5 MODELS OF OPTIMAL PROGRAMMING

The term optimal (mathematical) programming[1]) refers to a set of methods that are used for optimization of operations, e.g. the well-known linear, non-linear, and dynamic programming.

There is an extensive literature dealing with the methods of optimal programming. Therefore, references will be used and our attention will be concentrated on their application in WRS. Principles of optimal programming will be cited where necessary for computation in application models. Not all the known applications can be presented; only those directions of application that seem promising and progressive will, therefore, be given. We shall proceed from simple to more complicated models.

5.1 LINEAR PROGRAMMING

5.1.1 Models of Linear Programming

Linear programming is the most widely used and the simplest method of operations research. It is explained e.g. by Dantzig, 1963; Gass, 1969; Korda *et al.*, 1967; Walter *et al.*, 1973, and others.

Every optimization task tries to solve one of two alternative problems: either to attain a maximum (minimum) objective function with the given resources or to meet the given goal with a minimum of resources. The tasks dealt with in linear programming are expressed by:

- a set of equations and inequalities (constraints),
- the condition of non-negativity (the unrealistic condition of e.g. the negative value of production is suppressed),
- an objective (criterial) function (expressing the goal or objective that has to be met).

All the constraints and the objective function must be linear. This requirement of linearity implies simplicity but involves a limited applicability of the models of linear programming. They can be used for one-stage optimization in contrast to dynamic programming that makes a repeated optimization in a series of stages of the optimization process possible.

---

[1]) Programming used in this sense is not equivalent to programming computers, and in its broader meaning it is closer to planning.

The general mathematical model of linear programming can be formulated as follows:

$$x^{\circ} = [x_1^{\circ}, x_2^{\circ}, \ldots, x_n^{\circ}] \tag{6.1}$$

under the constraints of the task ($m < n$; the equations are linearly independent),

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned} \tag{5.2}$$

and, further, under conditions of non-negativity,

$$x_j \geqq 0 \; (j = 1, 2, \ldots, n) \tag{5.3}$$

minimizing the linear objective function,

$$z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{5.4}$$

where x_i are variables and a_i, b_i, c_i are known constraints.

The important fact is that there are more variables than equations. Therefore, an unlimited number of feasible solutions exists. Such cases are typical of linear programming. If the number of equations and variables were identical, a single solution would exist, and the search for an optimal solution given by an objective function would not be possible.

A simple example taken from economics can serve as an illustration of the methods of linear programming:

A factory produces two kinds of products V_1 and V_2. Two different raw materials are required, S_1 and S_2. They are available in limited quantities, 1200 kg and 1600 kg, respectively. The first product V_1 requires 3 kg of raw material S_1 and 2 kg of raw material S_2; the second product V_2 requires 1 kg of raw material S_1 and 2 kg of raw

Table 5.1 Linear programming input scheme

Raw material	Consumption of raw materials per one product (kg)		Amount of raw materials at disposal (kg)
	V_1	V_2	
S_1	3	1	1200
S_2	2	2	1600
Price of one product in $	20	10	

material S_2. The market price of product V_1 is \$ 20 and of product V_2 \$ 10. The task is to determine which product and which quantity must be produced for the factory to gain the maximum benefit. The linear programming input scheme is given in Table 5.1.

There are many feasible production schedules. Some of them are listed in Table 5.2. Which procedure can be used in search of an optimal program? It is obvious that it is not advantageous to list all the possible programs and thus try to find the best case. An enormous number of computations would be necessary even if the clearly disadvantageous alternatives are omitted.

Table 5.2 List of some possible production schedules

Number of products		Benefits	Quantity of raw materials not utilized (kg)	
V_1	V_2		S_1	S_2
400	0	8000	0	800
300	300	9000	0	400
250	450	9500	0	200
200	600	10 000	0	0
150	650	9500	100	0
100	700	9000	200	0
0	800	8000	400	0

Linear programming determines the optimal schedule as a precise and single solution in the following way:

x_1 denotes the number produced of the first product V_1; similarly, x_2 denotes the number of product V_2. Using variables x_1 and x_2, and the known quantities of the raw materials necessary for the production of each product, we can express the condition that the maximum consumption of the first raw material is 1200 kg and of the second raw material is 1600 kg. If one product V_1 requires 3 kg of raw material S_1, then x_1 products need $3x_1$ kg of raw material S_1. If the sum of the amount used for the two products is less than or equal to 1200 kg, the following inequality can be met

$$3x_1 + 1x_2 \leqq 1200$$

The same consideration for the consumption of the second raw material S_2 in the production of both products results in the inequality

$$2x_1 + 2x_2 \leqq 1600$$

If both these inequalities are considered, then couples (x_1, x_2) will be obtained expressing feasible production schedules. It is apparent that the negative solution of the inequalities has no real interpretation. Further conditions are therefore added:

$$x_1 \geqq 0; \quad x_2 \geqq 0$$

The objective of production can be formulated as the attainment of maximum benefit. This benefit denoted as z, will be expressed in terms of the price \$ 20 of product V_1 and \$ 10 of product V_2 and quantities x_1 and x_2 produced, in the following way:

$$z = 20x_1 + 10x_2$$

The production schedule, expressed by the couple (x_1, x_2) must meet the given inequalities (i.e. constraints) and maximize the expression $z = 20x_1 + 10x_2$.

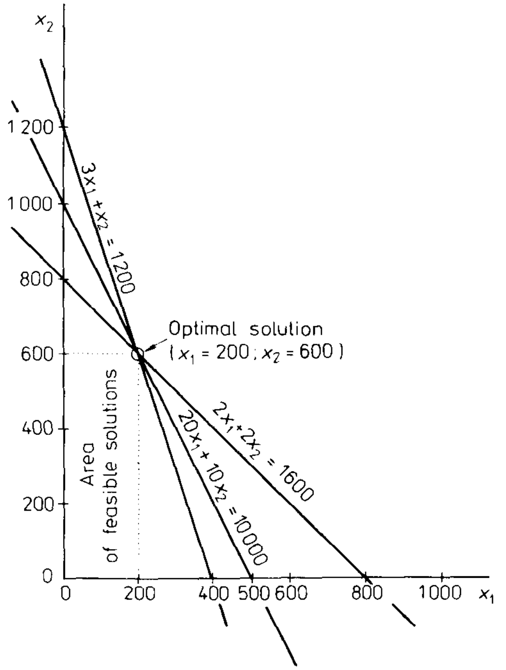


Fig. 5.1 Graphic solution of the simple example of linear programming

The task can be expressed in this way:
A couple (x_1, x_2) has to be found, meeting the inequalities

$$3x_1 + x_2 \leqq 1200 \tag{5.5}$$

$$2x_1 + 2x_2 \leqq 1600$$

$$x_1 \geqq 0; \quad x_2 \geqq 0 \tag{5.6}$$

so that the function

$$z = 20x_1 + 10x_2 \tag{5.7}$$

should reach its maximum.

The inequalities (5.5) limit or constrain the choice of program. Therefore they are called constraints of the task. The inequalities (5.6) are also constraints, and they are called non-negativity restrictions. The function (5.7) is called the objective or criterial function. All the inequalities and the objective function are linear; the problem can therefore be solved by linear programming.

Our economic task has thus been transformed to a mathematical set of inequalities (5.5) and (5.6) and the objective function (5.7).

Computation of this simple example is easy, that is, it can be performed by a graphical constructions (Fig. 5.1). In the optimal production schedule, given in the first row in Table 5.2, the numbers of products V_1 and V_2 produced will be 200 and 600, respectively. The benefits are $ 10,000 and all the raw material is consumed. In a practical application, a set consisting of dozens of equations is solved on a computer.

Most of the tasks of linear programming are dealt with by two basic procedures:

– the simplex method that can be used generally for each model of linear programming, although in some cases it is cumbersome;

– a model of distribution type – e.g. the transportation problem, the computation of which is easier in some cases, but which requires the transported objects to be homogeneous and mutually exchangeable.

5.1.2 Application of Linear Programming in WRS

In recent years linear programming has been very often used. The final report of the Harvard Program (Maass *et al.*, 1962) includes a detailed and comprehensive analysis. Interesting applications of linear programming were published by Buras, 1972; Kos, 1972; Vedula and Rogers, 1981; Rydzewski and Rashid, 1981, and others.

These applications of linear programming have produced many positive results. The occupation is simple in principle. The software of almost every computer includes universal programs for linear programming such as the MPSX linear programming package for IBM computers, e.g. IBM 370(168) where no programming and program debugging is necessary. It is only necessary to assemble the data for the task under investigation and transform them into the form required by the program.

The main drawback is the assumption of linear relationships, particularly in the objective function. Only for a limited set of problems is this assumption acceptable without an undesirable distortion of reality. Another drawback is the one-stage optimization whereas the process of resource management is dynamic. Therefore repeated optimization is periodically required to account for the changes in the state of the system¹). A further drawback is the deterministic nature of most linear programming models; this can be overcome by using chance-constrained models (see Chapter 10).

In general, it can be stated that the linear programming models are used mainly as "screening" models for the preliminary design of WRS at a rough discriminative level. A large number of alternatives are considered with different input values. Thus, a region with acceptable parameters is selected for further investigation by simulation models at a more detailed discriminative level. Such a procedure was used for WRS for the water supply of New York. Linear programming models are sometimes combined with dynamic programming or other models.

Linear programming for two hydrological periods

The simplest model is deterministic as it assumes known hydrological conditions. The computation is performed for one year, and only two different hydrological periods are assumed, a wet and a dry period. Both are described by the average flow in volume units. This model can be illustrated by an example:

Investigations are carried out on a WRS, the scheme of which is shown in Fig. 5.2. It includes reservoir N_1 with active storage V_1 and reservoir N_2 with active storage V_2 an irrigated area with annual water requirements V_z and a water power plant with annual production E_{prod}.

The following input values are given:

– the flows in the wet and dry periods at given points. The upper number in the scheme applies to the wet period, the lower applies to the dry period. Both are in 10^9 m^3;

– 40% of the irrigation water requirements V_z should be met in the wet period, 60% in the dry period. The return flow from the irrigation area is assumed to be 10% of V_z in the wet period and 30% of V_z in the dry period;

– power production in the water power plant must not fall below the given value E_f in any period; half of the annual production is required in the wet period and half in the dry period.

The active storage of both reservoirs, the water withdrawal for irrigation, and the power production will be determined so as to maximize the benefits from this WRS.

The constraints given in Fig. 5.2 were derived from input values and requirements.

---

[1] The static approach of linear programming models is acceptable in, for example, optimization of the allocation of withdrawals of water from reservoirs on the River Nile for different users. Every year follows the same pattern: the reservoirs are filled at the same time every year by approximately the same volume of melt-water from the mountainous region of the upper reaches, and the withdrawals, mainly for irrigation, take place in the same season of the year. The dynamic nature of the process can be neglected without causing great distortion.

It is assumed that in the wet period the active storage of the reservoirs is filled up and in the dry period it is emptied. Therefore, the release from reservoir N_1 for example in the wet period is $0.95 - V_1$ and in the dry period $0.54 + V_1$. The other data in this scheme are clear.

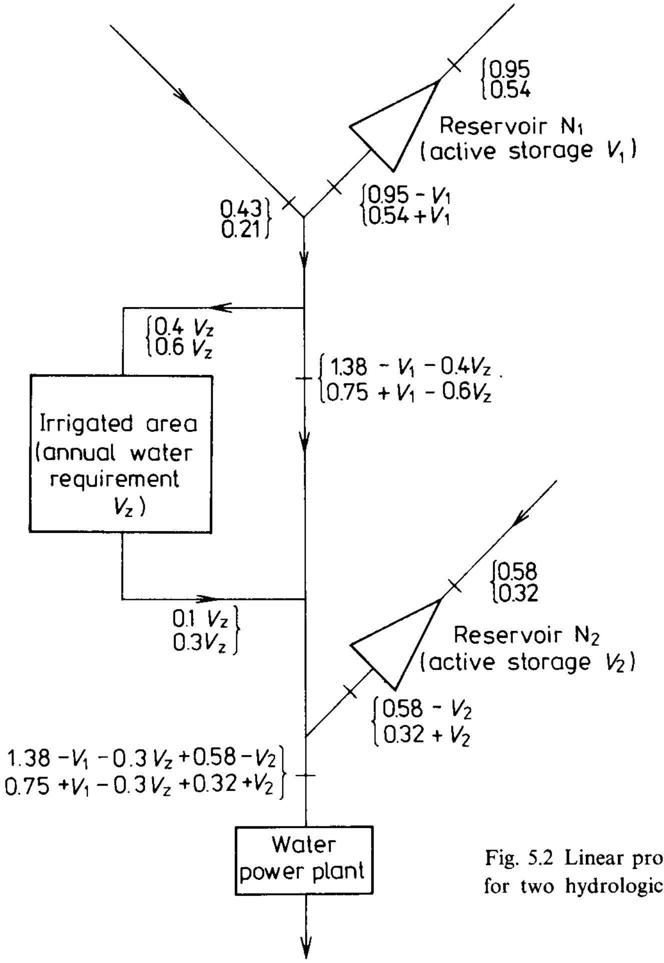


Fig. 5.2 Linear programming for two hydrological periods

The task is formulated in terms of three groups of constraints and the objective function. The constraints of the first group require that no variable should be negative $(V_1, V_2, V_z, E_{\text{prod}})$. The second group of constraints requires that flows at all points of the WRS should be non-negative. These constraints are

$$\begin{aligned}
0.95 - V_1 & \geqq 0 \\
1.38 - V_1 - 0.4V_z & \geqq 0 \\
0.75 + V_1 - 0.6V_t & \geqq 0 \\
0.58 - V_2 & \geqq 0
\end{aligned} \tag{5.8}$$

The third group of constraints requires that the flow in the power plant should be sufficient both in dry and wet periods for the production of firm energy:

$$\begin{aligned} 1.38 - V_1 - 0.3V_z + 0.58 - V_2 &\geqq 0.5k_e E_f \\ 0.75 + V_1 - 0.3V_z + 0.32 + V_2 &\geqq 0.5k_e E_f \end{aligned} \tag{5.9}$$

The coefficient k_e transforms the volume of water flowing through the turbines into power production, the coefficient 0.5 express the necessity to produce half of the annual production in each period and E_f is the required firm energy production.

The objective function is maximized:

$$\begin{aligned} \pi = SH(VE) + SH(V_v) - SH(PN_1 + IN_1) - SH(PN_2 + IN_2) - \\ - SH(PN_E + IN_E) - SH(PN_z + IN_z) \end{aligned} \tag{5.10}$$

where π is the present value[1]) of net benefits from WRS,

$SH(VE)$ – present value of benefits from power production,

$SH(V_v)$ – present value of benefits from irrigation water supply,

$SH(PN_1 + IN_1)$ – present value of operation and investment costs of reservoir N_1 with active storage V_1,

$SH(PN_2 + IN_2)$ – present value of operation and investment costs of reservoir N_2 with active storage V_2

$SH(PN_E + IN_E)$ – present value of operation and investment costs of water power plant with capacity E_{prod} (kWh/year)

$SH(PN_z + IN_z)$ – present value of operation and investment costs of irrigation system with capacity V_z.

In the formulation of such tasks for linear programming the problem of a non-linear objective function is encountered, and a possible solution of this problem was described by Maass *et al.*, 1982. In principle, there are two possibilities:

– Each non-linear function is replaced by a broken line. This procedure can be followed if each non-linear function can be separated into components, each including only one variable (separable objective function).

---

[1]) The present value of benefits (V) and operation costs (VN) is given by the sum of discounted annual values of benefits and costs from the beginning of operation to the end of the economic lifetime period of investment (economic planning horizon). If i per cent is the interest rate per period and $\alpha = [1 + (i/100)]^{-1}$ is the single period discount factor, the present value R of the sequence of R_t is

$$R = \sum_{t=1}^{N} \alpha^{t-1} R_t$$

The present value of investment (initial) costs is the sum of investment costs and interest from the beginning of construction to the beginning of operation

$$\text{Present value } I = \sum_{t=1}^{M} \beta^{M-t} I_t$$

where $\beta = 1 + i/100$ (in ČSSR $i = 6$ per cent is recommended).

– If such a separation is not possible, the linearization becomes more complicated.

If the constraints are non-linear functions of only two variables, linearization uses the analogy between a curve for a function of one variable and a surface in the case of two variables. Analogous to the line given by two points on a curve is a plane given by three points in space. Linear approximation of the function of two variables can therefore be performed by a rectangular grid so that the values of the function in all intersection nodes of the grid are determined. Linear approximation can be a weighted mean of these values. Other approaches are also possible, of course, e.g. the finite elements method.

The problem of non-linear functions being solved, further computation is not difficult in principle, and modern computers have enough memory capacity and speed for complex systems. The main drawback of the model is the assumption of a regular pattern of flow and annual control. The model described can therefore be used for a first screening.

Linear model for more than two hydrological periods

The model for two hydrological periods described in the previous section cannot be used if carry-over occurs. In that case a more comprehensive model must be applied, based on the following assumptions:

– the hydrological conditions are assumed to be known; i.e. it is a deterministic model,

– unlike the previous model, the carry-over is considered in a series of different hydrological periods.

The method is explained for four periods, so that a large series can be used. The scheme of the model is given in Fig. 5.3. To facilitate comparison with the previous model, a similar scheme was chosen with the following changes:

– reservoir N_2 was omitted,

– the total water requirement for irrigation V_z for all four periods was given,

– the values of flow were determined in 10^9 m^3 at three points for each period,

– the release from reservoir N_1 in individual periods was denoted as o_t, where $t = 1, 2, 3, 4$. The other data in the scheme are clear, and they were determined in a similar way to the previous case.

As the amount of irrigation water was given, the aim of the task is: the determination of the active storage in reservoir N_1 and the power production which maximizes benefits from WRS. The following objective function will be maximized:

$$\pi = SH(VE) - SH(PN_1 + IN_1) - SH(PN_E + IN_E) \qquad (5.11)$$

where π is present value of net benefits from WRS,

$SH(VE)$ – present value of benefits from power production,

$SH(PN_1 + IN_1)$ – present value of operation and investment costs of reservoir N_1 with active storage V_1,

$SN(PN_E + IN_E)$ – present value of operation and investment costs of power plant with capacity E_{prod} (kWh/4 periods).

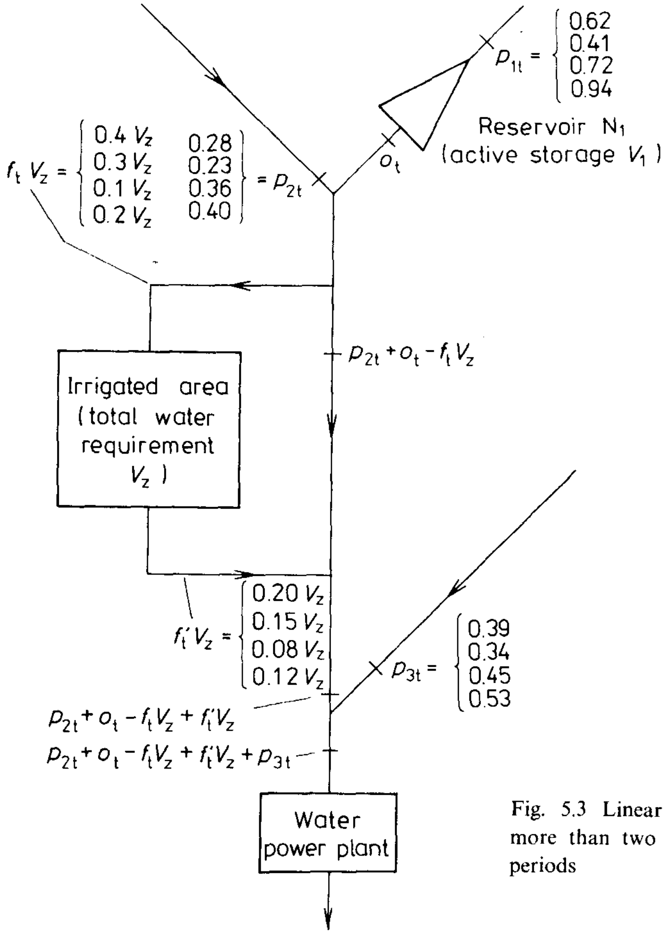


Fig. 5.3 Linear model for more than two hydrological periods

The objective function is, of course, non-linear. The methods of linear approximation are the same as in the previous example. For each hydrological season there are six constraints. The first one is apparent from the expression

$$o_t + p_{2,t} \geqq f_t V_z \tag{5.12}$$

The second constraint states that the volume of water released in each period must not exceed the sum of reservoir storage at the beginning of that period and the reservoir inflow during that period.

The reservoir storage at the beginning of period t is n_t; then

$$o_t \leqq n_t + p_{1,t} \tag{5.13}$$

The third constraint specifies that reservoir storage at the beginning of any period must not exceed the storage at the end of the previous period:

$$n_t = n_{t-1} + p_{1,t-1} - o_{t-1} \tag{5.14}$$

According to the fourth constraint, the reservoir storage at the end of any period must not exceed the active storage V_1:

$$n_t + p_{1,t} - o_t \leqq V_1 \tag{5.15}$$

The last two constraints concern power production. The fifth constraint guarantes the firm energy production

$$o_t + p_{2,t} - (f_t - f_t')\, V_z + p_{3,t} \geqq k_e E_{t,f} \tag{5.16}$$

where the coefficient k_e transforms the volume of water flowing through the turbines into power production.

According to the sixth, and last, constraint, power production in each period should equal, or exceed, the pre-determined proportion c_t of the total production.

$$E_{t,\text{prod}} \geqq c_t E_{\text{prod}} \tag{5.17}$$

The six constraints listed are valid in each period. The values of variables o_t, n_t, V_1, $E_{t,\text{prod}}$, E_{prod} are determined so as to maximize the objective function. This example with 24 constraints can easily be handled by any computer. Usually, substantially more hydrological periods are considered. Then a reduction of the computation burden is achieved by the principle of decomposition of linear programs (Maass *et al.*, 1962).

Linear programming with stochastic flows

In this model the variables are not deterministic, as in the previous models, and only the expected values and the probability distributions are known. The principles can be explained by a simple example:

A WRS with the scheme given in Fig. 5.4 is considered. It includes reservoir N, an irrigated area, and a power plant. The computation is performed for three hydrological periods. In this scheme the following notation is used:

p_t – inflow to reservoir N in period t ($t = 1, 2, 3$)
o_t – release from reservoir N in period t,
z_t – withdrawal of water for irrigation in period t,
$g_t z_t$ – return flow from irrigation.

The system was simplified, compared with the previous model. The tributary inflows α, β, marked in Fig. 5.4 by a dashed line, are considered in a simple way as being deterministically related to stochastic flow p_t. In computation, the stochastic releases from reservoir N are increased by the values of the tributary inflow α. Similarly, the flows in the water power plant are increased by β, determined with reference to p_t.

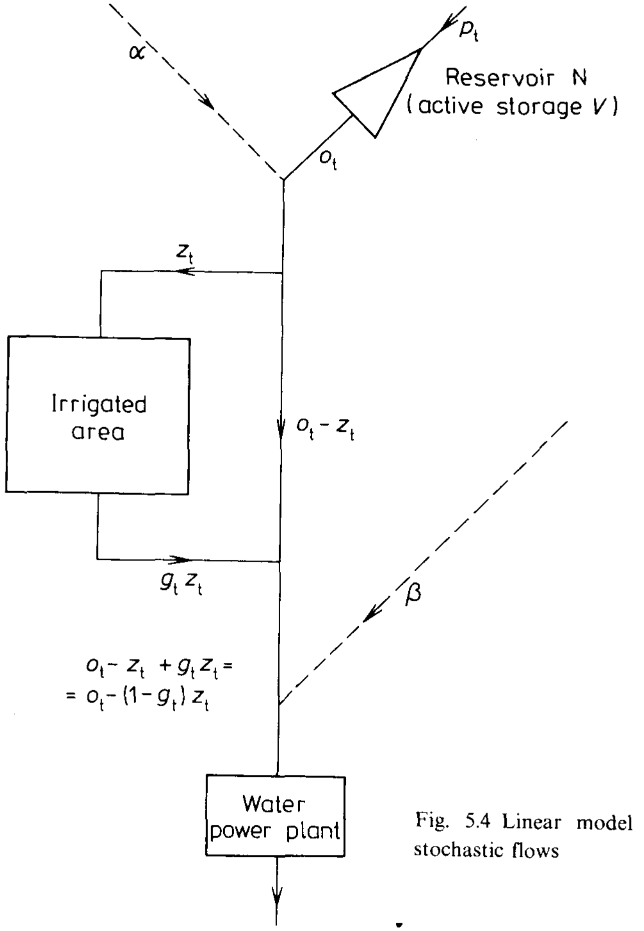


Fig. 5.4 Linear model with stochastic flows

The aim of the task is to determine the values of the following parameters that will maximize the objective function:

- active storage of reservoir N,
- irrigation water requirements V_z for all periods,
- rated generating capacity of power plant I_e,
- target power production in each period E_c.

The flows p_t are given by their probability distributions for each of the three periods involved. The flows in each period are assumed to be statistically independent.

The maximized objective function has the following form:

$$\pi = SH(VE) + SH(V_v) - SH(PN_1 + IN_1) - SH(PN_E + IN_E) + \tag{5.18}$$

+ value of dump energy − losses due to deficits in irrigation water supply and in firm energy,

where π is present value of net benefits from WRS for all periods,

$SH(VE)$ – present value from power production,

$SH(V_v)$ – the present value of benefits from the irrigation water supply,

$SH(PN_1 + IN_1)$ – the present value of operation and investment costs of reservoir N_1 with active storage V_1,

$SH(PN_E + IN_E)$ – the present value of operation and investment costs of power plant with rated generating capacity I_e.

Since the objective function is non-linear, its linearization is performed in a similar way to the first model of this section (Maass *et al.*, 1962).

Linear programming cannot be directly used for this model with a linearized objective function, since it depends on stochastic inlows p_t. Therefore the maximization of the expected value of benefits from WRS is performed and all random variables in the objective function substituted by their expectations.

As an example of some variable expectation computation, the withdrawal of irrigation water in the first period z_1 is considered. This variable can acquire four possible values $z_{1,2}$, $z_{1,3}$, $z_{1,4}$, or $z_{1,5}$ with probabilities r_2, r_3, r_4, r_5. The same probabilities r_y are assumed for flows p_1, e.g. the probability that $z_1 = z_{1,2}$ equals the probability that $p_1 = p_{1,2}$ etc. Then the expected value of irrigation withdrawals in the first period is

$$z_1 = \sum_{y=2}^{5} r_y z_{1,y} \tag{5.19}$$

where r_y is the probability that $z_1 = z_{1,y}$ or $p_1 = p_{1,y}$ (the flow in the first period p_1 will be equal to $p_{1,y}$)

The constraints for the first period are:

$$\begin{aligned}
&n_1 + p_{1,y} \geqq o_{1,y} \\
&n_1 + p_{1,y} - o_{1,y} \leqq V_1 \\
&n_1 + \sum_{y=2}^{5} r_y p_{1,y} - \sum_{y=2}^{5} r_y z_{1,y} \geqq n_2 \\
&z_{1,y} \leqq o_{1,y} \\
&z_{1,y} \leqq k_z V_z \\
&u_{1,y} \leqq o_{1,y} - (1 - g_z)\, z_{1,y} \\
&k_e u_{1,y} \leqq k_d I_e \\
&e_{1,y} \geqq E_c - k_e u_{1,y}
\end{aligned} \tag{5.20}$$

where

n_1 is the assumed reservoir storage at the beginning of the first period,

$o_{1,y}$ – reservoir releases in the first period for the given value of inflow $p_{1,y}$ when $y = 2, 3, 4, 5$,

$z_{1,y}$ – irrigation water supply in the first period for the given flow $p_{1,y}$,

$u_{1,y}$ – the flow through the turbines (power flow) for the given flow $p_{1,y}$,

$e_{1,y}$ – the value of energy added by thermal power plants in the first period for the given flow $p_{1,y}$,

n_2 – the minimum reservoir storage at the end of the first period,

k_z – the coefficient determining that proportion of the irrigation water requirement V_z that is to be supplied in the first period,

k_e – the constant that relates power flow and power output,

k_d – the duration of the period in hours multiplied by the loading factor (e.g. 5/24 if the power plant operates 5 hours per day).

In this example, there are 60 constraints for three periods, which presents no problem for computation. If the problem were more complicated, the decomposition principle mentioned above would be used. This model is also used as a screening model in the design of WRS.

Model using mixed integer linear programming

This type of model was developed in Czechoslovakia by Korsuň *et al.*, 1972, 1975, 1976, and its mathematical principles were published by Glückaufová and Trčka, 1972. Although most applications were used for irrigation systems, they can be applied to any WRS.

Mathematical principles of the model: Integer linear programming is a non-linear procedure that would be linear if it were not for the fact that some variables can acquire integer values only. These procedures may be classified into two types:

– pure integer problems of linear programming are those where all variables are integer-valued,

– mixed integer problems of linear programming are linear programs where some variables are integer-valued and others are continuous.

In practical applications, mixed integer linear programming is often used when some variables can acquire the values zero and one only. Such problems are called bivalued ("go-no-go", or zero-one) mixed integer linear programming problems. The importance of this class of problems lies in the fact that many non-linear problems, which seemingly have nothing in common with the bivalued program, can be formulated as bivalued mixed integer linear programs. The advantage of bivalued programming is the possibility of using universal programs for the optimization of large and complex problems with some non-linear and discontinuous functions. The dynamics of processes can to some degree be reflected in such models, i.e. the time dimension can be considered.

The mathematical model of WRS (Koršuň, 1976) is formed by a set of m equalities and inequalities with n variables. The equalities and inequalities are of the following type:

$$\sum_{j \in I_1} a_{ij} x_j + \sum_{k \in I_2} a_{ik} x_k \gtreqless b_i \qquad (5.21)$$

where

$$\begin{aligned} & x \geqq 0 \\ & i \in \{1, 2, \ldots, m\} \\ & j \in I_1 \\ & k \in I_2 \\ & I_1 \cup I_2 = \{1, 2, \ldots, n\} \end{aligned} \qquad (5.22)$$

where x_j are continuous variables used for modelling various aspects of the system investigated (e.g. production, technological, financial, economic activities, etc.),

x_k – bivalued variables, or zero-one variables; these variables can be used for modelling integer (indivisible) activities or for discontinuous or non-linear relationships,

a – technological or economic coefficients of variables,

b – the right-hand side constants in equalities and inequalities,

I_1 – the set of subscripts of continuous variables,

I_2 – the set of subscripts of bivalued variables.

The solution of the mathematical model is obtained by a computer run for the optimization criterion determined, formulated as the objective function of the model. In WRS the following objective functions are often used:

– maximization of technological parameters, e.g. the volume of water of the required quality that is produced by WRS,

– maximization of financial gross product, gross or net benefits,

– minimization of investment or operational costs of WRS,

– maximization of equivalent average return of investment costs of WRS or minimization of return period,

– a compromise objective function that minimizes the deviation of several functions from the extreme values. Different weights can be attributed to these functions.

For example, the mathematical model of the WRS on the River Kladénka and on the River Olšava in the south-eastern part of Moravia (ČSSR) was optimized by the following dynamic criterion:

$$\sum_{j\in I_1}\sum_{t=1}^{h} x_j z_{jt} v^t + \sum_{k\in I_2}\sum_{t=1}^{h} x_k z_{kt} v^t - \sum_{j\in I_1}\sum_{s=1}^{g} x_j c_{js} r^{(g-s)} -$$

$$- \sum_{k\in I_2}\sum_{s=1}^{g} x_k c_{ks} r^{(g-s)} \rightarrow \max ! \qquad (5.23)$$

where z are the incremental benefits in years of system operation for one activity unit,
c – investment costs in years of system construction for one activity unit,
r – interest rate factor $(1 + r_1)^t$; t – time in years, r_1 interest rate,
v – discount factor $(1 + r_1)^{-t}$,
g – the total construction time of the system,
s – individual years of construction of the system (index),
h – the average lifetime (planning horizon) of the system,
t – years of system operation.

This criterion expressed the requirement of obtaining the maximum difference between the total amount of discounted incremental benefits (present value of benefits) during the average lifetime of the system (starting from the end of the construction period), and the total amount of calculated investment costs (present value of investment costs) during the construction period. The computation of costs was performed to the end of the construction period.

The method of bivalued mixed integer linear programming described was successfully applied, and it is effective for large and complex systems. The model of the relatively simple WRS on the Rivers Kladénka and Olšava comprised approx. 300 equalities and inequalities with 220 variables. A further model of a rather simple WRS in the upper reaches of the River Svratka comprised 258 equalities and inequalities with 198 variables, five of which were bivalued. According to the authors of the method, modern computers can handle effectively models of up to a limit of 1000 to 1200 equalities and inequalities with 1000–1500 variables (with as many as 50 bivalued variables). The model for the River Dyje was developed for six existing reservoirs (completed or under construction) and a further 11 reservoirs in the design stage. The disadvantages of the method are the difficulties in handling longer hydrological time series. If monthly flows are used, optimization can be carried out for up to 3 to 5 years, since a significant increase in the length of the time series makes the dimensions of the model prohibitive. Therefore, for large systems the authors recommend the following method (involving a combination with a simulation model): The preliminary investigation is performed by a simulation model in long hydrological time series to determine one or several critical periods. Then optimization by bivalued mixed integer linear programming is carried out in only these critical periods. The resulting optimal solution is tested on a simulation model in longer time series.

The four linear models described above were illustrations of the various possibilities of applying linear programming in WRS. For a detailed description and further application of linear programming, see (e.g. Maass *et al.*, 1962; Buras, 1972; Kos, 1978, 1979; Loucks, 1981; Vedula and Rogers, 1981, etc.).

5.2 DYNAMIC PROGRAMMING

Dynamic programming is a very powerful, mathematically elegant, and well-known method of operations research. Its main advantage is the possibility of optimizing dynamic processes since the optimization is carried out in a long series of stages of the process, and the aim is to optimize control of the whole process. No restrictions are used for an objective function that need not be linear or analytical. Moreover, the stochastic nature of the processes investigated can be described easily and comprehensively.

These advantages compensate for some of its serious drawbacks. The method is complicated, and it requires a relatively broadly based investigation (it is a principle with many interpretations):

– in this case there are no standard programs for computers, unlike that of linear programming, i.e. for each task a new program must be elaborated and debugged;

– the models require the great memory capacity and high speed of computers; even with large computers a maximum of four independent parameters can be optimized by the basic method of discrete dynamic programming.

This condition determines the limits of the application of dynamic programming in WRS. For example, the maximum problem that can be optimized by dynamic programming¹) is a WRS with four reservoirs at a rough discriminating level, where the problem is reduced to control of the releases from these reservoirs.

Reference can be made to numerous publications dealing with dynamic programming: fundamental research was done by Bellman, 1957, 1961, 1963; Bellman and Dreyfus, 1962; Bellman and Kalaba, 1965; Vencel, 1964; Ter-Manuelianc, 1966; Walter *et al.*, 1973; Howard, 1960; and others. A brief description of the principles and methods of dynamic programming can be found in each publication dealing with operations research.

---

¹) In section 5.2.3 the limits of the problems are extended to 10, or even more, reservoirs by using special computational techniques.

There is an extensive literature on the applications of dynamic programming in water resource management: references given by Partl, 1968, and Buras, 1972, can be supplemented by references in Votruba, Nachàzel and Patera, 1974; Kartvelishvili, 1967, papers at symposia and conferences (e.g. Karlovy Vary, 1972) and a recent publication by Yakowitz, 1982.

5.2.1 Main Principles of Dynamic Programming

The *goal of dynamic programming* is to determine a policy for the whole process that maximizes or minimizes the objective function. The process is divided into a series of stages. This division is often done with reference to time (see Fig. 5.5). The computation is often performed in a recursive way, starting from the end, i.e. the last stage. In this *last stage*, for each possible state S of the system at the beginning of the last stage (i.e. at the end of the penultimate stage) the conditional optimal policy R for the last stage is determined. It is a policy that yields the required extreme value k^n for increments of the objective function at this last stage. Next, the *penultimate stage* is investigated. The conditional optimal policy is determined for each possible state of the system at the beginning of this stage. However, this condition

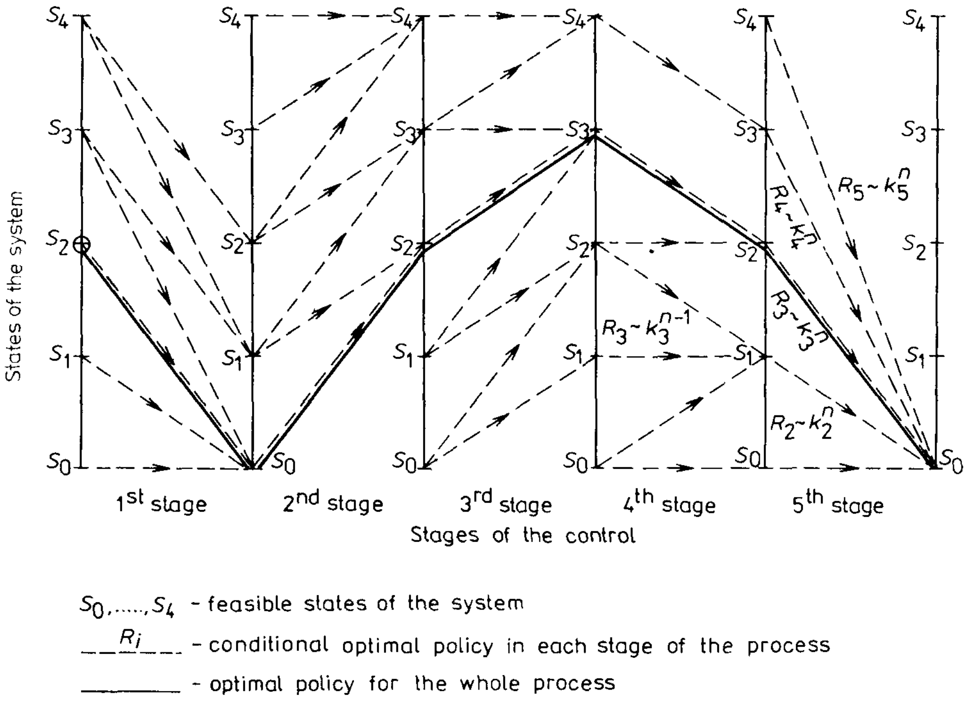


Fig. 5.5 The scheme of the method used in optimization of a five-stage process by dynamic programming

depends not only on increments of the objective function at this stage but on the sum of the last two stages. This relationship is allowed for in the following way:

The policy in the penultimate stage is sought, assuming that the state of the system at the beginning of this stage was S_2. If, for instance, a policy R_3 is used that gives in the penultimate stage the increment of the objective function k^{n-1}, the system at the end of the penultimate stage (i.e. at the beginning of the last stage) is in state S_1. For this state the conditional optimal policy in the last stage R_2 has been determined and the corresponding increment of the objective function k_2^n is known. If the sum of both increments $k_3^{n-1} + k_2^n$ is maximal, then the investigated policy R_3 in the penultimate stage is the conditional optimal policy for the state S_2 at the beginning of this stage. Similarly, the conditional optimal policy in the last two stages is determined in the penultimate stage for all possible states $S_0, \ldots, S_4$.

The method is repeated for the third, fourth, and further stages (counting from the end) until the beginning of the process is reached. At each stage, for all possible states of the system at the beginning of the stage, a conditional optimal policy is determined that gives the maximum (minimum) of the sum of the increments of the objective function at this stage and the objective function in all remaining stages of the process that were determined using the optimal policy in these remaining stages. In Fig. 5.5 the conditional optimal policy is marked with a dashed line.

Then follows the second part of the computation (in Fig. 5.5 the full broken line). Starting from the given (or chosen) state of the system, e.g. S_2, the optimal policy for the whole process is determined, based on all the computed conditional optimal policies at the individual stages (proceeding from the beginning to the end) that give the required extreme values of the objective function for the whole process. This second part of the procedure is carried out quite mechanically in Fig. 5.5. For certain initial states, e.g. S_3, there are several equivalent alternatives with the same total values of the objective function.

With this method, the determination of the optimal policy at a particular stage need not take into account all the possible consequences of this chosen policy for the rest of the process; only the chosen optimal policies in the rest of the process are considered. In this way the number of alternatives that must be compared in optimization is greatly reduced. For example, a very short process consisting of 20 stages with a system of 10 states provides 10^{20} possible alternative policies. The computation and comparison of all these alternatives is beyond the scope of even the largest computers. With the procedure described, the number of alternatives compared is reduced to $10 \cdot 20 = 200$ (i.e. the exponent is changed into a multiplier), comparison of which can be easily performed.

This procedure was developed and precisely formulated by Bellman, 1957. It can be expressed by the principle of optimality, which can be simplified to the following equation:

$$R_i^{\mathrm{opt}} \sim \max K_i^n = \max \left[k_i + K_{i+1}^{n,\mathrm{opt}}\right] \tag{5.24}$$

where R_i^{opt} is the optimal policy in stage i,

$\max K_i^n$ – the maximum value of the objective function from stage i to stage n (i.e. the end of the process),

k_i – the incremental objective function in stage i of the process,

$K_{i+1}^{n,\text{opt}}$ – the value of the objective function from stage $i + 1$ to stage n using the optimal policy at all these stages.

The principle of optimality can be formulated as follows: The optimal policy R_i^{opt} at stage i is a policy that maximizes the sum of the increments of the objective function at stage k_i and the objective function $K_{i+1}^{n,\text{opt}}$ in the remaining stages of this process using the optimal policy determined. Bellman's classical formulation of the principles of optimality is as follows: An optimal policy must have the property that, regardless of the route taken to enter a particular state, the remaining decisions must constitute an optimal policy for leaving that state.

5.2.2 Application of Dynamic Programming in WRS

The above-stated principle of optimality has been used for the solution of various problems in WRS. Well-known applications are those described by Little, 1955; Cvetkov, 1961; Maass *et al.*, 1962; Kartvelishvili, 1967; Buras, 1972. Recent applications are mentioned in section 5.2.3. The possibilities of this principle of optimality are apparently not yet exhausted and new approaches keep appearing.

A two-dimensional dynamic program

Firstly a simple model conforming to the scheme in Fig. 5.6 will be analysed. It includes two reservoirs and withdrawal for an industrial area. The aim is to determine the optimal river flow regulation at a point P at a distance from both reservoirs; the constraints are the given withdrawals for the industrial area. The operational policy further includes flood control and recreation which specifies flood control storage in both reservoirs, and in the recreation period, a certain minimum pool for recreation purposes is required. The runoff from an interbasin, schematized as tributary inflow, cannot be controlled. The dynamic programming approach is as follows:

– the releases from both reservoirs for the whole process are given by the optimization of river flow regulation at point P,

– the requirements of water supply for industry, flood control and recreation are used as constraints,

– as only two independent parameters are optimized (the operational policy of two reservoirs), a two-dimensional task is formulated that can be handled easily by computers.

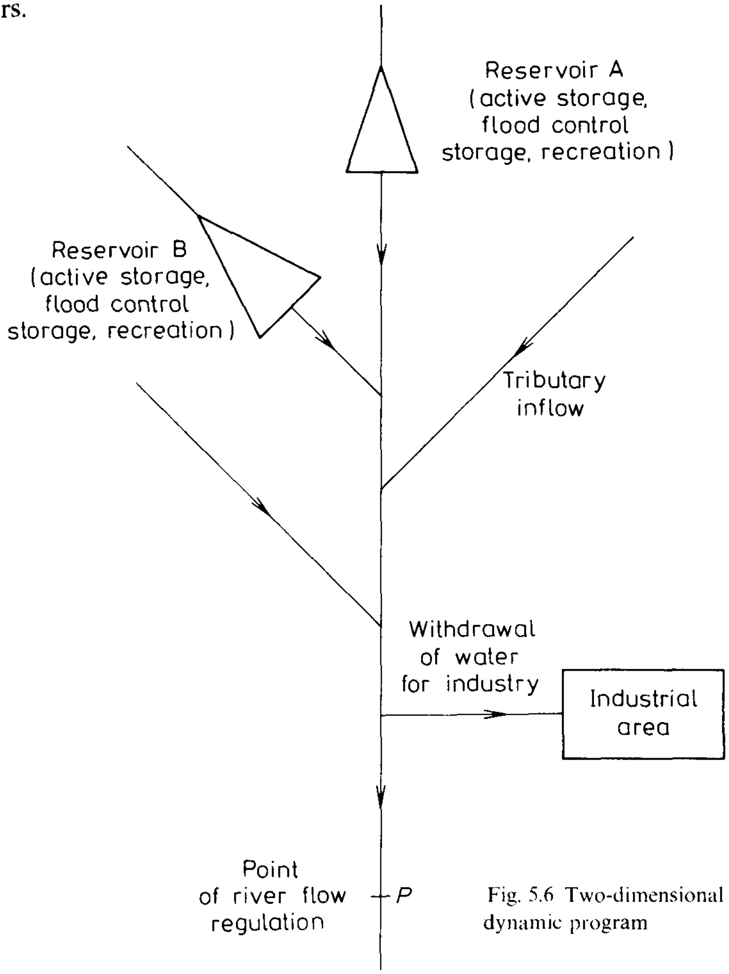


Fig. 5.6 Two-dimensional dynamic program

The choice of the form of objective function should be specified for each task individually. In this case, when optimal river flow regulation at point P was required, a parabolic function of the type $y^2 = 2x$ (Fig. 5.7) was used. This function minimized effectively both extreme values of flows, minimization of the upper values being rather more effective.

The computation was done for a period of thirty years of observed monthly flows (1931 – 1960) i.e. in monthly stages. The formulation was deterministic as the values of the flows were considered as known for the whole period in question. The optimal operational policy obtained cannot be implemented in practice; river flow regulation at point P is the optimum limited by releases from the two reservoirs A and B. Practical result: the computation showed the limited possibilities of river flow regulation at a point P at a distance from reservoirs A and B (due to the relatively large unregulated tributary flow) and proved that this WRS was not efficient.

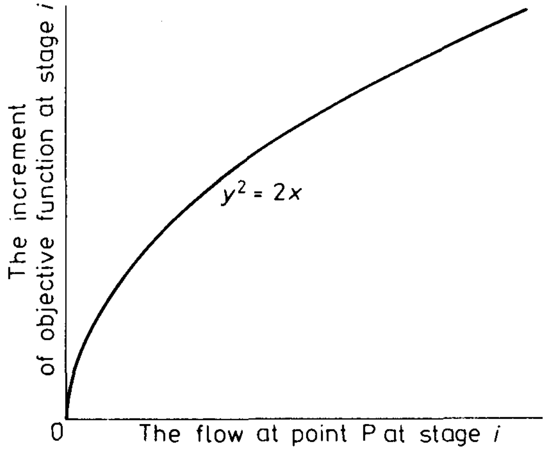


Fig. 5.7 The objective function for a two-dimensional dynamic program

The computation can be illustrated by Table 5.3 where, however, data have been simplified. It is valid from stage $n - 3$, i.e. the fourth from the end. It represents the first part of computation, i.e. determination of a conditional optimal policy (decisions) for all possible states of the system at the beginning of this stage. In this example, the state of the system is defined by the actual storage (contents) of the reservoir[1]).

Assuming that the actual storage (contents) of each reservoir can acquire ten different values (e.g. 0, 10, 20, ..., 90 mil. m^3), this system can be in $10^2 = 100$ states. The policy is determined by releases from both reservoirs at this stage. If 10 different values are assumed for each reservoir, the number of possible decisions equals 100.

In the upper part of Table 5.3 there are three numbers in each field. The first is the increment of objective function at this stage, determined by decision at this stage. The second number is the sum of the objective functions at the remaining stages until the end of the process (it has been determined by the same procedure). The third number is the sum of the first and the second. For each state the maximum sum (denoted by asterisks) determines the conditional operational policy at this stage. The free field contains the stages and states excluded by the constraints.

---

[1]) In other examples the state of the system can be characterized by other parameters, e.g. by values of flows at certain points, etc.

Table 5.3 Scheme of the computation of a conditional optimal operational policy for stage $(n-3)$

	State of the system at the beginning of the stage $(n-3)$					
	0	–	–	–	–	–
Operational policy at the stage $(n-3)$	1	10 95 105	–	–	–	–
	2	19 88 107	19 95 114	–	–	–
	3	27 81 * 108	27 88 * 115	27 95 * 122		–
	4	34 73 107	34 81 * 115	34 88 * 122	34 95 * 129	–
	5	40 65 105	40 73 113	40 81 121	40 88 128	40 95 * 135
	6	–	45 65 110	45 73 118	45 81 126	45 88 133
	7	–	–	49 65 122	49 73 122	49 81 130
	8	–	–	–	52 65 117	52 73 125

* maximum sum

State at the beginning of the stage	0	1	2	3	4
Conditional optimal operational policy at the stage	3	3; 4	3; 4	4	5
Increment of objective function at the stage	108	115	122	129	135
State at the end of the stage	2	3; 2	4; 3	4	4

The lower part of the table summarizes the results. Similar tables are computed for all stages from stage *n* to stage 1. Then, for the known (or chosen) state at the beginning of the process, the optimal policy for the whole process is determined using the lower part of these tables.

An illustration of the results is given in Fig. 5.8; for a detailed description see Partl, 1968, 1969; Palla and Partl, 1970.

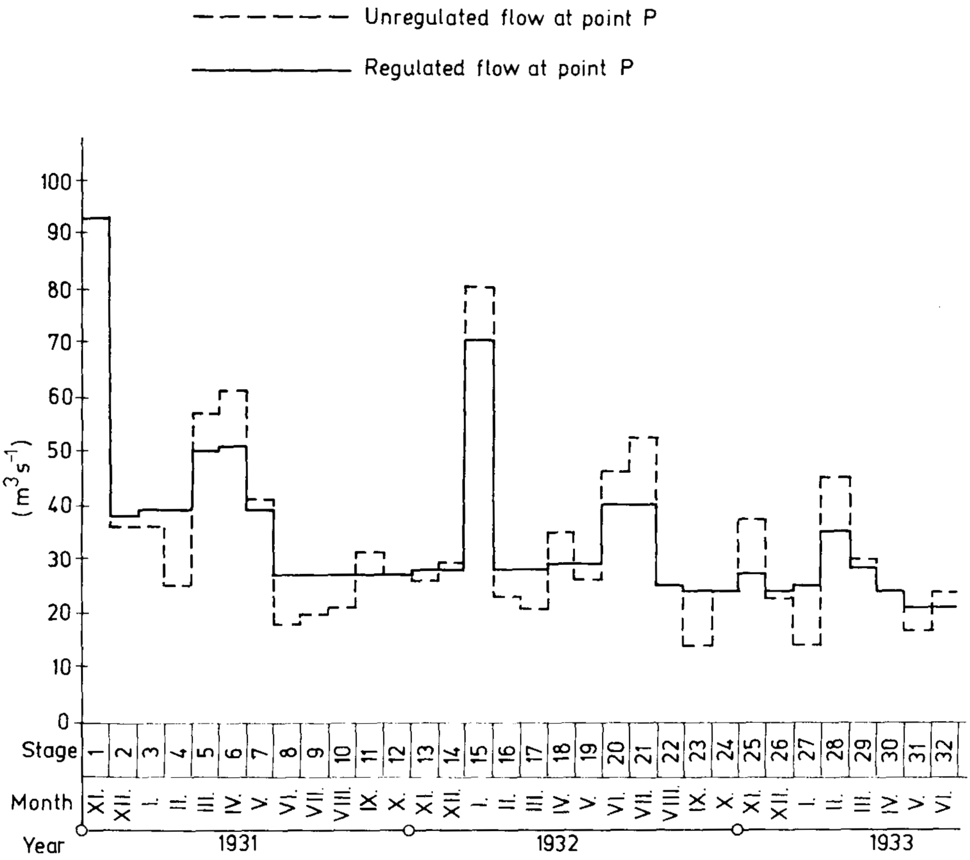


Fig. 5.8 River flow regulation by two reservoirs (optimal yield by parabolic objective function)

This example for 360 monthly stages was computed on a National Elliott 4120 computer. The computation of one stage took 15 seconds. The duration of computation is a linear function of the number of stages and an exponential function of the number of optimized parameters. The computation of a three-dimensional problem (e.g. combination of three reservoirs) took several hours and required the external memory of the computer.

A three-dimensional dynamic program

The WRS described in Fig. 5.9 has been analyzed. It contains three reservoirs. Reservoir A is used for supplying water to an industrial area, reservoir C for irrigation.

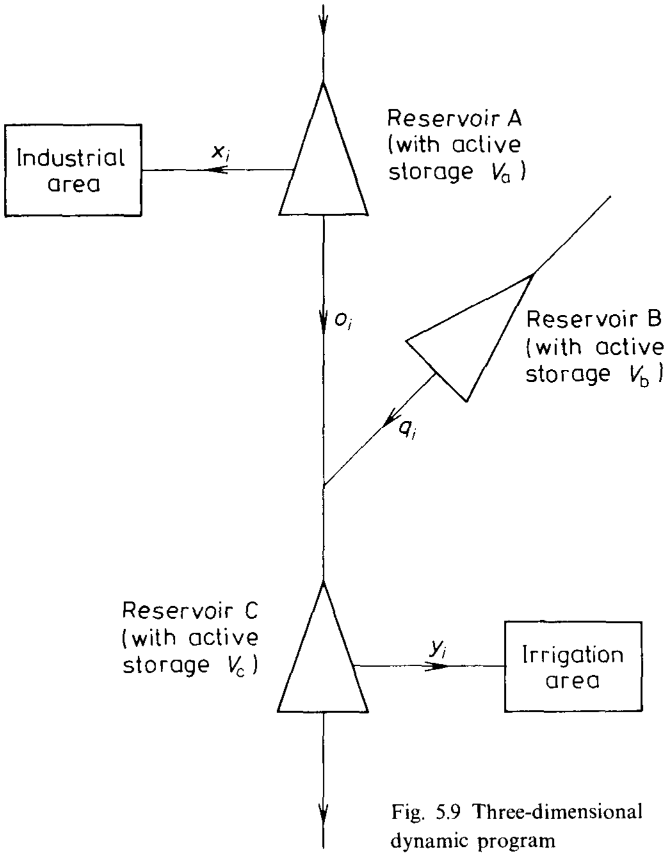


Fig. 5.9 Three-dimensional dynamic program

The economically optimal operational policy is to be determined, i.e. for each stage the releases from reservoirs A and B and withdrawal for the irrigation area that maximize the objective function for the whole period of operation. The withdrawals for industry are not optimized and are therefore not used in the objective function. They are predetermined for each stage.

The relationship between water supplies for irrigation and agricultural benefits is given. The following notation is used:

x_i – withdrawal of water for the industrial area at stage i (it is predetermined for each stage),
o_i – the release from reservoir A at stage i,
q_i – the release from reservoir B at stage i,

y_i – the withdrawal for irrigation at stage i,
a_i – actual storage (contents) of reservoir A at the beginning of stage i,
b_i – actual storage (contents) of reservoirs B at the beginning of stage i,
c_i – actual storage (contents) of reservoir C at the beginning of stage i.

The state vector has three components, a_i, b_i and c_i at each moment; it constitutes a three-dimensional problem.

Table 5.4 Results of the optimization of a three-dimensional dynamic program in a tabular form

Stage $n = 9$						
State of the system at the beginning of stage			Optimal policy at the stage			Value of objective function (mil. $)
Actual storage (content) of reservoirs $[\text{mil. m}^3]$			Releases from reservoirs $[\mathrm{m}^3\,\mathrm{s}^{-1}]$			
a_i	b_i	c_i	o_i	q_i	y_i	

The objective function $f_n(a, b, c)$ is defined as the expected benefit of the n-stage process using an optimal operational policy.

The stochastic nature of unregulated inflows in reservoirs is reflected in the following way: in computation of the optimal operational policy, the mean values (or quantiles, e.g. 10% quantiles) of the observed hydrological series are used. In the application of operating policy this simplification is corrected by a combination of operating policy with the actual state of the system at the beginning of the stage.

In optimization, many constraints must be complied with. These constraints are similar to those used in linear programming. In this example, the following constraints were imposed for every stage:

– the required withdrawal for industry should be met,

– the releases from all reservoirs must not decrease below the given minimum values,

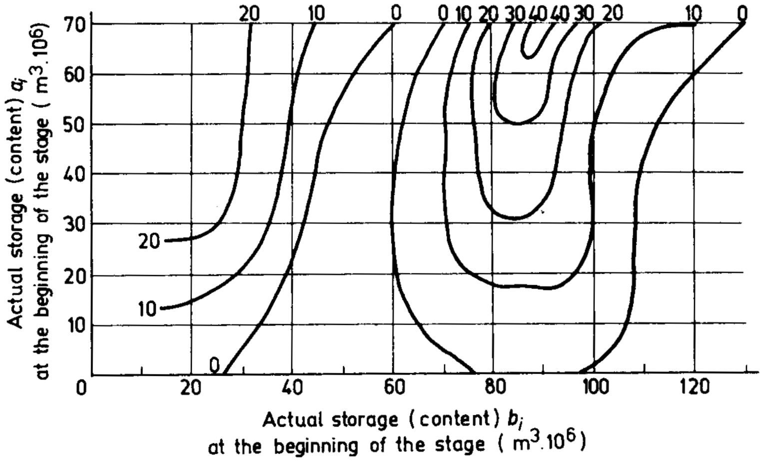
– the withdrawal for irrigation must not decrease below the required minimum values,

– the actual storage (contents) of reservoirs must not be negative and cannot exceed their active storage, etc.

The result of the solution for each stage is the determination of the optimal decision vector with three components o_i, q_i, y_i.

The results for stage $n = 9$ are given in tabular form in Table 5.4. The three left-

a) Optimal release o_i ($m^3 s^{-1}$) at stage i for c_i = 10 mil. m^3 at the beginning of the stage (stage $n = 9$)



b) Optimal release o_i ($m^3 s^{-1}$) at stage i for c_i = 20 mil. m^3 at the beginning of the stage (stage $n = 9$)

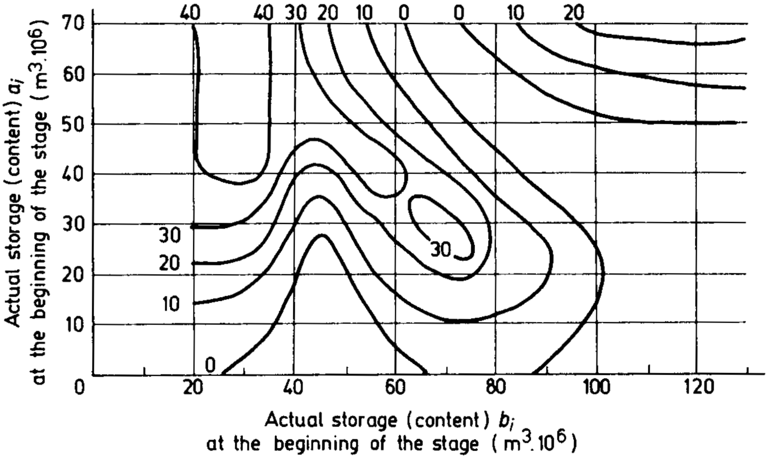


Fig. 5.10 The results of the optimization of a three-dimensional dynamic program in graphic form

hand columns of the table contain the states of the system, i.e. all possible combinations of actual reservoir storages.

If each reservoir is assumed to have ten possible states, this table has $10^3 = 1000$ rows. Sometimes the results are presented in graphical form. It is more instructive and interpolation is facilitated. Figure 5.10 provides an illustration of two graphs for stage $n = 9$. In the first graph, the actual storage (contents) of the reservoirs C at the beginning of the stage $c_i = 10$ mil. m^3; in the second graph, $c_i = 20$ mil. m^3. For the example under discussion, with 1000 possible states of the system, 30 graphs would be necessary for each stage. This number is acceptable, e.g. for an annual operational policy with weekly stages. Application of these results in graphical form is clear and easy.

Similar models have been applied by many authors (Little, 1955; Cvetkov, 1961; Buras, 1972) on computers of high speed and great memory capacity.

Dynamic program of a multi-purpose WRS

This model shows that a relatively complex WRS can be investigated by mathematical programming. The schema of the WRS is given in Fig. 5.11. Similar or even more complex WRS problems have been solved in this way (Parikh, 1966; Hall–Shephard, 1967).

The aim of the schematized WRS is:

- the production of power in three hydroelectric power plants,
- a supply of water for three irrigation systems; one system is supplied from a diversion point where the river flows are regulated,
- a municipal water supply from reservoir D,
- navigation in the main stream as far as the control point,
- the required water quality in watercourses.

As most elements of the system have already been built, or are being constructed, formulation of the problem is as follows:

a) estimation of the maximum water supply and power production that can be guaranteed by the WRS,

b) an operation policy that yields a maximum economic return.

With regard to the capacities of the computers, the use of monthly stages for a period of 10 years is acceptable. Therefore, a critical 10-year period is usually chosen from the hydrological series as a sample, which is used for computation of the WRS model. The method is as follows:

- the initial prices of water and energy are determined,
- the whole system is divided into subsystems, in our case α, β, γ, δ,
- for each subsystem the optimal operating policy is determined by dynamic programming,

– linear programming is used for determination of the operating policy of the whole system that maximizes the net benefits[1]).

– the linear program gives a set of shadow prices that are used as input values for the second run of the dynamic optimization of the four subsystems. New operational rules for releases of water from reservoirs and new optimal values of water supply and power production are thus generated in all subsystems,

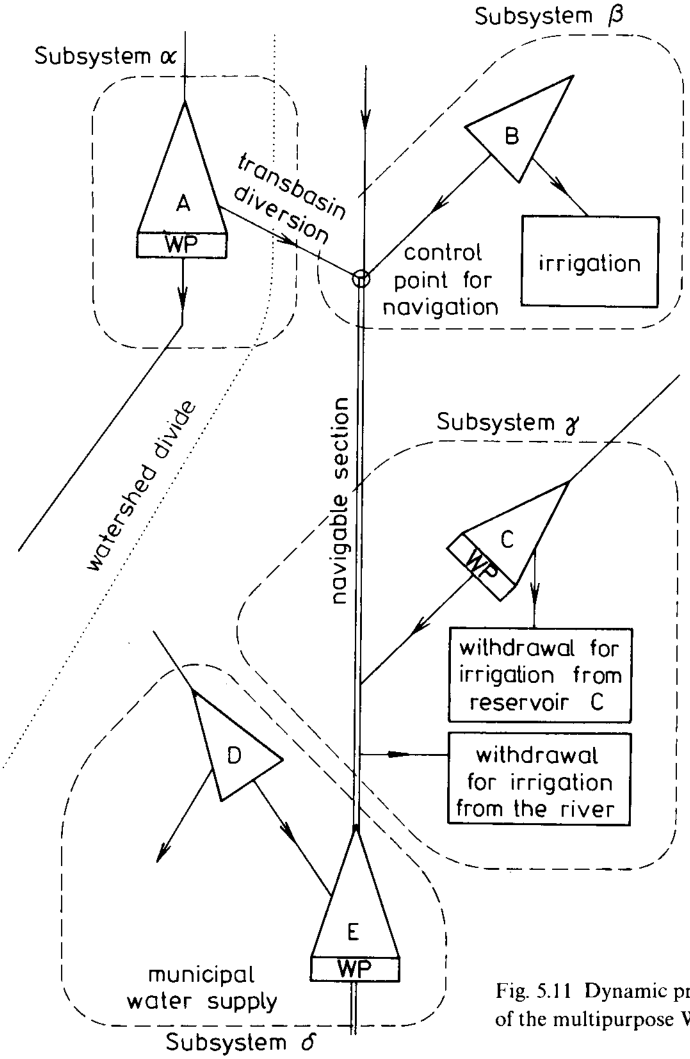


Fig. 5.11 Dynamic program of the multipurpose WRS

---

[1]) This linear program is often very extensive; it is therefore subdivided in accordance with the decomposition principle (Maass *et al.*, 1962; Dantzig, 1963).

– these results are used for a new run of the linear program for the whole system for the maximization of net benefits from water supply and power production.

The iterative procedure proceeds till the increment of the objective function of the linear program is lower than the given limit. The scheme of computation is given in Fig. 5.12.

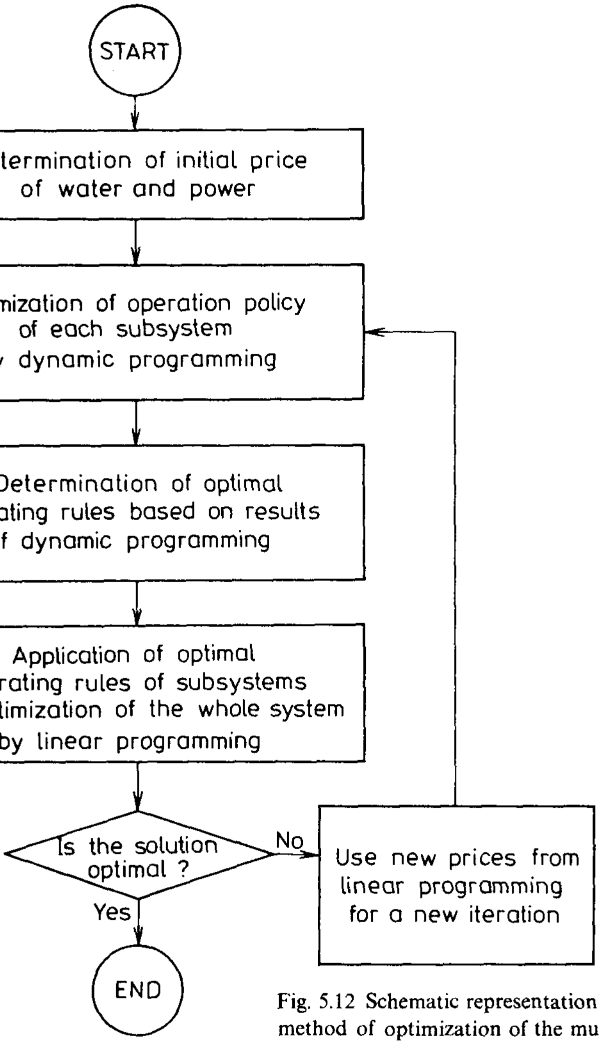


Fig. 5.12 Schematic representation of iteration method of optimization of the multipurpose WRS

This example shows the possibilities of combination of different methods of mathematical programming.

Non-linear programming and similar methods of mathematical programming are not discussed in this book as their practical application in WRS has not been reported yet.

5.2.3 The "Curse of Dimensionality" in Dynamic Programming

The computer technique of dynamic programming described so far was discrete dynamic programming. We have mentioned the exponential growth in memory and time requirements with the increase in the number of dimensions of the state vector. Three dimensions are the maximum that can be handled without difficulty by this method. Heidari *et al.*, 1971, considered a four-reservoir problem (see Fig. 5.13) which is probably the limit of this method of discrete dynamic programming because of the "curse of dimensionality".

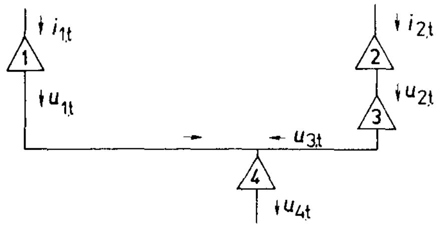


Fig. 5.13 Schematic representation of a four-reservoir problem
$i_{k,t}$ – inflow in reservoir k at stage t, $u_{k,t}$ – release of reservoir k at stage t

A new computational technique of dynamic programming called "discrete differential dynamic programming" has been proposed. It is a successive approximation algorithm using the term "corridor" in which computation is performed. This corridor substantially reduces the dimensions of the problem. However, the computation must be repeated if the conditional optimal policy reaches the limits of this corridor. The drawback of this method, and other subsequent computational techniques, is that the sequence of policies can converge to a local optimum or can diverge unless the control problem satisfies certain stringent assumptions. Chow *et al.*, 1975, used the same four-reservoir problem to text the computer time and memory requirements of discrete differential dynamic programming. In this technique the computation requirements were reduced, but the curse of dimensionality

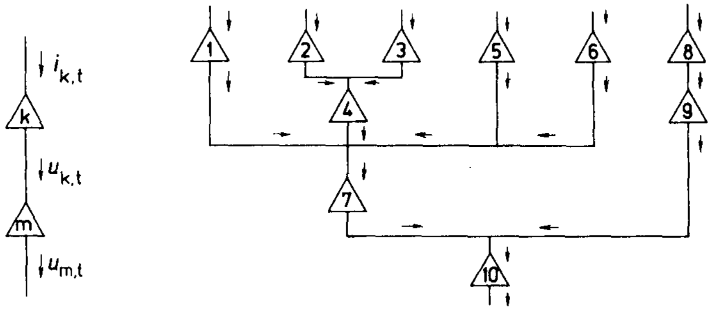


Fig. 5.14 Schematic representation of a ten-reservoir problem
$i_{k,t}$ – inflow to reservoir k at stage t, $u_{k,t}$; $u_{m,t}$ – releases from reservoirs k and m resp. at stage t

still remains. For example, a ten-reservoir problem is beyond the scope of this method.

Murray and Yakowitz, 1979, used this four-reservoir scheme for comparison with a ten-reservoir problem (see Fig. 5.14) and proposed the computational method called constrained differential programming. This method requires for its input a "good" policy (i.e. initial estimate of optimal policy) supplied by the user. The resulting algorithm describes how to compute a better policy given the initial policy.

Another method that can sidestep the curse of dimensionality is state incremental dynamic programming. Trott and Yeh, 1973, described the details of this method and used it for operations of a reservoir system. The principle of the method is the optimization of the releases of the *i*-th reservoir while the contents of the remaining reservoirs are kept constant throughout the decision horizon. This constraint reduces the dimensionality to one dimension. In successive iterations of the method the value of the fixed reservoir contents changes.

The main assumption of the method is that the dynamic relationship

$$x_{t+1} = f_t(x_t, u_t) \tag{5.25}$$

where x_t is the state at stage t and u_t is the policy at stage t can be inverted in u_t

$$u_t = g_t(x_t, x_{t+1}) \tag{5.26}$$

Thus, state incremental dynamic programming is essentially dynamic programming with a univariate state space.

Differential dynamic programming and state incremental dynamic programming have much greater computational possibilities than discrete dynamic programming, e.g. Yakowitz and Rutherford, 1981, solved a dynamic programming problem with as many as 40 state variables.

This remarkable progress was achieved in deterministic dynamic programming. In stochastic dynamic programming it is not possible to use these methods, and original discrete dynamic programming solution within or outside the water resource literature has the problem of having, at most, two or three state variables. For example, Schweig and Cole, 1968, used a very rough discretization for two state variables and reported severe computational difficulties. However, as water resource problems have served as a stimulus to the development of dynamic programming itself, especially stochastic programming, it is hoped that further mathematical and modelling research will solve the problem of dimensionality even in stochastic dynaming programming.