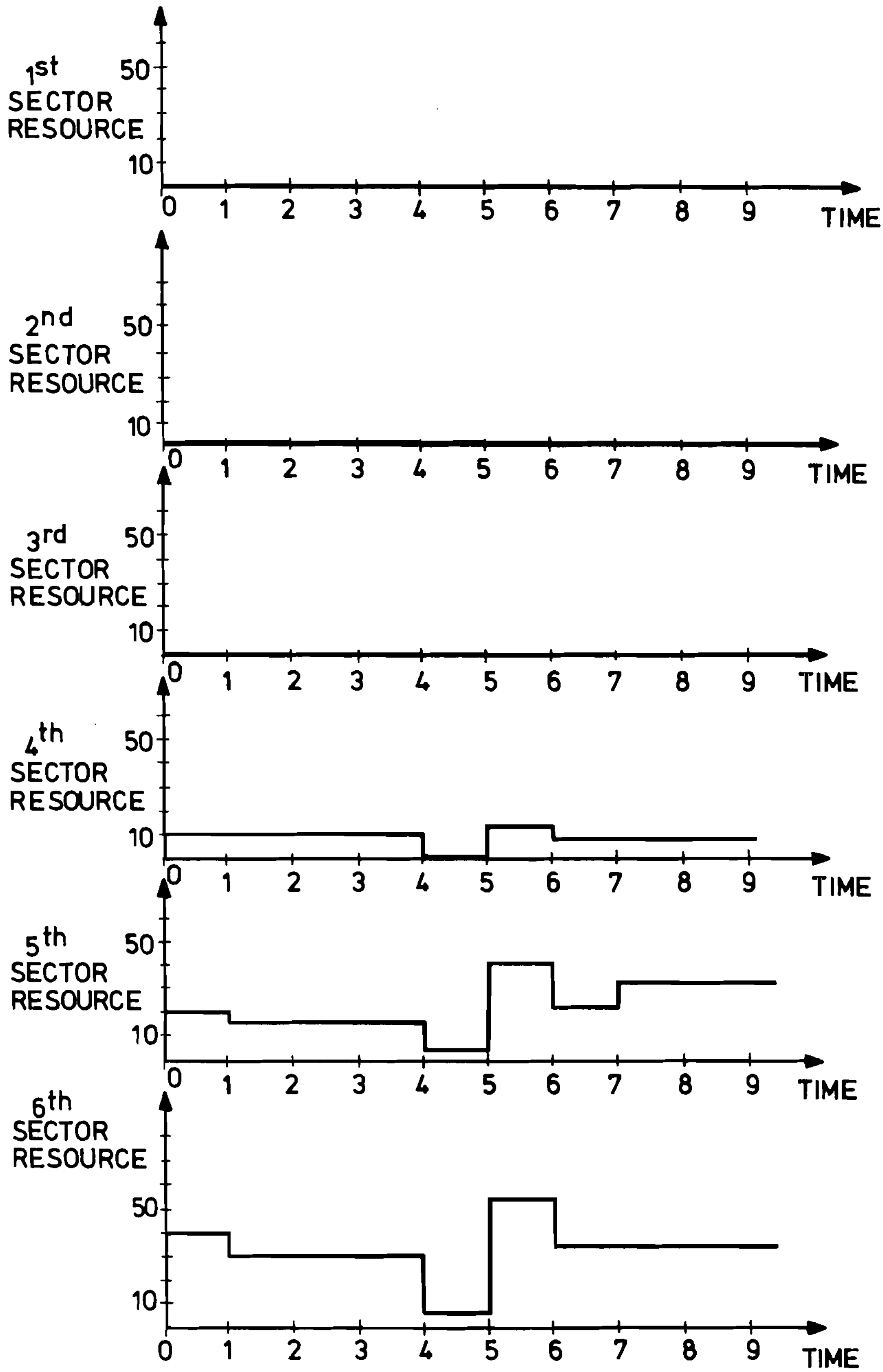


FIGURE 14: NUMERICAL EXAMPLE:
GENERAL PURPOSE PROGRAM RESOURCE CONSUMPTION (R * u(t))

D. Computer Network Optimization Problems

The approach presented in this paper can be applied to the solution of various optimization problems concerning computer network control. Here we consider only two of them.

The first problem is the construction of an optimal queue for computers with a given number of tasks and waiting for the performance on the computer.

Let us consider the case where only one computer is available to perform N tasks. The execution of tasks can be considered as a dynamic process. To describe this process we have to give concrete expressions for some variables and constraints on the program performance model in Section 2.

Again we denote the percentage of the i-th task performed up to moment t by $z^i(t)$, and the intensity of its performance at instant t by $u^i(t)$. $u^i(t) = 0$ if the i-th task is waiting for the computer central processor at instant t, and $u^i(t) > 0$ if the i-th task is in progress. Thus the task processing equations are

$$\dot{z}^i = u^i \quad ,$$

$$z^i(0) = 0 \quad , \quad i = 1,2,\ldots,N \quad . \tag{64}$$

The performance of the tasks is subject to the limited computer capacity. We consider the computer as a resource required to perform tasks. In the more general case the computer could be considered as a set of different kinds of limited resources (card readers, disks, central processors, tapes, line printers, etc.) required to perform the tasks. Such consideration does not make the problem more difficult or lead to a non-essential increase of the problem dimension.

Thus (β) or resource constraints in the model are the following:

$$\sum_{i=1}^{N} t_i u^i(t) \leq 1 \quad , \tag{65}$$

where t_i = nominal duration of processing the i-th task with no preemption. If we have several computers with the same operational characteristics, we have to substitute their total number in the right-hand side of inequality (65). We assume that processing of each task can be interrupted at every instant. This corresponds to the time-sharing regime of computer functioning. On-line regime is realized if interuption of each task is not allowed.

The objective function is total losses (expressed in monetary units) due to idling time for tasks in the queue:

$$I(u) = \int_0^T \sum_{i=1}^N k_i(1 - z^i(t))dt \tag{66}$$

For simplicity, losses are assumed to be the linear function of the idling time for each task; k_i is the penalty per unit of idling time for the i-th task. The process is considered over the finite time interval [O, T].

Thus the problem of optimal queue construction is to obtain the sequence of task performance (schedule u(t)) which minimizes the total cumulative losses (66). Numerical solution of the problem can be obtained by using the algorithm of Section 4 without any change.

In the second type of problem which we are considering here the performance of each task has two stages:

a) processing at the switching node for data transference (processing time of the i-th task is equal to d_i time units);

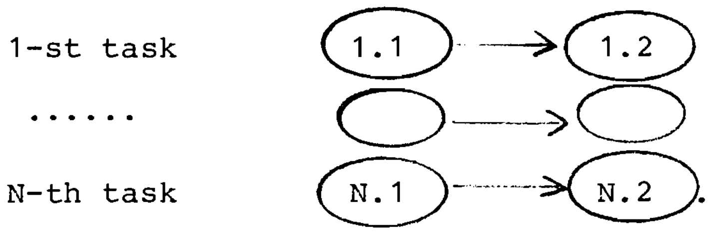
b) processing at the central processor of the computer.

The problem is to construct the schedule of task performance which minimizes total cumulative losses over the given time interval:

$$I(u) = \int_0^T \sum_{i=1}^N (k_i^1\theta(1 - z_1^i(t)) + k_i^2\theta(1 - z_2^i(t))dt \quad , \tag{67}$$

where the first item in (67) is losses due to waiting in the switching node queue and the second is losses due to waiting in the computer buffer storage.

Here we have divided each task into two subtasks. The first is data transference (variable z_1^i) and the second is computation in the central processor (variables z_2^i). The precedence network diagram is the following:



The dynamic equations and resource constraints for subtask performance are as follows

$$\dot{z}_1^i = u_1^i \quad , \qquad z_1^i(0) = 0 \quad , \tag{68}$$

$$\dot{z}_2^i = u_2^i \cdot (z_1^i(t) - 1) \quad , \qquad z_2^i(0) = 0 \quad , \tag{69}$$

$$\sum_{i=1}^{N} d_i u_1^i(t) \leq 1 \quad , \tag{70}$$

$$\sum_{i=1}^{N} t_i u_2^i(t) \leq 1 \quad . \tag{71}$$

The problem can be easily generalized by incorporating into the model several computers, several switching nodes with different characteristics and various interconnections between switching nodes and computers.

Our purpose in Section 7 was only to demonstrate applicability in principle of the scheduling optimization techniques presented to problems in different areas of human activity. Details in problem statements and realization of these methods in practice would require additional efforts.

Acknowledgement

I would like to thank Professors N.N. Moiseev, Yu.P. Ivanilov and A.A. Petrov, who provided me with useful comments during my work at the Computing Centre of the USSR Academy of Sciences and Moscow Physico-Technical Institute, for their quidance and assistance, and all IIASA scientific staff members who took part in discussions on the subject presented.

References

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

[2] Belyaev, I.P., and Zimin, I.N.,"Computer Assisted Procedures in Water Project Evaluation." IIASA RR-75-27 . Laxenburg, Austria, International Institute for Applied Systems Analysis, 1975.

[3] Bennington, G.E., and McGinnis, L.F., "A Critique of Project Planning with Constrained Resources." Lecture Notes in Economics and Mathematical Systems. Symposium on the Theory of Scheduling and its Applications (ed. Elmaghraby, S.E.), Springer-Verlag, Berlin, Heidelberg, New York,(1973), 1-28.

[4] Bowman, E.H., "The Schedule-Sequencing Problem," Operations Research 7, September 1959.

[5] Cleland, D.I., and King, W.R., Systems Analysis and Project Management. New York, McGraw-Hill Book Company 1969.

[6] Conway, R.W., Maxwell, W.L., and Miller, L.W., Theory of Scheduling, Addison-Wesley, 1967.

[7] Elmaghraby, S.E., "The Sequencing of n Jobs on n Parallel Processors with Extensions to the Scarce Resources Problem of Activity Networks," Proceedings of the Inaugural Conference of the Scientific Computation Center and the Institute of Statistical Studies and Research, Cairo, (1969), 230-255.

[8] Fisher, M.L., "Optimal Solution of Resource Constrained Network Scheduling Problems," Technical Report No. 56, Operations Research Center, MIT, September, 1970

[9] Giffler, B., and Thomson, G.L., "Algorithms for Solving Production-Scheduling Problems," Operations Research 8, No. 4, July 1960.

[10] Ivanilov, Yu.P., Moiseev, N.N., and Petrov, A.A., "On Mathematical Methods in the Theory of Programmed Control of an Economic System." In Cybernetics in the Service of Communism, 6, (1970), 9-22, (in Russian).

[11] Ivanilov, Yu.P. and Petrov, A.A. "The Problem of Optimization in Dynamic Multi-Branch Industrial Models." Lecture notes in Mathematics, Vol. 112, Colloquium on Methods of Optimization, Berlin, Springer, 1970.

[12] Kelley, J.F., "Critical Path Planning and Scheduling: Mathematical Basis." Operations Research, May-June 1969.

[13] Schrage, L.E., "Solving Resource Constrained Network Problems by Implicit Enumeration-Non premtive case." Operations Research, 18, 263-278 (1970).

[14] Malcomb, D.G., Rosenboom, J.H., Clark, C.E., and Fazar, W., "Applications of a Technique for Research and Development Program Evaluation," Operations Research, 7 219-223 (1959).

[15] Manne, A.S., "On the Job-Shop Scheduling Problems," Operations Research, 7, 219-223 (1959).

[16] Moder, J.J., and Phillips, C.R., Project Management with CPM and PERT, Reinhold Corporation, New York, 1964.

[17] Moiseev, N.N., Mathematics-Control-Economics. Moscow, Knowledge, 1970. (in Russian).

[18] Moiseev, N.N., "Numerical Methods in Optimal Control Theory." Moscow, Nauka, 1971.

[19] Moiseev, N.N., Belyaev, I.P., and Koryavov, P.P., "The Simplest Model of a River Basin" in System Analysis Methods for the Problems of Rational Use of Water Resources, IIASA, Committee of Systems Analysis of the USSR, (ed. Moiseev, N.N.), 1974.

[20] Nepomiastchy, P., "Application of Optimal Control Theory and Penalty Function Techniques to Solution of a Particular Scheduling Problem." ZVM and MF, No. 4 1970.

[21] Pospelov, G.T., "On the Problem of the Programmed Method for Control Multibranch Production." In Programmed Control Methods, 1971 (in Russian).

[22] Pritsker, A.A.B., Watters, L.J., and Wolfe, P.M., "Multi-project Scheduling with Limited Resources: A Zero-One Programming Approach," Management Science, 16 (1969) 93-108.

[23] Quade, E.S., "Systems Analysis Techniques for Planning-Programming-Budgeting." Planning-Programming-Budgeting: A Systems Approach to Management. Chicago, 1969.

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

[25] Wagner, H., "An Integer Programming Model for Machine Scheduling." NRLQ, 6, 131-139 (1959).

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

[27] Zimin, I.N., and Surguchev, G.D., "Short-Term Planning of an Integrated Industrial Complex." IIASA RM-75-13, Laxenburg, Austria, International Institute for Applied Systems Analysis, 1975.

[28] Zimin, I.N., "On Some Optimal Control Problems Arising from Project Management." IIASA-WP-75-6, Laxenburg, Austria, International Institute for Applied Systems Analysis, 1975.