

## 9 GRAPH AND NETWORK THEORY

### 9.1 APPLICABILITY OF NETWORK ANALYSIS METHODS

The mathematical theory of graphs is the theoretical basis of network analysis methods used in problems of sequential processes (Alexy, 1967). It is an extensive and highly developed theory of finite mathematics, including linear and non-linear programming as a group of problems.

Historically, the oldest task is the problem of the shortest path, which was the origin of some modifications of the transportation problem (transportation network). The applicability of these methods and their implementation is now much broader.

Let us assume a process that consists of partial processes or, stages with their qualitative and quantitative evaluation. Assume that these partial processes have a defined linkage in real time, which may, but need not be serial and direct for each partial process. Therefore indirect linkage in a combination of serial and parallel relationships exists in a group of partial processes (stages). The objective of solving problems of such complex linkage of processes in time and space is frequently a maximum reduction in total processing time.

Such a formulation is often encountered in problems of organization and scheduling of the construction of complex objects and systems, in the management of investment activities, in planning delivery and the set-up of intricate facilities, in planning maintenance, replacement, research and technology developments.

Very often problems of parametric linear programming with methods of flow in networks, mainly the maximum flow in networks with deterministic or stochastic inputs are applied.

In technical and economic activity some parts of the theory and the application of these methods and their modifications are called CPA: *Critical Path Analysis*; what they have in common is the set-up of networks and their analysis.

The *Critical Path Method* (CPM) can be used if these complex activities can be split up into partial activities. It is applied in planning the replacement times of large-scale and complex production units, and for the organization of building large-scale systems.

*Program Evaluation and Review Technique* (PERT) is also an alternative to the method of critical path analysis; it was successfully used for planning large-scale development and construction programmes. Its first modification was developed for time scheduling of these programmes and was called PERT/TIME, a second

modification called PERT/COST is used for planning the costs of the realization of these programmes and projects.

Together with these two basic methods, the method of *Resource Allocation and Multi-Project Scheduling* (RAMPS) is also used. The allocation of limited resources in space and time in different branches (WRS included) is, however, designed by different methods, e.g., by dynamic programming with relatively more possibilities than by methods derived from linear programming.

## 9.2 BASIC TERMS OF GRAPH THEORY

The *node* (vertex) is an isolated point in a plane or in space, or it can be the starting, or ending, point of an arc in a graph. The *arc* (edge) is any curve connecting various pairs of nodes. It does not include any node. The *graph* is a geometric form consisting of a set of points called nodes (vertices) and set of line segments joining any two nodes (vertices) and called arcs (edges). These arcs can be the segments of a line or of a curve.

In the theory of graphs, the definition of a graph  $G$  is possible without its geometrical representation as a diagram (as used in this chapter) by the set of nodes  $N$  and a set of arcs  $G \equiv (N, A)$ .

In the graph theory different kinds and types of graphs are investigated and only a part of them, under certain conditions, are useful for the schematic representation of WRS or critical path analysis of complex processes.

The *null graph* consists of a finite number of nodes that are isolated, i.e., they are not connected by arcs.

The *simple graph* is a graph that has neither self-loops (i.e., arcs having the same node as both their end nodes) nor parallel arcs (i.e., more than one arc associated with a given pair of nodes). The graph that is not simple is called a general graph (multi-graph).

The *complete graph* is a simple graph in which every pair of nodes is connected by an arc.

The *directed graph* (*digraph*, orientated graph) is a graph where every arc is represented by a line segment between the node  $n_i$  and  $n_j$  with an arrow directed from  $n_i$  to  $n_j$ .

Two graphs  $G_1$  and  $G_2$  are called *isomorphic* if there is a one-to-one correspondence between their nodes and arcs, such that a pair of nodes in the graph  $G_1$  are joined by arcs, if and only if, the corresponding pair of nodes in the graph  $G_2$  are also joined. In directed graphs the orientation is also maintained.

*Planar graphs* have some geometric representation which can be drawn on a plane in such a manner that no two of its edges intersect (they only meet in the node). When a node  $n_i$  is an end (or starting) node of an arc  $a_j$ ,  $n_i$  and  $a_j$  are said to be *incident* to each other. The number of arcs incident to a node is called the degree of this node.

The number of arcs is then equal to half the sum of the degrees of all the nodes.

The graph is termed *connected* if there is at least one chain between every pair of nodes in this graph. The *chain* (walk) is a broken line in the graph, such that no arc is traversed more than once.

A closed chain is called a *cycle*. A cycle in which no node (except the initial and final node) appears more than once is called an *elementary cycle* (circuit).

If a closed chain in a graph traverses every arc of the graph exactly once, the chain is called a *Euler line* (named after Euler, who in 1736 developed the basis of the graph theory). Each graph that contains this line is called a *Euler graph*, and it has some special properties (e.g., it is connected).

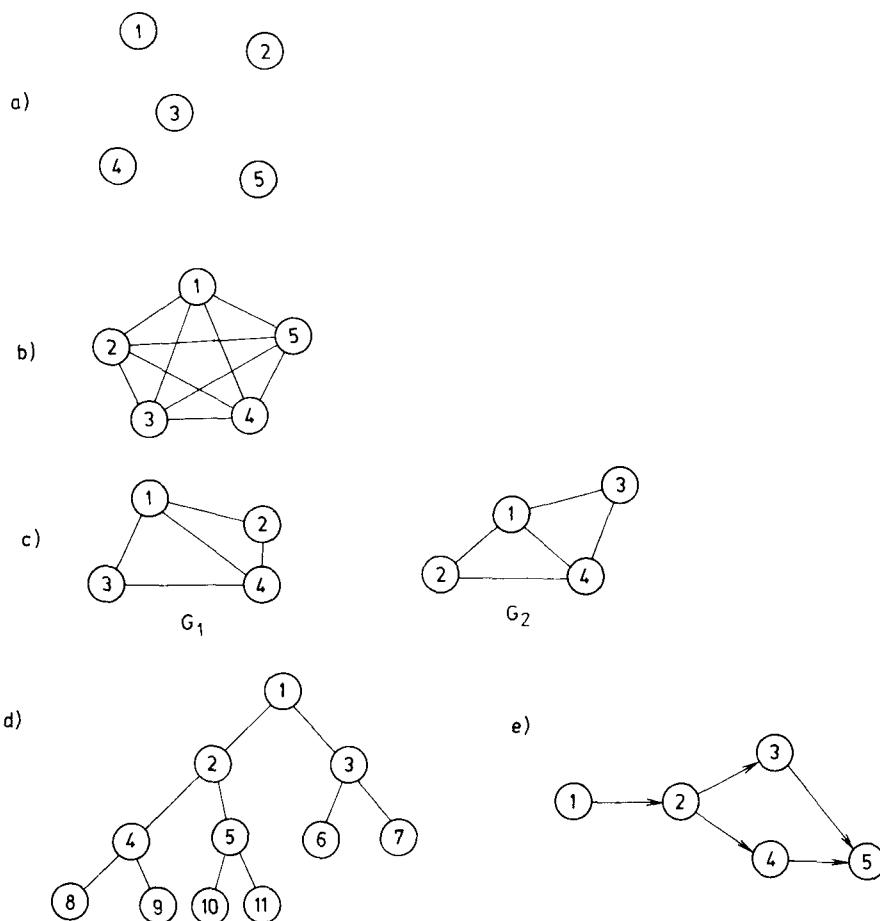


Fig. 9.1 Examples of some types of graphs

*a* – a null graph, *b* – a complete graph, *c* – isomorphic graphs  $G_1$  and  $G_2$ , *d* – a tree, *e* – a directed graph (digraph)

If the chain traverses every node of the graph, and if this chain is an elementary cycle, it is called the *Hamiltonian circuit*.

A connected graph without any cycle is called a *tree*. A tree with  $n$  nodes has  $(n - 1)$  arcs. If the tree is not directed, it can start in any node — each node of the graph can be the *root* of the tree. The remaining components of a tree graph form the *arborescence*.

Sometimes a combination of cycles and trees is encountered. Then the following task can be carried out: the determination of the number of arcs that should be dropped, in order to exclude all cycles. This number is called the *cyclomatic number* or *nullity* of the graph.

Further, the directed graphs will be investigated. The types of some graphs are given in Fig. 9.1.

### 9.3 GRAPH MODELS OF COMPLEX PROCESSES AND THEIR REPRESENTATION

For the CPA methods that are often applied to WRS, some of the basic terms of the general graph theory are important. An important aspect is the complex character of these processes, which require not only qualitative but also quantitative representation of the problem dealt with. Therefore directed graphs with some valuation of arcs are often applied.

Instead of the term vertex that has been derived from the geometrical representation, in application (and often in theory) the term *node* is used, or *time node*, as it often denotes the time of a phenomenon or event when one or several part processes begin or come to an end. A unique and non-ambiguous definition of the node is necessary.

Instead of arc or edge the term '*activity*' is often used; however, this term is sometimes ambiguous and bearing in mind the character and formulation of tasks it is not clearly defined.

The *graph* as a set of nodes (time nodes) and arcs (activities) is set up under some specific conditions and assumptions, which include in the first place:

- the orientation of the arcs, which is given by the direction of the partial process (stage, activity), proceeding from the starting node to the ending node;
- the valuation of arcs, which assumes not only a qualitative, but also quantitative characteristics of the arcs (the duration of the activity, costs of this activity, etc.).

If the first condition is met, the graph is called the *directed graph* (digraph) in a narrower sense. If it meets the second condition the graph is called the *valuated* (edgevaluated) *graph*. The orientation and valuation of the graph is the vectorization of all activities of a complex process.

Besides edge-valuation node-valuation is sometimes performed, and the graph is called a *node-valuated graph*. In principle it is an exchange of the function of nodes

and edges. Such an approach may sometimes be more advantageous. The node can then represent an activity, it is valued by the duration of the activity, its costs, etc.

In the generation of a graph representing a complex process the following rules have to be obeyed:

- each activity needs to have a uniquely defined starting and terminating point,
- no activity starting in one node can start before all the activities ending in this node have been completed,
- the graph can have only one initial and one terminal node; the initial node is a node in which no activity ends, the terminal node is a node in which no activity starts,
- two nodes can be connected by one activity only (this is not valid in a multi-graph),
- there is no sequence of partial processes (activities) following each other, which starts and ends in the same node; therefore no cycle (or loop, feedback) can appear in the graph.

These conditions are used for an edge-valued digraph (directed graph); some corrections would be necessary for a node-valued graph.

A network prepared for the solution of problems can be represented by data, presented in a tabular form, on the nodes and arcs, their connection, orientation and valuation in a matrix of the network called the *incidence matrix*. Computer processing of the problem cannot be done without this matrix. Some very complicated processes use this convenient and useful matrix representation, as the graphical representation is cumbersome and can be a source of errors.

In Fig. 9.2 there is a simple general scheme of this matrix for the network in Fig. 9.3.

$t_i^{(0)}$	$j$	0	1	2
$t_0^{(0)}$	0		$\tau_{01}$	$\tau_{02}$
$t_1^{(0)}$	1			$\tau_{12}$
$t_2^{(0)}$	2			
$t_j^{(1)}$		$t_0^{(1)}$	$t_1^{(1)}$	$t_2^{(1)}$
$t_j^{(1)} - t_i^{(0)}$				

Fig. 9.2 The scheme of the incidence matrix of a simple network

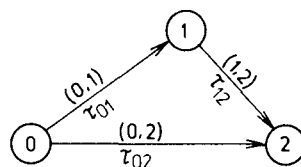


Fig. 9.3 The scheme of a simple network

The graph has three nodes and three arcs. In the first column the earliest possible starting points of activities  $t_i^{(0)}$  (earliest event time) are listed.

Similarly in the penultimate row of the matrix the latest event times of activities  $t_j^{(1)}$  and in the last row the differences  $t_j^{(1)} - t_i^{(0)}$  are listed, which are important for the computation of the total slack (see 9.4.1).

In the body of the matrix (in Fig. 9.2 inside the heavy lines) there are the non-zero elements in the cross-section of rows denoting the starting nodes and columns denoting the terminating nodes of arcs of this network. Each row of the matrix represents the valuation of this arc and in the critical path analysis it is the duration of the activities.

The calculation of the values  $t_i^{(0)}$  and  $t_j^{(1)}$  and of the total slack is done in section 9.4.1. In this matrix the earliest event times (in the upper right hand corner) and the latest event times (in the lower left hand corner) are often listed.

## 9.4 TIME ANALYSIS OF NETWORKS

### 9.4.1 Critical Path Analysis

The first stage of the time analysis by the method Critical Path Analysis (CPA) is the development of the network represented by a graph, its orientation, and valuation. It requires proper definition and notation of activities, their specification and description. For each activity the following items are listed (usually in tabular form):

- all the preceding activities,
- the estimation of the normal duration of the activities, (possibly their standard values).

The shortest time to complete all the activities consecutively is given by the longest directed path (walk) in the network called the *critical path*. The determination of the critical path consists of the determination of the earliest (expected) starting time and the earliest finishing time of each activity and the latest allowable starting time and the latest allowable finishing time, which are necessary for completing the process plan (complex activity). Their values are often listed directly in the network.

The computation of the earliest starting times is the aim of the forward pass starting from the initial project event (node) of the network in the direction of the orientation of the digraph (logical sequential activity dependencies). The earliest starting time is given by the maximum of the earliest finishing times of all preceding activities.

The latest allowable finishing time is the minimum of all latest allowable starting times of the successors of this activity, and it is determined by the backward pass calculation, beginning with the project terminal event (node).

The critical path is then the sequence of the dependent activities where the difference between the earliest starting time and the latest allowable starting time (or finishing time) is zero.

Each increase of the duration of the critical activities (activities on the critical path) will delay the entire project. If a reduction of project time is required, a reduction of the duration of critical activities is necessary. If the critical activities follow immediately upon each other, then the non-critical activities have some *time reserve*.

The positive difference between the earliest expected and the latest allowable starting time form a time reserve called the *total activity slack*. Its use can reduce the time reserve of the succeeding activities. Therefore the total activity slack should be used by the planner who is responsible for the whole project.

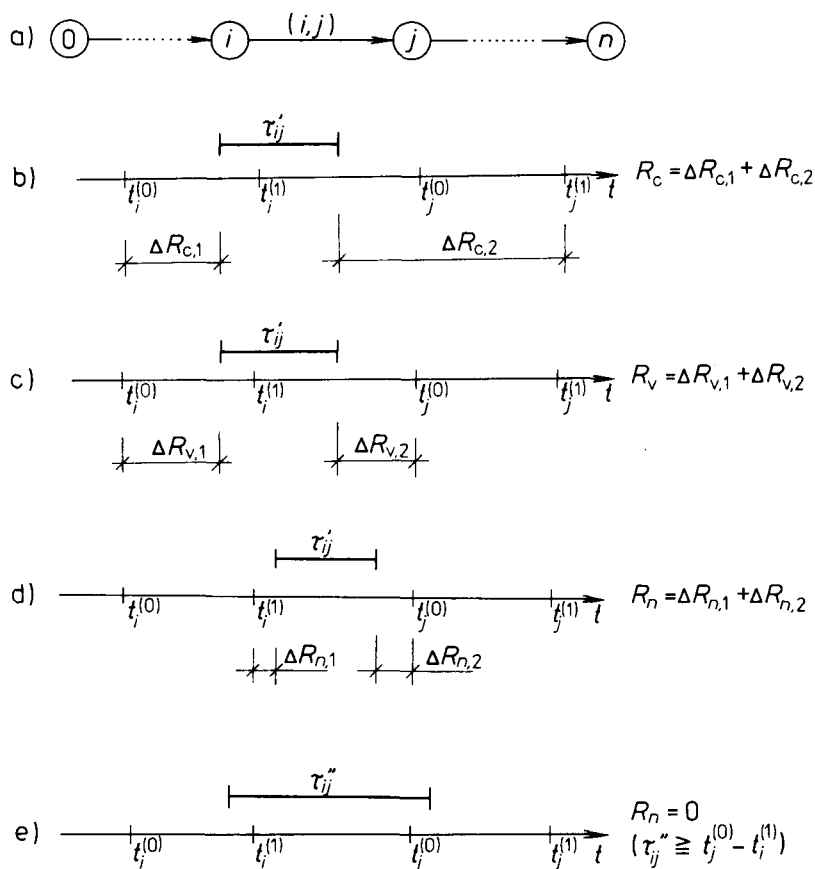


Fig. 9.4 The representation of activity slacks in analysis of the network by CPM  
 a – scheme of the activity (i, j), b – total activity slack, c – free activity slack, d – activity independent slack for  $\tau'_{i,j} < t_i^{(0)} - t_i^{(1)}$ , e – activity independent slack for  $\tau''_{i,j} \cong t_j^{(0)} - t_i^{(1)}$

There is a part of the total activity slack that does not influence the earliest starting time of the succeeding activities. This part is called *activity-free slack*. It can be used by the planner responsible for the individual activities as the use of this activity cannot influence the succeeding activities.

The activity-independent slack does not depend on the use of the preceding activities, and therefore it can be used independently.

The method outlined for the determination of the critical path and the activity slack and the rule of their use can be described mathematically (Fig. 9.4a), where  $i$  is the starting node of the activity  $(i, j)$ ;  $j$  is the ending node of the activity  $(i, j)$ ;  $\tau_{i,j}$  is the estimate of the mean (standard) duration time for the activity  $(i, j)$ ;  $t_i^{(0)}$  is the earliest starting time for the activity  $(i, j)$ ;  $t_i^{(1)}$  is the latest allowable starting time for the activity  $(i, j)$ ;  $t_j^{(0)}$  is the earliest finishing time for the activity  $(i, j)$ ; and  $t_j^{(1)}$  is the latest allowable finishing time for the activity  $(i, j)$ .

The earliest finishing time for the activity is given by

$$t_j^{(0)} = \max (t_i^{(0)} + \tau_{i,j}) \quad \text{for } n \geq j \geq 1 \quad (9.1)$$

If the start of the activity  $t_i$  is the initial point of a complex activity (process), then

$$t_i = t_i^{(0)} = t_i^{(1)} = 0$$

The activities  $(i, j)$  are elements of the set of feasible activities  $P$ , i.e.  $(i, j) \in P$ . The relationship between the termination time of a complex activity (of the whole process)  $\lambda$  and the earliest finishing time  $t_n^{(0)}$  has the three forms:

a)  $\lambda = t_n^{(0)}$  and also  $\lambda = t_n^{(1)}$  when the capacity of the system meets the system requirements exactly;

b)  $\lambda > t_n^{(0)}$  where there are time limits for the whole process and also for the individual activities, which are not specified in detail;

c)  $\lambda < t_n^{(0)}$  where the system cannot meet the required termination time and in the project of the complex activity the transition to the case a) or b) should be achieved to obtain  $t_n^{(0)} \leq \lambda \leq t_n^{(1)}$ .

Assume that the capacity of the system meets the requirements as far as the terminal time is concerned, i.e.,  $t_n^{(1)} = \lambda$ . Then the latest allowable starting time of any activity is given by:

$$t_i^{(1)} = \min (t_j^{(1)} - \tau_{i,j}) \quad \text{for } n - 1 \geq i \geq 0 \quad (9.2)$$

and the activity  $(i, j)$  can be allocated arbitrarily in the time interval  $\langle t_i^{(0)}; t_j^{(1)} \rangle$ . The parts of the time reserve  $R$  can have a different meaning, and they can be used for different goals inside the time reserve.

Two cases occur: a)  $R = \tau_{i,j}$ , and b)  $R > \tau_{i,j}$ .

In case a) the equality of the time reserve  $R$  and the duration time of the activity  $\tau_{i,j}$  is the proof that this activity is on the critical path.

It can be proved that for  $t_n^{(0)} = \lambda$  the network contains at least one critical path composed of activities where the condition  $t_j^{(1)} - t_i^{(0)} = \tau_{i,j}$  is met, i.e., at least

one critical path. For the activities on this path it is essential that  $t_j^{(1)} = t_j^{(0)}$  and  $t_i^{(0)} = t_i^{(1)}$ . In case b) when  $R > \tau_{i,j}$  these activities are not on the critical path. The time reserve

$$t_j^{(1)} - t_i^{(0)} - \tau_{i,j} = R_c \quad (9.3)$$

is called the *activity total slack*.

Activity  $(i, j)$  can be arbitrarily allocated inside the interval  $\langle t_i^{(0)}; t_j^{(1)} \rangle$  (Fig. 9.4b). Using the total activity slack  $R_c$ , this activity becomes a critical one: it is on the critical path. This means the development of an additional critical path, which is not desirable.

If the total activity slack ( $R_c = \Delta R_{c1} + \Delta R_{c2}$ ) is not used, but only a part of it given by time intervals  $\Delta R_{v,1}$ ,  $\Delta R_{v,2}$  (Fig. 9.4c), a more convenient state will be developed not affecting the earliest finish time for this activity  $(i, j)$ . This part of the total activity slack

$$t_j^{(0)} - t_i^{(0)} - \tau_{i,j} = R_v \quad (9.4)$$

is called the *free activity slack*, and it does not influence the successors in the network.

If neither the activity's earliest finish time nor the latest allowable start time are not to be influenced, then the activity-independent slack is used, that for  $t_j^{(0)} - t_i^{(1)} > \tau_{i,j}$  is equal (see Fig. 9.4d).

$$R_n = t_j^{(0)} - t_i^{(1)} - \tau_{i,j} \quad (9.5)$$

and for  $t_j^{(0)} - t_i^{(1)} \leq \tau_{i,j}$  is equal to zero (Fig. 9.4e).

Some simple examples are explained by Alexy, 1967.

## 9.4.2 The PERT Method

The PERT method is suitable for the time analysis of complex processes where the time dimension and its exploitation is the most important feature. In this method two basic interrelated components appear: the planning and the control components.

The basic methodological approach to the treatment of complex problems by the PERT method is similar to that by the CPM method: a network is developed, the duration of the activities is estimated and a critical path is sought. The difference between the two methods is in the estimation of the duration times of the activities. In CPM this estimation is deterministic, as it is assumed that the duration time can adequately be characterized by a single value. In PERT these inputs are stochastic: the duration time for the activities in the network is characterized by a probability distribution of the time values. Therefore CPM can be considered to be a special case of the PERT method if the estimates of the expected duration times have a probability equal to one, or, the PERT method can be considered as a generalization of

the CPM method, if the probability of estimation of the mean value is less than one, i.e., there is some uncertainty in its occurrence. The PERT analysis can be considered as a *stochastic method*.

In practical applications of this method the whole probability distribution of the duration time is not used; only some values (usually three) are applied. These values are:

- optimistic (minimum) estimate of the duration time for the activity  $\tau_a$  (in the

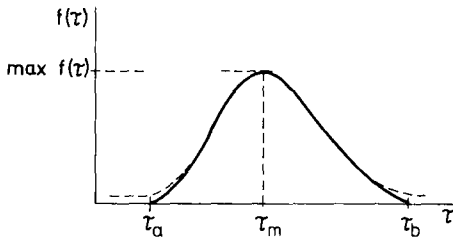


Fig. 9.5 Probability density function of the duration time for the activity in the time analysis of a network by the PERT method

graph in Fig. 9.5) is the lowest value in the case of the lower-bounded probability distribution or the value  $\tau'_a$  given by a very small portion of the area under the probability density function (e.g., 1% or 5%),

- pessimistic (maximum) estimate of the duration time for activity  $\tau_b$  (Fig. 9.5) or  $\tau'_b$  similarly defined as an optimistic value (e.g. 99 or 95%),
- the most likely estimation of the duration time  $\tau_m$  – the modal value of the distribution.

It is assumed that the duration time of activities in a complex process has a normal distribution or a beta distribution<sup>1)</sup>. In the case of a beta distribution, the probability density function can be written in the form

$$f(\tau) = k(\tau - \tau_a)^\alpha (\tau_b - \tau)^\gamma \quad (9.6)$$

where  $k$  is a constant and  $\alpha$  and  $\gamma$  are functions of  $\tau_a$ ,  $\tau_b$  and  $\tau_m$ .

The value used in computation is estimated by

$$\tau_0 = \frac{\tau_a + 4\tau_m + \tau_b}{6} \quad (9.7)$$

<sup>1)</sup> The beta distribution is defined by the cumulative distribution function, which uses the Euler integral of the first kind

$$B_x(\alpha, \beta) = \int_0^x x^{\alpha-1} (1-x)^{\beta-1} dx$$

with two parameters  $\alpha$ ,  $\beta$ . For some combinations of  $\alpha$ ,  $\beta$  it is similar to the Pearson type I distribution (see 3.1.2.7).

with variation

$$\sigma^2(\tau_0) = \left( \frac{\tau_b - \tau_a}{6} \right)^2 \quad (9.8)$$

This method is used for processing all input data of the activities in time analysis. The further procedure is similar to that used in CPM.

The earliest expected and latest allowable activity starting times  $t_i^{(0)}$  and  $t_i^{(1)}$  and their variances  $\sigma^2(t_i^{(0)})$  and  $\sigma^2(t_i^{(1)})$  are computed. The finishing time of the whole process can be given by a deterministic value, otherwise it is unknown. In the latter case it is made equal to the latest allowable time of the latest activity of the complex process in the critical path  $t_n^{(1)}$ .

A specific feature of the PERT method is the determination of slack probability at the end of the complex process. It includes assessing the probability of completing the project on target with the probability estimation that does not appear in CPM.

Let us analyse the duration of the activities. The value  $x$  is given by

$$x = \frac{t_j^{(0)} - t_i^{(1)}}{\sqrt{\sigma^2(t_j^{(0)}) + \sigma^2(t_i^{(1)})}} \quad (9.9)$$

where  $t_j^{(0)}$  is the estimation of the earliest finishing time for the activity,  $t_i^{(1)}$  is the estimation of the latest allowable finishing time for the activity, and  $\sigma^2(t_j^{(0)})$ ;  $\sigma^2(t_i^{(1)})$  are the respective variations.

The value  $x$  gives a relative deviation from the negative value of the total activity slack for this activity, i.e., from the difference  $t_j^{(0)} - t_i^{(1)}$ . To  $x$ , a probability of  $F(x)$  can be assigned from the cumulative distribution function that is equal to the probability of the activity slack. Using these values, the slack for the whole process can be determined.

Three cases can be classified in the practical applications of the PERT method:

- a)  $F(x) \leq 0.25$ , then the time target (the duration of the whole project within a given time limit) is endangered,
- b)  $0.25 < F(x) \leq 0.60$  with a realistic assumption of meeting the time target,
- c)  $F(x) > 0.60$  with a time reserve.

Comparison of the two basic methods for the critical path analysis of the network shows that the PERT method which makes the development of a stochastic model possible is a more detailed and sophisticated tool of the network analysis with a more general scope of application.

### 9.4.3 Other Methods of Critical Path Analysis

All other methods of critical path analysis (CPA) are modifications of the CPM and PERT methods.

If there are a great number of activities with a complicated dependence, the development of the diagram of the network may be tedious and cumbersome. Then the matrix representation is often used with better insight and prevention of possible errors. The calculation of the incidence matrix is usually done by computer. The critical path programs are often provided in standard computer software.

The CPA can also be carried out by the simplex method of linear programming. The CPM is related to the transportation problem, and it can be used for the search for the Hamiltonian path as well as the travelling salesman problem.

Linear programming with the CPM method is used for decisions about the reduction of activity times, which are not on the critical path where a suboptimal solution can cause a non-effective use of funds.

There are programs for the optimization of postponing the starts of some activities under budgetary constraints to achieve the minimum duration of the whole process.

A general advantage of the CPM is the possibility of adapting to new conditions: to re-run alternatives on computers using new parameters, to new formulations of the model, and changes in the network. This advantage is used in practice and, in addition, it has some theoretical significance: the model is more sophisticated, and it can react flexibly to dynamic changes in time. These methods facilitate effective and flexible decision-making.

These methods are not suitable, however, for long-term planning. The general problem common to all methods of CPA is collecting reliable input data.

The analysis of the possibilities of reducing the duration of the whole process is done by a method somewhere between the deterministic CPM and the stochastic PERT method. In practice it means a flexible increase of funds for particular activities on the critical path in order to achieve the necessary reduction in time.

Then the duration time  $\tau_{i,j}$  of an activity  $(i, j)$  is not characterized by one constant value, but by two independent variables:

$$\begin{aligned} \vartheta_{i,j} &= \text{minimum duration for the activity } (i, j), \text{ and} \\ \Theta_{i,j} &= \text{maximum duration for the activity } (i, j). \end{aligned}$$

Therefore:

$$0 \leq \vartheta_{i,j} \leq \tau_{i,j} \leq \Theta_{i,j} \quad (9.10)$$

The value  $\Theta_{i,j}$  is often called the expected (standard) duration for the activity (under standard conditions) using minimum costs. It is possible to reduce this value to as low as  $\vartheta_{i,j}$  by the development of better conditions and at greater costs.

The reduction of the duration of the whole process is accompanied by the require-

ment of minimum costs. For each activity  $(i, j)$ , a cost curve  $N_{i,j} = f(\tau_{i,j})$  is constructed, which is generally a non-linear, convex, and decreasing function. In practical applications it is linearly approximated in the interval  $\tau_{i,j} \in \langle \vartheta_{i,j}, \Theta_{i,j} \rangle$  so that

$$N_{i,j} = a_{i,j}\tau_{i,j} + b_{i,j} \quad (9.11)$$

The cost function of the whole process can be written in the form

$$N_p = \sum_{(i,j)} N_{i,j} = \sum_{(i,j)} (a_{i,j}\tau_{i,j} + b_{i,j}) \quad (9.12)$$

The duration of each activity  $(i, j)$   $\tau_{i,j}$  is a function of the required duration of the whole process  $\lambda$ ; therefore  $N_p = N_p(\lambda)$ . If the constraints  $\tau_{i,j} < \Theta_{i,j}$  and  $\lambda < t_n^{(0)}$  occur, then the duration of some activities has to be curtailed.

Often, there are several choices and an optimum is chosen which requires some increase of total costs for the realization of the complex process. Such an optimization problem is often solved by linear programming.

Minimize the objective function

$$\sum_{i,j} (a_{i,j}\tau_{i,j} + b_{i,j}) = \min \quad (9.13)$$

under constraints

$$\tau_{i,j} \leq \Theta_{i,j}; \tau_{i,j} \geq \vartheta_{i,j}; \tau_{i,j} = t_j - t_i;$$

$$t_0 = 0 \text{ and } t_n = \lambda$$

and for

$$(i, j) \in P$$

where  $i = 0, 1, \dots, n-1$ ;  $j = 1, 2, \dots, n$ , and  $\tau_{i,j} \geq 0$ .

The reduction of the duration of the whole process implies the reduction of those activities where it is most effective.

Assume that the complex process represents a project of some actual system. In real applications the optimization of the reduction of the total duration time is to include losses of production, operation, maintenance and replacement costs, etc., which would occur if the original project duration were assumed.

## 9.5 GRAPH THEORY APPLICATION IN WRS

The methods of network analysis by the graph theory have been extensively used in municipal water supply networks and irrigation network systems. The application of the graph theory with the representation of a complex problem by an incidence matrix is used for the simplification of the mathematical formulation of the task. The electrical network analysis was an original model for this application of this network and matrix analysis.

The representation of the topological structure<sup>1)</sup> of the river network by a graph was investigated by Schneidegger, 1967. The matrix analysis of municipal water supply networks was studied in Czechoslovakia by Šerek, 1968, 1972, and was used by Ošlejšek, 1975.

The topological structure of WRS is modelled by a digraph, the arcs of which represent the water transportation facilities and their valuation represents the capacity of these facilities. The nodes form the points of connection of these arcs, storage volumes, points of beginning and ending of water diversions, mains, etc. The orientation of the digraph is given by the character of the problem or some chosen convention. The incidence matrix represents the topological structure of WRS in an algebraic form that is convenient for algorithmization for computation of the problem.

### 9.5.1 Description and Analysis of River Networks by the Graph Theory and Matrix Analysis

A part of the river network is represented in Fig. 9.6 by a digraph (Šerek, 1972). The river network does not form cycles and its graph is a tree, whose arcs correspond to the branches. The graph is orientated in the direction of flow in the branches.

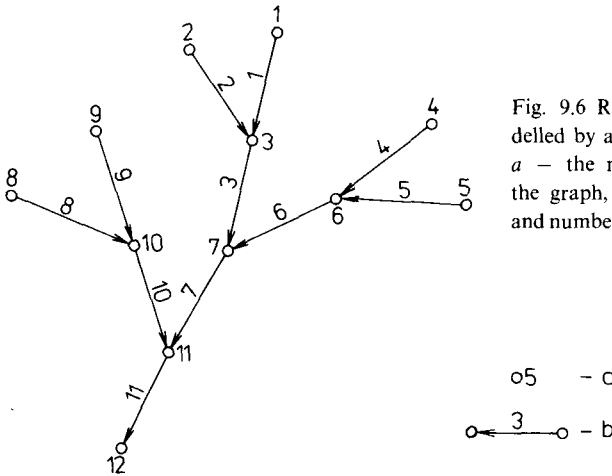


Fig. 9.6 River network modelled by a tree  
*a* – the numbered node of the graph, *b* – the oriented and numbered arc of the graph

<sup>1)</sup> Topological structure is the system structure of an abstract system defined on the WRS in some topological space – in this case on some graph topological space – where the characteristics of the system are invariant to continuous, one-to-one correspondence transformations.

If the graph has  $u$  nodes and  $v$  arcs (branches), then  $v = u - 1$ . As a formality, the terminating node of the graph is not used in the incidence matrix  $\mathbf{A}$  and only  $(u - 1)$  nodes are considered. Each row represents one arc (branch) and each column one node. Then the incidence matrix will be a square matrix ( $v$  rows and  $u - 1 = v$  columns) and a regular<sup>1)</sup> matrix.

For the graph in Fig. 9.6, the incidence matrix has the following form

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (9.14)$$

The elements of the matrix have the following values:

$a_{i,j} = 0$  if the node  $j$  is not the ending point of the  $i$ -th arc;  $a_{i,j} = 1$  if the node  $j$  is the starting node of the  $i$ -th arc<sup>2)</sup>;  $a_{i,j} = -1$  if the node  $j$  is the end node of the  $i$ -th arc.

Let us read the matrix (9.14) row by row: arc 1 starts in node 1 ( $a_{11} = 1$ ) and ends in node 3 ( $a_{13} = -1$ ), arc 2 starts in node 2 ( $a_{22} = 1$ ) and ends in node 3 ( $a_{23} = -1$ ), etc.

Assume the task of flow determination in arcs (branches) of this river network with an assumed stationary flow pattern without storage in the river bed and flood plains. The unknown flows in  $v$  arcs and withdrawals and inflows in  $(u - 1)$  nodes are expressed as column vectors

$$\mathbf{q} = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_i \\ \vdots \\ Q_r \end{pmatrix}; \quad \mathbf{r} = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_j \\ \vdots \\ R_{-1} \end{pmatrix} \quad (9.15)$$

$$(9.16)$$

<sup>1)</sup> In a regular (non-singular) matrix, the rank is equal to the order, i.e., the number of the rows or columns; in a square matrix, the rank is defined.

<sup>2)</sup> In the sense of graph orientation.

where  $Q_i$  is the flow in the  $i$ -th arc,  $R_j > 0$  is the withdrawal from the  $j$ -th node,  $R_j < 0$  is the inflow to the  $j$ -th node. Using the mass balance equation<sup>1)</sup> we get

$${}^T\mathbf{A}\mathbf{q} + \mathbf{r} = \mathbf{o} \tag{9.17}$$

where  ${}^T\mathbf{A}$  is transposed<sup>2)</sup> matrix of matrix  $\mathbf{A}$ ,  $\mathbf{q}$ ,  $\mathbf{r}$  are the column vectors defined by (9.15), (9.16), and  $\mathbf{o}$  is a zero vector. The flow vector  $\mathbf{q}$  is given by the relationship

$$\mathbf{q} = -\mathbf{r}{}^T\mathbf{A}^{-1} \tag{9.18}$$

where  ${}^T\mathbf{A}^{-1}$  is the inverse<sup>3)</sup> matrix of the matrix  ${}^T\mathbf{A}$ .

Reservoirs can be added to the river network described (Šerek, 1972) in Fig. 9.7.

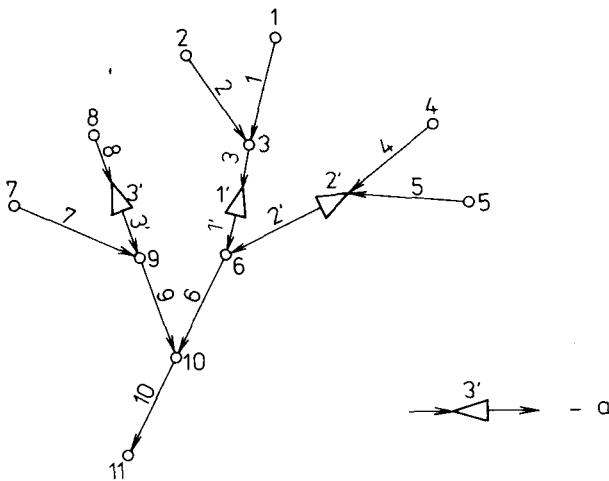


Fig. 9.7 A river network with reservoirs modelled by a tree  
 $a$  – a reservoir and its number

It is then necessary to use a new numbering system and the added reservoirs are given new numbers, e.g., nodes  $1', 2', 3'$  up to  $n' = 3$ . Thus new arcs of the graph will originate at  $1', 2', 3'$  (the total sum being  $v' = n' = 3$ ).

<sup>1)</sup> The mass balance equation, or, in electrical networks, Kirchhoff's law, states that the sum of the flows entering the node is equal to the sum of the flows exiting from the node.

<sup>2)</sup> A transposed matrix is generated by the change of rows and columns and vice-versa. Therefore it satisfies  $a_{i,j} = T_{a_{j,i}}$  and vice-versa for all  $i, j$  where  $a_{i,j}$  is the element of the original matrix and  $T_{a_{j,i}}$  is the element of the transposed matrix.

<sup>3)</sup> Inverse matrix  $\mathbf{A}^{-1}$  (of matrix  $\mathbf{A}$ ) is such a matrix that satisfies  $\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{I}$  where  $\mathbf{I}$  is the identity matrix (unit matrix with non-zero elements on the main diagonal equal to one).

The incidence matrix has the following form

$$\mathbf{A} = \begin{array}{c}
 \begin{array}{cccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 1' & 2' & 3' \\
 1 & \left| \begin{array}{cccccccccccc}
 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 2 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
 4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
 5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
 7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\
 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 \hline
 1' & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 2' & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 3' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
 \hline
 \end{array} \right| \\
 \end{array}
 \end{array} \quad (9.19)$$

A non-stationary flow pattern is assumed, with a vector function for the storage of water  $V_j$  in reservoirs in time  $t$  calculated in the following manner:

$$\mathbf{v}(t) = \begin{array}{c} \left\| \begin{array}{c} V_1(t) \\ V_2(t) \\ \vdots \\ V_j(t) \\ \vdots \\ V_u(t) \end{array} \right\| = \begin{array}{c} \left\| \begin{array}{c} V_1'(t) \\ V_2'(t) \\ \vdots \\ V_3'(t) \end{array} \right\| \end{array} \quad (9.20)$$

The time dependent vector function of withdrawals from the  $j$ -th node or of inflow into this node for reservoirs and other nodes have the following form

$$\mathbf{r}(t) = \begin{array}{c} \left\| \begin{array}{c} R_1(t) \\ R_2(t) \\ \vdots \\ R_j(t) \\ \vdots \\ R_u(t) \end{array} \right\| = \begin{array}{c} \left\| \begin{array}{c} R_1(t) \\ R_2(t) \\ R_3(t) \\ \vdots \\ R_8(t) \end{array} \right\| ; \mathbf{r}'(t) = \begin{array}{c} \left\| \begin{array}{c} R_{1'}(t) \\ R_{2'}(t) \\ \vdots \\ R_{j'}(t) \\ \vdots \\ R_{u'}(t) \end{array} \right\| = \begin{array}{c} \left\| \begin{array}{c} R_1(t) \\ R_2(t) \\ R_3(t) \end{array} \right\| \end{array} \quad (9.21)$$

(9.22)

The relationship of the flow in the  $i'$ -th arc under the  $j'$ -th reservoir in time  $t$ , representing the reservoir release is similarly expressed by a vector function

$$s(t) = \begin{pmatrix} S_1(t) \\ S_2(t) \\ \vdots \\ S_{i'}(t) \\ \vdots \\ S_v(t) \end{pmatrix} = \begin{pmatrix} S'_1(t) \\ S'_2(t) \\ S'_3(t) \end{pmatrix} \quad (9.23)$$

The vector function of flows in the arcs is

$$q(t) = \begin{pmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_i(t) \\ \vdots \\ q_v(t) \end{pmatrix} = \begin{pmatrix} q_1(t) \\ q_2(t) \\ \vdots \\ q_8(t) \end{pmatrix} \quad (9.24)$$

Matrix  $A$  can be partitioned as shown in Fig. 9.9. Submatrix  $A_0$  is similarly to matrix (9.14): a square and regular matrix. As in nodes 1, 2, ...,  $j$ , ...,  $n - 1$  the mass balance is satisfied, it is

$${}^T A_0 q(t) + {}^T A_2 s(t) + r(t) = 0 \quad (9.25)$$

and the function sought is

$$q(t) = -{}^T A_0^{-1} [{}^T A_2 s(t) + r(t)] \quad (9.26)$$

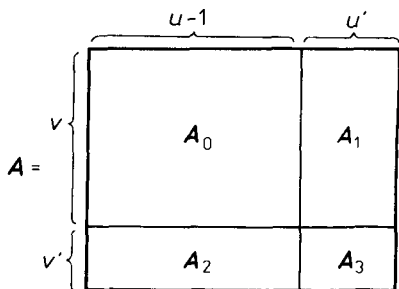


Fig. 9.8 Partition of the incidence matrix of the river network with reservoirs into submatrices

In the nodes representing the reservoirs ( $1', 2', \dots, j', \dots, u'$ ) the mass balance is not satisfied (in time  $t$ ) due to storage. The difference of inflow and the outflow in the

node is not necessarily equal to zero, and it can be described by a vector function for the whole system

$$z(t) = \begin{Bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_j(t) \\ \vdots \\ z_u(t) \end{Bmatrix} = \begin{Bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \end{Bmatrix} \quad (9.27)$$

This function equals

$$z(t) = -[{}^T\mathbf{A}_1 q(t) + {}^T\mathbf{A}_3 s(t) + r'(t)] \quad (9.28)$$

Substituting  $q(t)$  from the equation (9.26) the following expression is obtained

$$z(t) = {}^T\mathbf{A}_1 \{ {}^T\mathbf{A}_0^{-1} [ {}^T\mathbf{A}_2 s(t) + r(t) ] \} - {}^T\mathbf{A}_3 s(t) - r'(t) \quad (9.29)$$

The time-related change of reservoir storage can be expressed by

$$dv(t) = z(t) dt \quad (9.30)$$

and the differential equation for the determination of the vector function  $v(t)$  has the following form

$$\frac{dv(t)}{dt} = {}^T\mathbf{A}_1 \{ {}^T\mathbf{A}_0^{-1} [ {}^T\mathbf{A}_2 s(t) + r(t) ] \} - {}^T\mathbf{A}_3 s(t) - r'(t) \quad (9.31)$$

For the given boundary condition in the point  $t = t_0$ :  $v(t_0) = v(t_0)$  the equation (9.31) can be solved numerically.

## 9.5.2 Matrix Analysis of Cyclic Public Water Supply Networks

The methods of matrix analysis of water supply networks were developed by Šerek (1968, 1972, 1975). The algorithms for computers were set up for the solution of practical problems.

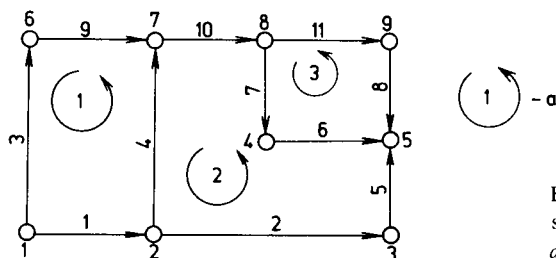


Fig. 9.9 Network model of the cyclic water supply network

$a$  — orientation and notation of the cycle

The topological structure of the graph of a cyclic water supply network is characterized by cycles with a chosen orientation in the graph. The incidence matrix of the graph is not generally a square matrix. The rows represent the arcs of the graph and the columns represent the cycles. For the graph in Fig. 9.9 the incidence matrix has the following form

$$\mathbf{B} = \begin{matrix} & & 1 & 2 & 3 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{matrix} & \left\| \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{matrix} \right\| & & & \end{matrix} \quad (9.32)$$

Then the vector of flows  $\mathbf{q}$  and the vector of pressure head losses  $\mathbf{h}$  are computed in each arc (sections of the water supply network) which are the vector function of the flows, i.e.

$$\mathbf{h} = \mathbf{f}(\mathbf{q}) \quad (9.33)$$

The vectors  $\mathbf{h}$  and  $\mathbf{q}$  must satisfy both Kirchhoff's laws:

$${}^T\mathbf{A}\mathbf{q} + \mathbf{r} = \mathbf{o} \quad (9.34)$$

$${}^T\mathbf{B}\mathbf{h} = \mathbf{o} \quad (9.35)$$

where the matrix  $\mathbf{A}$  is set up as in section 9.5.1.

An irrigation network can be treated similarly. The solution of problems of water supply for municipalities, or irrigation, is the determination of optimal sites for water mains and other facilities, using an economic objective (minimization of initial and operation costs). The task of optimization is similar to that of the transportation problem in linear programming. The non-linear relationship between cost and the amount of water transported in the pipeline requires the application of non-linear programming (Šerek, 1975). The automation of the design of the water supply and irrigation networks is a great help in this task.

### 9.5.3 Network Model of WRS

A WRS as a set of water management facilities and resources with mutual relationships, can be represented as a valuated digraph network. The algebraic description is done in the form of an incidence matrix that is set up using the method described in 9.5.1.

For the investigation of WRS by optimization models, the optimization of flow in networks can be used (Štichová, 1972). The valuation of the arcs of the graph is given by the capacity of the transportation or transfer facilities, possibly the valuation of the nodes is given by (a) the capacity of the facilities and water resources objects (waterworks reservoir, water and waste water treatment stations, etc.), or (b) by the required withdrawal in the node.

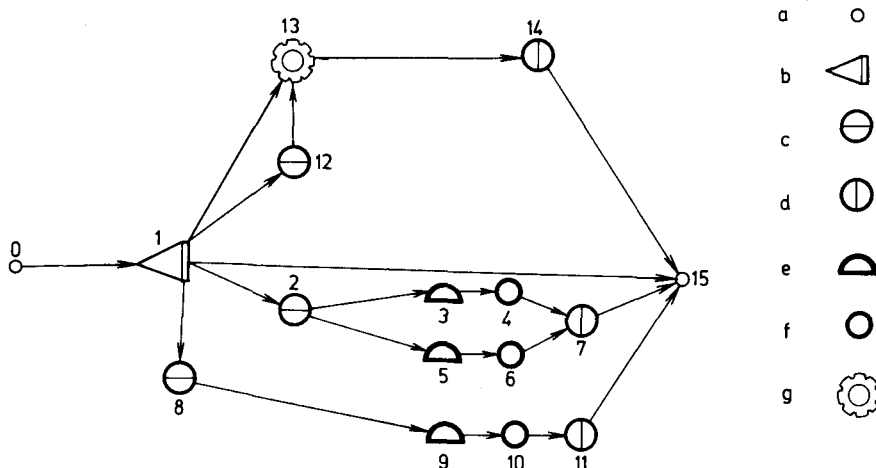


Fig. 9.10 The network model of the WRS

*a* – input and output, *b* – reservoir, *c* – water treatment station, *d* – waste water treatment station, *e* – waterworks reservoirs, *f* – withdrawal of drinking water, *g* – withdrawal of water for industry

An example of a model of WRS is given in Fig. 9.10. It is a static model with no time dimension and using one stage only. The incidence matrix of the graph can be set up as in section 9.5.1. The following boundary conditions must be satisfied:

a)  $x_{i,j} \leq k_j$ , i.e., the flow in the arc  $j^1$  from node  $i$  to node  $j$  must be less than, or equal to, the capacity of the node (in the case of node valuation (a));

<sup>1)</sup> The arc connecting nodes  $i$  and  $j$  is denoted  $x_{i,j}$ .

b)  $x_{i,j} \geq k_j$ , i.e., the flow in arc  $j$  from node  $i$  to node  $j$ , where this arc ends, must be greater than, or equal to, the withdrawal in this node (in case of node valuation (b));

$$c) \sum_{i=1}^m x_{i,p} = \sum_{j=1}^n x_{p,j} + \zeta$$

i.e., the sum of all inflows to the node must be equal to the sum of the outflows, withdrawals and losses (Kirchhoff's law), where  $x_{p,j}$  are outflows and withdrawals and  $\zeta$  is the losses;

$$d) \sum_{j=1}^{n'} x_{0,j} = \sum_{i=1}^{m'} x_{i,k},$$

i.e., the total flow of the network on the input must be equal to the total flow on the output of the network;

e)  $x_{1,k} \geq d$ , i.e., at the end of the graph a minimum flow  $d$  must be maintained (minimum acceptable flow).

The values of flows in all arcs of the network are determined by optimization of this model of WRS satisfying all the boundary constraints. The determination of the optimal flow in arcs of the network is solved by linear programming with minimization of the objective function (minimum costs of water treatment, waste water treatment, pumping, etc.).

The linear optimization model can be formulated mathematically as a linear function

$$z = \sum_{i,j} c_{i,j} x_{i,j} \tag{9.36}$$

where  $c_{i,j}$  are the unit costs.

This function is minimized under constraints a) to e) and under the non-negativity constraint  $x_{i,j} \geq 0$ .

V. Štichová, 1972 used a simple network model for optimization of the WRS in the basin of the Blanice River with a single reservoir for municipal and industrial water supply. The water demands, capacities of water treatment stations, the estimation of losses in reservoirs and the network were given together with the costs of water treatment, waste water treatment, pumping into water works reservoirs, etc. The linear model was optimized by a standard linear program "MOSI 3" on computer NE4100.

In summary, the static model is criticised as it neglects the time dimension and yields an unrealistic representation of actual tasks. The random variation of the model input of the WRS can be used to give a dynamic model of the network. However, the dimension of the model with the time change of only one parameter of the system (reservoir inflow) increases linearly (Fig. 9.11.). This idea of the model dyna-

mization cannot reflect the process of storage of water in reservoirs, which is a complex inventory problem and its optimization is assumed by methods of linear programming.

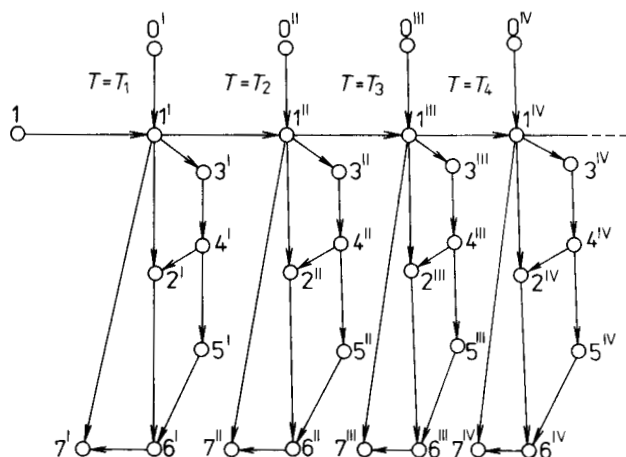


Fig. 9.11 Scheme of the model network with time dimension on input

The application of the graph theory in mathematical modelling of WRS offers an interesting impetus. However, the advantages of this method (good insight into the problems, easy algorithmization with representation of WRS as a graph and in matrix analysis) seem to diminish with the growing complexity of WRS, with the inclusion of the time changes of all the important input parameters and with optimization reflecting the actual properties of the process.