18 Optimization

18.1 Linear Programming

18.1.1 Formulation of the Problem and Geometrical Representation

18.1.1.1 The Form of a Linear Programming Problem

1. The Subject

of *linear programming* is the minimization or maximization of a *linear objective function* (**OF**) of finitely many variables subject to a finite number of *constraints* (**CT**), which are given as linear equations or inequalities.

Many practical problems can be directly formulated as a linear programming problem, or they can be modeled approximately by a linear programming problem.

2. General Form

A linear programming problem has the following general form:

$$\textbf{OF:}\quad f(\underline{\mathbf{x}}) = c_1x_1 + \cdots + c_rx_r + c_{r+1}x_{r+1} + \cdots + c_nx_n = \max! \tag{18.1a}$$

$$\textbf{CT:}\quad \left.\begin{array}{rcccccl} a_{1,1}x_1 & +\cdots+ & a_{1,r}x_r & + & a_{1,r+1}x_{r+1} & +\cdots+ & a_{1,n}x_n \le b_1,\\ \vdots & & \vdots & & \vdots & & \vdots\\ a_{s,1}x_1 & +\cdots+ & a_{s,r}x_r & + & a_{s,r+1}x_{r+1} & +\cdots+ & a_{s,n}x_n \le b_s,\\ a_{s+1,1}x_1 & +\cdots+ & a_{s+1,r}x_r & + & a_{s+1,r+1}x_{r+1} & +\cdots+ & a_{s+1,n}x_n = b_{s+1},\\ \vdots & & \vdots & & \vdots & & \vdots\\ a_{m,1}x_1 & +\cdots+ & a_{m,r}x_r & + & a_{m,r+1}x_{r+1} & +\cdots+ & a_{m,n}x_n = b_m,\\ & & & & & & \\ x_1 \ge 0, \ldots, x_r \ge 0; & & & & x_{r+1}, \ldots, x_n & \text{free.} & \end{array}\right\} \tag{18.1b}$$

In a more compact vector notation this problem becomes:

$$\textbf{OF:}\quad f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}^{1\mathrm{T}}\underline{\mathbf{x}}^1 + \underline{\mathbf{c}}^{2\mathrm{T}}\underline{\mathbf{x}}^2 = \max! \tag{18.2a}$$

$$\textbf{CT:}\quad \left.\begin{array}{l} \mathbf{A}_{11}\underline{\mathbf{x}}^1 + \mathbf{A}_{12}\underline{\mathbf{x}}^2 \le \underline{\mathbf{b}}^1,\\ \mathbf{A}_{21}\underline{\mathbf{x}}^1 + \mathbf{A}_{22}\underline{\mathbf{x}}^2 = \underline{\mathbf{b}}^2,\\ \underline{\mathbf{x}}^1 \ge 0,\ \underline{\mathbf{x}}^2 \text{ free.} \end{array}\right\} \tag{18.2b}$$

Here, the following notations are used:

$$\underline{\mathbf{c}}^1 = \begin{pmatrix} c_1\\ c_2\\ \vdots\\ c_r \end{pmatrix},\ \underline{\mathbf{c}}^2 = \begin{pmatrix} c_{r+1}\\ c_{r+2}\\ \vdots\\ c_n \end{pmatrix}, \qquad \underline{\mathbf{x}}^1 = \begin{pmatrix} x_1\\ x_2\\ \vdots\\ x_r \end{pmatrix},\ \underline{\mathbf{x}}^2 = \begin{pmatrix} x_{r+1}\\ x_{r+2}\\ \vdots\\ x_n \end{pmatrix}, \tag{18.2c}$$

$$\mathbf{A}_{11} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1,r}\\ a_{21} & a_{22} & \cdots & a_{2,r}\\ \vdots & & & \\ a_{s,1} & a_{s,2} & \cdots & a_{s,r} \end{pmatrix}, \qquad \mathbf{A}_{12} = \begin{pmatrix} a_{1,r+1} & a_{1,r+2} & \cdots & a_{1,n}\\ a_{2,r+1} & a_{2,r+2} & \cdots & a_{2,n}\\ \vdots & & & \\ a_{s,r+1} & a_{s,r+2} & \cdots & a_{s,n} \end{pmatrix}, \tag{18.2d}$$

$$\mathbf{A}_{21} = \begin{pmatrix} a_{s+1,1} & a_{s+1,2} & \cdots & a_{s+1,r}\\ a_{s+2,1} & a_{s+2,2} & \cdots & a_{s+2,r}\\ \vdots & & & \\ a_{m,1} & a_{m,2} & \cdots & a_{m,r} \end{pmatrix}, \qquad \mathbf{A}_{22} = \begin{pmatrix} a_{s+1,r+1} & a_{s+1,r+2} & \cdots & a_{s+1,n}\\ a_{s+2,r+1} & a_{s+2,r+2} & \cdots & a_{s+2,n}\\ \vdots & & & \\ a_{m,r+1} & a_{m,r+2} & \cdots & a_{m,n} \end{pmatrix}. \tag{18.2e}$$

3. Constraints

with the inequality sign " $\ge$ " will have the above form if they are multiplied by (-1).

I.N. Bronshtein et al., *Handbook of Mathematics*,
DOI 10.1007/978-3-662-46221-8_18

4. Minimum Problem

A minimum problem $f(\underline{\mathbf{x}}) = \min!$ becomes an equivalent maximum problem by multiplying the objective function by (-1)

$$-f(\underline{\mathbf{x}}) = \max! \tag{18.3}$$

5. Integer Programming

Sometimes certain variables are restricted to be only integers. This discrete problem is not discussed here.

6. Formulation with only Non-Negative Variables and Slack Variables

In applying certain solution methods, only non-negative variables are considered, and constraints (18.1b), (18.2b) given in equality form.

Every free variable x_k must be decomposed into the difference of two non-negative variables $x_k = x_k^1 - x_k^2$. The inequalities become equalities by adding non-negative variables; they are called *slack variables*. That is, the problem is considered in the form as given in (18.4a,b), where n is the increased number of variables. In vector form:

$$\mathbf{OF}: f(\underline{\mathbf{x}}) = c_1 x_1 + \cdots + c_n x_n = \max! \tag{18.4a}$$

$$\mathbf{CT}: \quad \left.\begin{array}{rcl} a_{1,1}x_1 + \cdots + a_{1,n}x_n &=& b_1, \\ \vdots \qquad\qquad\quad & & \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n}x_n &=& b_m, \\ x_1 \geq 0, \ldots, x_n \geq 0. & & \end{array}\right\} \tag{18.4b}$$

$$\mathbf{OF:} \quad f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}} = \max! \tag{18.5a}$$

$$\mathbf{CT:} \quad \mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}\,, \quad \underline{\mathbf{x}} \geq \underline{\mathbf{0}}\,. \tag{18.5b}$$

The relation $m \leq n$ can be supposed, otherwise the system of equations contains linearly dependent or contradictory equations.

7. Feasible Set

The set of all vectors $\underline{\mathbf{x}}$ satisfying constraints (18.2b) is called the *feasible set* of the original problem. If the free variables are rewritten as above, and every inequality of the form "$\leq$" into an equation as in (18.4a) and (18.4b), then the set of all non-negative vectors $\underline{\mathbf{x}} \geq \underline{\mathbf{0}}$ satisfying the constraints is called the *feasible set* M:

$$M = \{\underline{\mathbf{x}} \in \mathbb{R}^n : \underline{\mathbf{x}} \geq \underline{\mathbf{0}},\ \mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}\}. \tag{18.6a}$$

A point $\underline{\mathbf{x}}^* \in M$ with the property

$$f(\underline{\mathbf{x}}^*) \geq f(\underline{\mathbf{x}}) \quad \text{for every } \underline{\mathbf{x}} \in M \tag{18.6b}$$

is called the *maximum point* or the *solution point* of the linear programming problem. Obviously, the components of $\underline{\mathbf{x}}$ not belonging to slack variables form the solution of the original problem.

18.1.1.2 Examples and Graphical Solutions

1. Example of the Production of Two Products

Suppose primary materials R_1, R_2, and R_3 are needed to produce two products E_1 and E_2. **Scheme 18.1** shows how many units of primary materials are needed to produce each unit of the products E_1 and E_2, and there are given also the available amount of the primary materials.

Selling one unit of the products E_1 or E_2 results in 20 or 60 units of profit, respectively (PR). Determine a production program which yields maximum profit, if at least 10 units must be produced from product E_1.

Denoting by x_1 and x_2 the number of units produced from E_1 and E_2, the problem is:

Scheme 18.1	R_1 / E_i	R_2 /E_i	R_3 /E_i
E_1	12	8	0
E_2	6	12	10
Amount	630	620	350

$$\begin{array}{lrcl} \mathbf{OF:} & f(\underline{\mathbf{x}}) = 20x_1 + 60x_2 &=& \max! \\ \mathbf{CT:} & 12x_1 + 6x_2 &\leq& 630, \\ & 8x_1 + 12x_2 &\leq& 620, \\ & 10x_2 &\leq& 350, \\ & x_1 &\geq& 10. \end{array}$$

Introducing the slack variables x_3, x_4, x_5, x_6, one gets:

$$\begin{aligned}
&\mathbf{OF}: \quad f(\underline{\mathbf{x}}) = 20x_1 + 60x_2 + 0\cdot x_3 + 0\cdot x_4 + 0\cdot x_5 + 0\cdot x_6 = \max!\\
&\mathbf{CT}: \quad \begin{aligned}
12x_1 + 6x_2 + x_3 &= 630,\\
8x_1 + 12x_2 + x_4 &= 620,\\
10x_2 + x_5 &= 350,\\
-x_1 + x_6 &= -10.
\end{aligned}
\end{aligned}$$

2. Properties of a Linear Programming Problem

On the basis of this example, some properties of the linear programming problem can be demonstrated by graphical representation. Here the slack variables are not considered; only the original two variables are used.

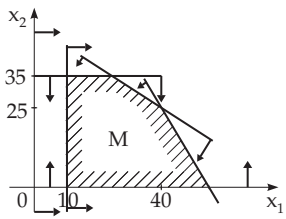

Figure 18.1

a) A line $a_1x_1+a_2x_2 = b$ divides the x_1, x_2 plane into two half-planes. The points (x_1, x_2) satisfying the inequality $a_1x_1+a_2x_2 \le b$ are in one of these half-planes. The graphical representation of this set of points in a Cartesian coordinate system can be made by a line, and the half-plane containing the solutions of the inequalities is denoted by an arrow. The set of feasible solutions M, i.e., the set of points satisfying all inequalities is the intersection of these half-planes **(Fig. 18.1)**.

In this example the points of M form a polygonal domain. It may happen that M is unbounded or empty. If more then two boundary lines go through a vertex of the polygon, this vertex is called a degenerate vertex **(Fig. 18.2)**.

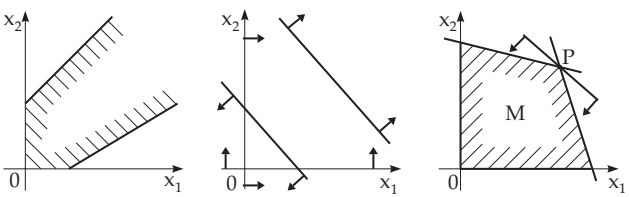

Figure 18.2

b) Every point in the x_1, x_2 plane satisfying the equality $f(x) = 20x_1 + 60x_2 = c_0$ is on one line, on the level line associated to the value c_0. With different choices of c_0, a family of parallel lines is defined, on each of which the value of the objective function is constant. Geometrically, those points are the solutions of the programming problem, which belong to the feasible set M and also to the level line $20x_1 + 60x_2 = c_0$ with maximal value of c_0. In this example, the solution point is $(x_1, x_2) = (25, 35)$ on the line $20x_1 + 60x_2 = 2600$. The level lines are represented in **Fig. 18.3**, where the arrows point in the direction of increasing values of the objective function.

Obviously, if the feasible set M is bounded, then there is at least one vertex such that the objective function takes its maximum. If the feasible set M is unbounded, it is possible that the objective function is unbounded, as well.

18.1.2 Basic Notions of Linear Programming, Normal Form

Now, the problem (18.5a,b) is considered with the feasible set M .

18.1.2.1 Extreme Points and Basis

1. Definition of the Extreme Point

A point $\underline{\mathbf{x}} \in M$ is called an *extreme point* or *vertex* of M, if for all $\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2 \in M$ with $\underline{\mathbf{x}}_1 \neq \underline{\mathbf{x}}_2$:

$$\underline{\mathbf{x}} \neq \lambda \underline{\mathbf{x}}_1 + (1-\lambda)\underline{\mathbf{x}}_2, \quad 0 < \lambda < 1, \tag{18.7}$$

i.e., $\underline{\mathbf{x}}$ is not on any line segment connecting two different points of M.

2. Theorem about Extreme Points

The point $\underline{\mathbf{x}} \in M$ is an *extreme point* of M if the columns of matrix $\mathbf{A}$ associated to the positive components of $\underline{\mathbf{x}}$ are linearly independent.

If the rank of $\mathbf{A}$ is m, then the maximal number of independent columns in $\mathbf{A}$ is m. So, an extreme point can have at most m positive components. The other components, at least $n-m$, are equal to zero. In the usual case, there are exactly m positive components. If the number of positive components is less then m, it is called a *degenerate extreme point*.

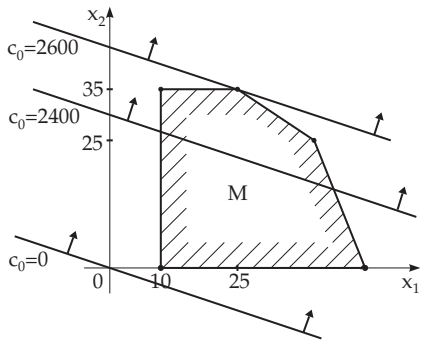

Figure 18.3

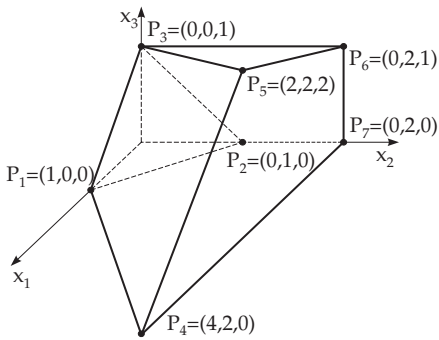

Figure 18.4

3. Basis

To every extreme point m linearly independent column vectors of the matrix $\mathbf{A}$ can be assigned, the columns belonging to the positive components. This system of linearly independent column vectors is called *the basis of the extreme point*. Usually, exactly one basis belongs to every extreme point. However several bases can be assigned to a degenerate extreme point. There are at most $\binom{n}{m}$ possibilities to choose m linearly independent vectors from n columns of $\mathbf{A}$. Consequently, the number of different bases, and therefore the number of different extreme points is $\binom{n}{m}$. If M is not empty, then M has at least one extreme point.

■ **OF:** $f(\underline{\mathbf{x}}) = 2x_1 + 3x_2 + 4x_3 = \max!$

$$\textbf{CT:} \quad \begin{aligned} x_1 + x_2 + x_3 &\geq 1, \\ x_2 &\leq 2, \\ -x_1 + 2x_3 &\leq 2, \\ 2x_1 - 3x_2 + 2x_3 &\leq 2. \end{aligned} \tag{18.8}$$

The feasible set M determined by the constraints is represented in **Fig. 18.4**. Introduction of slack variables x_4, x_5, x_6, x_7 leads to:

$$\textbf{CT}: \quad \begin{aligned} x_1 + x_2 + x_3 - x_4 &= 1, \\ x_2 + x_5 &= 2, \\ -x_1 + 2x_3 + x_6 &= 2, \\ 2x_1 - 3x_2 + 2x_3 + x_7 &= 2. \end{aligned}$$

The extreme point $P_2 = (0,1,0)$ of the polyhedron corresponds to the point $\underline{\mathbf{x}} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (0,1,0,0,1,2,5)$ of the extended system. The columns 2, 5, 6 and 7 of $\mathbf{A}$ form the corresponding basis. The degenerated extreme point P_1 corresponds to $(1,0,0,0,2,3,0)$. A basis of this extreme point contains the columns $1, 5, 6$ and one of the columns $2, 4$ or 7.

Remark: Here, the first inequality was a "$\geq$" inequality and x_4 was not added but subtracted. Frequently these types of additional variables both with a negative sign and a corresponding $b_i > 0$ are called *surplus variables*, rather than slack variables. As will be seen in 18.1.3.3, p. 916, the occurrence of surplus variables requires additional effort in the solution procedure.

4. Extreme Point with a Maximal Value of the Objective Function

Theorem: If M is not empty, and the objective function $f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}^{\mathrm{T}}\underline{\mathbf{x}}$ is bounded from above on M, then there is at least one extreme point of M where it has its maximum.

A linear programming problem can be solved by determining at least one of the extreme points with maximum value of the objective function. Usually, the number of extreme points of M is very large in practical problems, so a method is needed by which the solution can be found in a reasonable time. Such a method is the *simplex method* which is also called the *simplex algorithm* or *simplex procedure*.

18.1.2.2 Normal Form of the Linear Programming Problem

1. Normal Form and Basic Solution

The linear programming problem (18.4a,b) can always be transformed to the following form with a suitable renumbering of the variables:

$$\textbf{OF:} \quad f(\underline{\mathbf{x}}) = c_1x_1 + \cdots + c_{n-m}x_{n-m} + c_0 = \max! \tag{18.9a}$$

$$\textbf{CT:} \quad \left.\begin{array}{l} a_{1,1}x_1 + \cdots + a_{1,n-m}x_{n-m} + x_{n-m+1} = b_1\,, \\ \quad\vdots \qquad\qquad\qquad\quad \ddots \qquad\quad \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n-m}x_{n-m} \qquad + x_n = b_m\,, \\ x_1, \ldots, x_{n-m}, x_{n-m+1}, \ldots, x_n \geq 0. \end{array}\right\} \tag{18.9b}$$

The last m columns of the coefficient matrix are obviously independent, and they form a basis. The *basic solution* $(x_1, x_2, \ldots, x_{n-m}, x_{n-m+1}, \ldots, x_n) = (0, \ldots, 0, b_1, \ldots, b_m)$ can be determined directly from the system of equations, but if $\underline{\mathbf{b}} \geq \underline{\mathbf{0}}$ does not hold, it is not a feasible solution.

If $\underline{\mathbf{b}} \geq \underline{\mathbf{0}}$, then (18.9a,b) is called a *normal form* or *canonical form of the linear programming problem*. In this case, the basic solution is a feasible solution, as well, i.e., $\underline{\mathbf{x}} \geq 0$, and it is an extreme point of M. The variables $x_1, \ldots, x_{n-m}$ are called *non-basic variables* and $x_{n-m+1}, \ldots, x_n$ are called *basic variables*. The objective function has the value c_0 at this extreme point, since the non-basic variables are equal to zero.

2. Determination of the Normal Form

If an extreme point of M is known, then a normal form of the linear programming problem (18.5a,b) can be obtained in the following way. A basis is chosen from the columns of $\mathbf{A}$ corresponding to the extreme point. Usually, these columns are determined by the positive components of the extreme point. Suppose the basic variables are collected into the vector $\underline{\mathbf{x}}_B$ and the non-basic variables are in $\underline{\mathbf{x}}_N$. The columns associated to the basis form the basis matrix $\mathbf{A}_B$, the other columns form the matrix $\mathbf{A}_N$. Then,

$$\mathbf{A}\underline{\mathbf{x}} = \mathbf{A}_N\underline{\mathbf{x}}_N + \mathbf{A}_B\underline{\mathbf{x}}_B = \underline{\mathbf{b}}. \tag{18.10}$$

The matrix $\mathbf{A}_B$ is non-singular and it has an inverse $\mathbf{A}_B^{-1}$, the so-called *basis inverse*. Multiplying (18.10) by $\mathbf{A}_B^{-1}$ and changing the objective function according to the non-basic variables results in the canonical form of the linear programming problem:

$$\textbf{OF}: \quad f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}_N^{\mathrm{T}}\underline{\mathbf{x}}_N + c_0, \tag{18.11a}$$

$$\textbf{CT}: \quad \mathbf{A}_B^{-1}\mathbf{A}_N\underline{\mathbf{x}}_N + \underline{\mathbf{x}}_B = \mathbf{A}_B^{-1}\underline{\mathbf{b}} \quad \text{with} \quad \underline{\mathbf{x}}_N \geq \underline{\mathbf{0}},\ \ \underline{\mathbf{x}}_B \geq \underline{\mathbf{0}}. \tag{18.11b}$$

Remark: If the original system (18.1b) has only constraints of type "$\leq$" and simultaneously $b \geq \underline{\mathbf{0}}$, then the extended system (18.4b) contains no surplus variables (see 18.1.2.1, p. 911). In this case a normal form is immediately known. Selecting all slack variables as basic variables $\underline{\mathbf{x}}_B$ the result is $\mathbf{A}_B = \mathbf{I}$ and $\underline{\mathbf{x}}_B = \underline{\mathbf{b}}$ and $\underline{\mathbf{x}}_N = \underline{\mathbf{0}}$ is a feasible extreme point.

■ In the above example $\underline{\mathbf{x}} = (0, 1, 0, 0, 1, 2, 5)$ is an extreme point. Consequently:

$$\mathbf{A}_B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \\ x_2 & x_5 & x_6 & x_7 \end{pmatrix}, \qquad \mathbf{A}_B^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{A}_N = \begin{pmatrix} 1 & 1 & -1 \\ 0 & 0 & 0 \\ -1 & 2 & 0 \\ 2 & 2 & 0 \\ x_1 & x_3 & x_4 \end{pmatrix}, \tag{18.12a}$$

$$\mathbf{A}_B^{-1}\mathbf{A}_N = \begin{pmatrix} 1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & 2 & 0 \\ 5 & 5 & -3 \\ x_1 & x_3 & x_4 \end{pmatrix}, \qquad \mathbf{A}_B^{-1}\underline{\mathbf{b}} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 5 \end{pmatrix}. \tag{18.12b}$$

$$\left.\begin{array}{rrrrrrr} x_1 & + x_2 & + x_3 & - x_4 & & & = 1, \\ -x_1 & & - x_3 & + x_4 & + x_5 & & = 1, \\ -x_1 & & + 2x_3 & & & + x_6 & = 2, \\ 5x_1 & & + 5x_3 & - 3x_4 & & + x_7 & = 5. \end{array}\right\} \tag{18.13}$$

From $f(\underline{\mathbf{x}}) = 2x_1 + 3x_2 + 4x_3$ the transformed objective function

$$f(\underline{\mathbf{x}}) = -x_1 + x_3 + 3x_4 + 3 \tag{18.14}$$

is obtained, if the triple of the first constraint is subtracted.

18.1.3 Simplex Method

18.1.3.1 Simplex Tableau

The *simplex method* is used to produce a sequence of extreme points of the feasible set with increasing values of the objective function. The transition to the new extreme point is performed starting from the normal form corresponding to the given extreme point, and arriving at the normal form corresponding to the new extreme point. In order to get a clear arrangement, and easier numerical performance, the normal form (18.9a,b) is represented in the simplex tableau **(Scheme 18.2a, 18.2b)**:

Scheme 18.2a

	x_1	$\cdots$	x_{n-m}	
x_{n-m+1}	$a_{1,1}$	$\cdots$	$a_{1,n-m}$	b_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_n	$a_{m,1}$	$\cdots$	$a_{m,n-m}$	b_m
	c_1	$\cdots$	c_{n-m}	$-c_0$

or briefly

Scheme 18.2b

	$\underline{\mathbf{x}}_N$	
$\underline{\mathbf{x}}_B$	$\mathbf{A}_N$	$\underline{\mathbf{b}}$
	$\underline{\mathbf{c}}$	$-c_0$

The k-th row of the tableau corresponds to the constraint

$$x_{n-m+k} + a_{k,1}x_1 + \cdots + a_{k,n-m}x_{n-m} = b_k. \tag{18.15a}$$

The objective function is

$$c_1x_1 + \cdots + c_{n-m}x_{n-m} = f(\underline{\mathbf{x}}) - c_0. \tag{18.15b}$$

From this simplex tableau, the extreme point $(\underline{\mathbf{x}}_N, \underline{\mathbf{x}}_B) = (\underline{\mathbf{0}}, \underline{\mathbf{b}})$ can be found. The value of the objective function at this point is $f(\underline{\mathbf{x}}) = c_0$. To put down $-c_0$ into the right below vertex of the tableau is advantageous for carrying out the simplex method. In every tableau always exactly one of the following three cases can be found:

a) $c_j \le 0$, $j = 1, \ldots, n-m$: The tableau is optimal. The point $(\underline{\mathbf{x}}_N, \underline{\mathbf{x}}_B) = (\underline{\mathbf{0}}, \underline{\mathbf{b}})$ is a maximal point. If all the c_j are positive, then this vertex is the only maximal point.

b) There exists at least one j such that $c_j > 0$ and $a_{ij} \le 0$, $i = 1, \ldots, m$: The linear programming problem has no solution, since the objective function is not bounded on the feasible set; for increasing

values of x_j it increases without a bound.

c) For every j with $c_j > 0$ there exists at least one i with $a_{ij} > 0$: It is possible to move from the extreme point $\underline{\mathbf{x}}$ to a neighboring extreme point $\tilde{\underline{\mathbf{x}}}$ with $f(\tilde{\underline{\mathbf{x}}}) \geq f(\underline{\mathbf{x}})$. In the case of a non-degenerate extreme point $\underline{\mathbf{x}}$, the "$>$" sign always holds.

18.1.3.2 Transition to the New Simplex Tableau

1. Non-Degenerate Case

If a tableau is not in final form (case **c)**), then a new tableau **(Scheme 18.3)** is determined. A basic variable x_p and a non-basic variable x_q are interchanged by the following calculations:

$$\textbf{a)}\ \tilde{a}_{pq} = \frac{1}{a_{pq}}. \tag{18.16a}$$

$$\textbf{b)}\ \tilde{a}_{pj} = a_{pj} \cdot \tilde{a}_{pq}, \qquad j \neq q, \qquad \tilde{b}_p = b_p \cdot \tilde{a}_{pq}. \tag{18.16b}$$

$$\textbf{c)}\ \tilde{a}_{iq} = -a_{iq} \cdot \tilde{a}_{pq}, \qquad i \neq p, \qquad \tilde{c}_q = -c_q \cdot \tilde{a}_{pq}. \tag{18.16c}$$

$$\textbf{d)}\ \tilde{a}_{ij} = a_{ij} + a_{pj} \cdot \tilde{a}_{iq}, \quad i \neq p,\ j \neq q,$$
$$\tilde{b}_i = b_i + b_p \cdot \tilde{a}_{iq}, \qquad i \neq p, \qquad \tilde{c}_j = c_j + a_{pj} \cdot \tilde{c}_q, \quad j \neq q, -\tilde{c}_0 = -c_0 + b_p \cdot \tilde{c}_q. \tag{18.16d}$$

The element a_{pq} is called the *pivot element* , the p-th row is the *pivot row*, and the q-th column is the *pivot column* . For the choice of a pivot element the following two requirements are to be considered:
a) $\tilde{c}_0 \geq c_0$ should hold;

b) the new tableau must also correspond to a feasible solution, i.e., $\tilde{\underline{\mathbf{b}}} \geq \underline{\mathbf{0}}$ must hold.

Then, $(\tilde{\underline{\mathbf{x}}}_N, \tilde{\underline{\mathbf{x}}}_B) = (\underline{\mathbf{0}}, \tilde{\underline{\mathbf{b}}})$ is a new extreme point at which the value of the objective function $f(\tilde{\underline{\mathbf{x}}}) = \tilde{c}_0$ is not smaller than it was previously. These conditions are satisfied if the pivot element is chosen in the following way:

a) To increase the value of the objective function, a column with $c_q > 0$ can be chosen for a pivot column;

b) to get a feasible solution, the pivot row must be chosen as

$$\frac{b_p}{a_{pq}} = \min_{\substack{1 \leq i \leq m \\ a_{iq} > 0}} \left\{ \frac{b_i}{a_{iq}} \right\}. \tag{18.17}$$

If the extreme points of the feasible set are not degenerate, then the simplex method terminates in a finite number of steps (case **a)** or case **b)**).

■ The normal form in 18.1.2, p. 911ff can be written in a simplex tableau **(Scheme 18.4a)**. This tableau is not optimal, since the objective function has a positive coefficient in the third column. The third column is assigned as the pivot column (the second column could also be taken under consideration). The quotients b_i/a_{iq} are calculated with every positive element of the pivot column (there is only one of them). The quotients are denoted behind the last column. The smallest quotient determines the pivot row.

Scheme 18.3

	$\tilde{\underline{\mathbf{x}}}_N$	
$\tilde{\underline{\mathbf{x}}}_B$	$\tilde{\mathbf{A}}_N$	$\tilde{\underline{\mathbf{b}}}$
	$\tilde{\underline{\mathbf{c}}}$	$-\tilde{c}_0$

Scheme 18.4a

	x_1	x_3	x_4	
x_2	1	1	$-\underline{1}$	1
x_5	$-\underline{1}$	-1	$\underline{1}$	$\underline{1}$
x_6	-1	2	$\underline{0}$	2
x_7	5	5	$-\underline{3}$	5
	-1	1	$\underline{3}$	-3

1 : 1

Scheme 18.4b

	x_1	x_3	x_5	
x_2	0	$\underline{0}$	1	2
x_4	-1	$-\underline{1}$	1	1
x_6	$-\underline{1}$	$\underline{2}$	$\underline{0}$	$\underline{2}$
x_7	2	$\underline{2}$	3	8
	2	$\underline{4}$	-3	-6

2 : 2

8 : 2

If it is not unique, then the extreme point corresponding to the new tableau is degenerate. After per-

forming the steps of (18.16a)–(18.16d) the tableau in **Scheme 18.4b** is obtained. This tableau determines the extreme point $(0, 2, 0, 1, 0, 2, 8)$, which corresponds to the point P_7 in **Fig. 18.4**. Since this new tableau is still not optimal, x_6 and x_3 are interchanged **(Scheme 18.4c)**. The extreme point of the third tableau corresponds to the point P_6 in **Fig. 18.4**. After an additional change an optimal tableau is obtained **(Scheme 18.4d)** with the maximal point $\underline{\mathbf{x}}^* = (2, 2, 2, 5, 0, 0, 0)$, which corresponds to the point P_5, and the objective function has a maximal value here: $f(\underline{\mathbf{x}}^*) = 18$.

Scheme 18.4c

	x_1	x_6	x_5	
x_2	$\underline{0}$	0	1	2
x_4	$-\underline{\frac{3}{2}}$	$\frac{1}{2}$	1	2
x_3	$-\underline{\frac{1}{2}}$	$\frac{1}{2}$	0	1
x_7	$\underline{3}$	$-\underline{1}$	$\underline{3}$	$\underline{6}$
	$\underline{4}$	-2	-3	-10

$6 : 3$

Scheme 18.4d

	x_7	x_6	x_5	
x_2	0	0	1	2
x_4	$\frac{1}{2}$	0	$\frac{5}{2}$	5
x_3	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$	2
x_1	$\frac{1}{3}$	$-\frac{1}{3}$	1	2
	$-\frac{4}{3}$	$-\frac{2}{3}$	-7	-18

Scheme 18.5

	x_1	$\cdots$	x_n	
y_1	$a_{1,1}$	$\cdots$	$a_{1,n}$	b_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_m	$a_{m,1}$	$\cdots$	$a_{m,n}$	b_m
OF	c_1	$\cdots$	c_n	0
OF*	$\sum_{j=1}^{m} a_{j,1}$	$\cdots$	$\sum_{j=1}^{m} a_{j,n}$	$\sum_{j=1}^{m} b_j = -g(\underline{\mathbf{0}}, \underline{\mathbf{b}})$

2. Degenerate Case

If the next pivot element cannot be chosen uniquely in a simplex tableau, then the new tableau represents a degenerate extreme point. A degenerate extreme point can be interpreted geometrically as the coincident vertices of the convex polyhedron of the feasible solutions. There are several bases for such a vertex. In this case, it can therefore happen that some steps are performed without reaching a new extreme point. It is also possible that one gets a tableau that has occurred already before, so an infinite cycle may occur.

In the case of a degenerate extreme point, one possibility is to perturb the constants b_i by adding ε^i (with a suitable $\varepsilon^i > 0$) such that the resulting extreme points are no longer degenerate. The solution can be got from the solution of the perturbed problem, if $\varepsilon = 0$ is substituted.

If the pivot column is chosen "randomly" in the non-uniquely determined case, then the occurrence of an infinite cycle is unlikely in practical cases.

18.1.3.3 Determination of an Initial Simplex Tableau

1. Secondary Program, Artificial Variables

If there are equalities among the original constraints (18.1b) or inequalities with negative b_i, then it is not easy to find a feasible solution to start the simplex method. In this case, one starts with a secondary program to produce a feasible solution, which can be a starting point for a simplex procedure for the original problem. The system $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ is modified by multiplying some of the equations with (-1) in order to satisfy the condition $\underline{\mathbf{b}} \geq \underline{\mathbf{0}}$. Now, an *artificial variable* $y_k \geq 0$ $(k = 1, 2, \ldots, m)$ is added to every left-hand side of $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ with $\underline{\mathbf{b}} \geq \underline{\mathbf{0}}$, and the secondary program is considered:

$$\mathbf{OF}^*\text{: } g(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = -y_1 - \cdots - y_m = \max! \tag{18.18a}$$

$$\mathbf{CT}^*\text{: } \left.\begin{array}{l} a_{1,1}x_1 + \cdots + a_{1,n}x_n + y_1 = b_1, \\ \quad\vdots \qquad\qquad\quad \vdots \qquad\quad \ddots \quad\; \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n}x_n + y_m = b_m, \\ x_1, \ldots, x_n \geq 0;\;\; y_1, \ldots, y_m \geq 0. \end{array}\right\} \tag{18.18b}$$

For this problem, the variables $y_1, \ldots, y_m$ are basic variables, and one can start the first simplex tableau **(Scheme 18.5)**. The last row of the tableau contains the sums of the coefficients of the non-basic variables , and these sums are the coefficients of the new secondary objective function **OF***. Obviously, $g(\underline{\mathbf{x}}, \underline{\mathbf{y}}) \leq 0$ always. If $g(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*) = 0$ for a maximal point $(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*)$ of the secondary problem, then obviously $\underline{\mathbf{y}}^* = 0$, and consequently $\underline{\mathbf{x}}^*$ is a solution of $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$. If $g(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*) < 0$, then $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ does not have any solution.

2. Solution of the Secondary Program

Our goal is to eliminate the artificial variables from the basis. A scheme is prepared not only for the secondary program separately. The original tableau is completed by columns of the artificial variables and the row of the secondary objective function. The secondary objective function now contains the sums of the corresponding coefficients from the rows corresponding to the equalities, as shown below. If an artificial variable becomes a non-basic variable, its column can be omitted, since it will be never chosen again as a basis variable. If a maximal point $(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*)$ is determined, then two cases are distinguished:

1. $g(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*) < 0$: The system $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ has no solution, the linear programming problem does not have any feasible solution.

2. $g(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*) = 0$: If there are no artificial variables among the basic variables, this tableau is an initial tableau for the original problem. Otherwise all artificial variables among the basic variables are removed by additional steps of the simplex method.

By introducing the artificial variables, the size of the problem can be increased considerably. It is not necessary to introduce artificial variables for every equation. If the system of constraints before introducing the slack and surplus variables (see Remark in 18.1.2, **3.**, p. 913) has the form $\mathbf{A}_1\underline{\mathbf{x}} \geq \underline{\mathbf{b}}_1$, $\mathbf{A}_2\underline{\mathbf{x}} = \underline{\mathbf{b}}_2$, $\mathbf{A}_3\underline{\mathbf{x}} \leq \underline{\mathbf{b}}_3$ with $\underline{\mathbf{b}}_1, \underline{\mathbf{b}}_2, \underline{\mathbf{b}}_3 \geq 0$, then artificial variables must be introduced only for the first two systems. For the third system the slack variables can be chosen as basic variables.

■ In the example of 18.1.2, p. 912, only the first equation requires an artificial variable:

$$\begin{aligned}
&\mathbf{OF}^*\text{:} \quad g(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = && && -y_1 && && = \max!\\
&\mathbf{CT}^*\text{:} && x_1 + x_2 + x_3 - x_4 + y_1 && && && = 1,\\
& && x_2 && + x_5 && && = 2,\\
& && -x_1 + 2x_3 && && + x_6 && = 2,\\
& && 2x_1 - 3x_2 + 2x_3 && && && + x_7 = 2.
\end{aligned}$$

The tableau **(Scheme 18.6b)** is optimal with $g(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}^*) = 0$. After omitting the second column the first tableau of the original problem is obtained.

Scheme 18.6a

	x_1	x_2	x_3	x_4		
y_1	1	1	1	−1	1	1 : 1
x_5	0	1	0	0	2	2 : 1
x_6	−1	0	2	0	2	
x_7	2	−3	2	0	2	
OF	2	3	4	0	0	
OF*	1	1	1	−1	1	

Scheme 18.6b

	x_1	y_1	x_3	x_4	1
x_2	1	1	1	−1	1
x_5	−1	−1	−1	1	1
x_6	−1	0	2	0	2
x_7	5	3	5	−3	5
OF	−1	−3	1	3	−3
OF*	0	−1	0	0	0

18.1.3.4 Revised Simplex Method

1. Revised Simplex Tableau

Suppose the linear programming problem is given in normal form:

$$\mathbf{OF:} \quad f(\underline{\mathbf{x}}) = c_1x_1 + \cdots + c_{n-m}x_{n-m} + c_0 = \max! \tag{18.19a}$$

$$\textbf{CT:}\quad \left.\begin{array}{llll} \alpha_{1,1}x_1 + \cdots + \alpha_{1,n-m}x_{n-m} + x_{n-m+1} & & = \beta_1, \\ \vdots \qquad\qquad \vdots & \ddots & \vdots \\ \alpha_{m,1}x_1 + \cdots + \alpha_{m,n-m}x_{n-m} & + x_n & = \beta_m, \\ x_1 \geq 0, \ldots, x_n \geq 0. & & \end{array}\right\} \tag{18.19b}$$

Obviously, the coefficient vectors $\underline{\alpha}_{n-m+i}$ $(i = 1, \ldots, n)$ are the i-th unit vectors.
In order to change into another normal form and therefore to reach another extreme point, it is sufficient to multiply the system of equations (18.19b) by the corresponding basis inverse. (Recall the fact that if $\mathbf{A}_B$ denotes a new basis, then the coordinates of a vector $\underline{\mathbf{x}}$ can be expressed in this new basis as $\mathbf{A}_B^{-1}\underline{\mathbf{x}}$. If the inverse of the new basis is known, then any column as well as the objective function from the very first tableau can be got by simple multiplication.) The simplex method can be modified so that only the basis inverse is determined in every step instead of a new tableau. From every tableau only those elements are calculated which are required to find the new pivot element. If the number of variables is considerably larger than the number of constraints ($n > 3m$), then the revised simplex method requires considerably less computing cost and therefore has better accuracy.
The general form of a revised simplex tableau is shown in **Scheme 18.7**.

Scheme 18.7

	$x_1 \cdots x_{n-m}$	$x_{n-m+1} \cdots x_n$		x_q
x_1^B		$a_{1,n-m+1} \cdots a_{1,n}$	b_1	r_1
$\vdots$		$\vdots \qquad\quad \vdots$	$\vdots$	$\vdots$
x_m^B		$a_{m,n-m+1} \cdots a_{m,n}$	b_m	r_m
	$c_1 \cdots c_{n-m}$	$c_{n-m+1} \cdots c_n$	$-c_0$	c_q

The quantities of the scheme have the following meaning:

$x_1^B, \ldots, x_m^B$: Actual basic variables (in the first step the same as $x_{n-m+1} \cdots x_n$).
$c_1, \ldots, c_n$: Coefficients of the objective function (the coefficients associated to the basic variables are zeros).
$b_1, \ldots, b_m$: Right-hand side of the actual normal form.
c_0 : Value of the objective function at the extreme point $(x_1^B, \ldots, x_m^B) = (b_1, \ldots, b_m)$.
$\mathbf{A}^* = \begin{pmatrix} a_{1,n-m+1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,n-m+1} & \cdots & a_{m,n} \end{pmatrix}$: Actual basis inverse, where the columns of $\mathbf{A}^*$ are the columns of $x_{n-m+1}, \ldots, x_n$ corresponding to the actual normal form;
$\underline{\mathbf{r}} = (r_1, \ldots, r_m)^{\mathrm{T}}$: Actual pivot column.

2. Revised Simplex Step

a) The tableau is not optimal when at least one of the coefficients c_j $(j = 1, 2, \ldots, n)$ is positiv. A pivot column q is chosen for a $c_q > 0$.
b) One calculates the pivot column $\underline{\mathbf{r}}$ by multiplying the q-th column of the original coefficient matrix (18.19b) by $\mathbf{A}^*$ and introduces the new vector as the last vector of the tableau.
The pivot row k is determined in the same way as in the simplex algorithm (18.17).
c) The new tableau is calculated by the pivoting step (18.16a–d), where a_{iq} is formally replaced by r_i and the indices are restricted for $n - m + 1 \leq j \leq n$. The column $\tilde{\underline{\mathbf{r}}}$ is omitted. x_q becomes a basic variable. For $j = 1, \ldots, n - m$, the results are $\tilde{c}_j = c_j + \underline{\alpha}_j^{\mathrm{T}}\tilde{\underline{\mathbf{c}}}$, where $\tilde{\underline{\mathbf{c}}} = (\tilde{c}_{n-m+1}, \ldots, \tilde{c}_n)^{\mathrm{T}}$, and $\underline{\alpha}_j$ is the j-th column of the coefficient matrix of (18.19b).

■ Consider the normal form of the example in 18.1.2, p. 912. One wants to bring x_4 into the basis. The corresponding pivot column $\underline{\mathbf{r}} = \underline{\alpha}_4$ is placed into the last column of the tableau **(Scheme 18.8a)** (initially $\mathbf{A}^*$ is the unit matrix).

For $j = 1, 3, 4$ one gets $\tilde{c}_j = c_j - 3\alpha_{2j}$: $(c_1, c_3, c_4) = (2, 4, 0)$.
The determined extreme point $\underline{\mathbf{x}} = (0, 2, 0, 1, 0, 2, 8)$ corresponds to the point P_7 in **Fig. 18.4**, p. 912. The next pivot column can be chosen for $j = 3 = q$.

Scheme 18.8a

	x_1	x_3	x_4	x_2	x_5	x_6	x_7		x_4	
x_2				1	0	0	0	1	$\underline{-1}$	
x_5				$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{0}$	$\underline{1}$	$\underline{1}$	1 : 1
x_6				0	0	1	0	2	$\underline{0}$	
x_7				0	0	0	1	5	$\underline{-3}$	
	-1	1	$\underline{3}$	0	0	0	0	-3	$\underline{3}$	

Scheme 18.8b

	x_1	x_3	x_4	x_2	x_5	x_6	x_7		x_3	
x_2				1	1	0	0	2	$\underline{0}$	
x_4				0	1	0	0	1	$\underline{-1}$	
x_6				$\underline{0}$	$\underline{0}$	$\underline{1}$	$\underline{0}$	$\underline{2}$	$\underline{\underline{2}}$	2 : 2
x_7				0	3	0	1	8	$\underline{2}$	8 : 2
	2	$\underline{4}$	-3	0	-3	0	0	-6	$\underline{4}$	

The vector $\underline{\mathbf{r}}$ is determined by

$$\underline{\mathbf{r}} = (r_1, \ldots, r_m) = \mathbf{A}^* \underline{\alpha}_3 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \\ 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 2 \end{pmatrix}$$

and it is placed into the very last column of the second tableau **(Scheme 18.8b)**. One proceedes as above analogously to the method shown in 18.1.3.2, p. 915. If one wants to return to the original method, then the matrix of the original columns of the non-basic variables must be multiplied by $\mathbf{A}^*$ and only these columns will be kept.

18.1.3.5 Duality in Linear Programming

1. Correspondence

To any linear programming problem (primal problem) an other unique linear programming problem can be assigned (dual problem):

Primal problem

OF: $f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}_1^{\mathrm{T}} \underline{\mathbf{x}}_1 + \underline{\mathbf{c}}_2^{\mathrm{T}} \underline{\mathbf{x}}_2 = \max!$ (18.20a)

CT: $\mathbf{A}_{1,1}\underline{\mathbf{x}}_1 + \mathbf{A}_{1,2}\underline{\mathbf{x}}_2 \le \underline{\mathbf{b}}_1$,
$\mathbf{A}_{2,1}\underline{\mathbf{x}}_1 + \mathbf{A}_{2,2}\underline{\mathbf{x}}_2 = \underline{\mathbf{b}}_2$,
$\underline{\mathbf{x}}_1 \ge 0$, $\underline{\mathbf{x}}_2$ free. (18.20b)

Dual problem

OF*: $g(\underline{\mathbf{u}}) = \underline{\mathbf{b}}_1^{\mathrm{T}} \underline{\mathbf{u}}_1 + \underline{\mathbf{b}}_2^{\mathrm{T}} \underline{\mathbf{u}}_2 = \min!$ (18.21a)

CT*: $\mathbf{A}_{1,1}^{\mathrm{T}}\underline{\mathbf{u}}_1 + \mathbf{A}_{2,1}^{\mathrm{T}}\underline{\mathbf{u}}_2 \ge \underline{\mathbf{c}}_1$,
$\mathbf{A}_{1,2}^{\mathrm{T}}\underline{\mathbf{u}}_1 + \mathbf{A}_{2,2}^{\mathrm{T}}\underline{\mathbf{u}}_2 = \underline{\mathbf{c}}_2$,
$\underline{\mathbf{u}}_1 \ge 0$, $\underline{\mathbf{u}}_2$ free. (18.21b)

The coefficients of the objective function of one of the problems form the right-hand side vector of the constraints of the other problem. Every free variable corresponds to an equation, and every variable with restricted sign corresponds to an inequality of the other problem.

2. Duality Theorems

a) If both problems have feasible solutions, i.e., $M \neq \emptyset$, $M^* \neq \emptyset$ (where M and M^* denote the feasible sets of the primal and dual problems respectively), then

$$f(\underline{\mathbf{x}}) \le g(\underline{\mathbf{u}}) \quad \text{for all } \underline{\mathbf{x}} \in M,\ \underline{\mathbf{u}} \in M^*, \tag{18.22a}$$

and both problems have optimal solutions.

b) The points $\underline{\mathbf{x}} \in M$ and $\underline{\mathbf{u}} \in M^*$ are optimal solutions for the corresponding problem, if and only if

$$f(\underline{\mathbf{x}}) = g(\underline{\mathbf{u}}). \tag{18.22b}$$

c) If $f(\underline{\mathbf{x}})$ has no upper bound on M or $g(\underline{\mathbf{u}})$ has no lower bound on M^*, then $M^* = \emptyset$ or $M = \emptyset$, i.e., the dual problem has no feasible solution.

d) The points $\underline{\mathbf{x}} \in M$ and $\underline{\mathbf{u}} \in M^*$ are optimal points of the corresponding problems if and only if:

$$\underline{\mathbf{u}}_1^{\mathrm{T}}(\mathbf{A}_{1,1}\underline{\mathbf{x}}_1 + \mathbf{A}_{1,2}\underline{\mathbf{x}}_2 - \underline{\mathbf{b}}_1) = 0 \quad \text{and} \quad \underline{\mathbf{x}}_1^{\mathrm{T}}(\mathbf{A}_{1,1}^{\mathrm{T}}\underline{\mathbf{u}}_1 + \mathbf{A}_{2,1}^{\mathrm{T}}\underline{\mathbf{u}}_2 - \underline{\mathbf{c}}_1) = 0. \tag{18.22c}$$

Using these last equations, a solution $\underline{\mathbf{x}}$ of the primal problem can be found from a non-degenerate optimal solution $\underline{\mathbf{u}}$ of the dual problem by solving the following linear system of equations:

$$\mathbf{A}_{2,1}\underline{\mathbf{x}}_1 + \mathbf{A}_{2,2}\underline{\mathbf{x}}_2 - \underline{\mathbf{b}}_2 = \underline{\mathbf{0}}, \tag{18.23a}$$

$$(\mathbf{A}_{1,1}\underline{\mathbf{x}}_1 + \mathbf{A}_{1,2}\underline{\mathbf{x}}_2 - \underline{\mathbf{b}}_1)_i = \underline{\mathbf{0}} \quad \text{for } u_i > 0, \tag{18.23b}$$

$$x_i = 0 \quad \text{for } (\mathbf{A}_{1,1}^{\mathrm{T}}\underline{\mathbf{u}}_1 + \mathbf{A}_{2,1}^{\mathrm{T}}\underline{\mathbf{u}}_2 - \underline{\mathbf{c}}_1)_i \neq 0. \tag{18.23c}$$

The dual problem also can be solved by the simplex method.

3. Application of the Dual Problem

Working with the dual problem may have some advantages in the following cases:

a) If it is simple to find a normal form for the dual problem, one switches from the primal problem to the dual.

b) If the primal problem has a large number of constraints compared to the number of variables, then the revised simplex method can be used for the dual problem.

■ Consider the original problem of the example of 18.1.2, p. 912.

Primal problem

$$\begin{aligned}
\textbf{OF:}\quad & f(\underline{\mathbf{x}}) = 2x_1 + 3x_2 + 4x_3 = \max! \\
\textbf{CT:}\quad & -x_1 - x_2 - x_3 \leq -1, \\
& x_2 \leq 2, \\
& -x_1 + 2x_3 \leq 2, \\
& 2x_1 - 3x_2 + 2x_3 \leq 2, \\
& x_1, x_2, x_3 \geq 0.
\end{aligned}$$

Dual problem

$$\begin{aligned}
\textbf{OF}^*\textbf{:}\quad & g(\underline{\mathbf{u}}) = -u_1 + 2u_2 + 2u_3 + 2u_4 = \min! \\
\textbf{CT}^*\textbf{:}\quad & -u_1 - u_3 + 2u_4 \geq 2, \\
& -u_1 + u_2 - 3u_4 \geq 3, \\
& -u_1 + 2u_3 + 2u_4 \geq 4, \\
& u_1, u_2, u_3, u_4 \geq 0.
\end{aligned}$$

If the dual problem is solved by the simplex method after introducing the slack variables, then the optimal solution $\underline{\mathbf{u}}^* = (u_1, u_2, u_3, u_4) = (0, 7, 2/3, 4/3)$ with $g(\underline{\mathbf{u}}) = 18$ is got. A solution $\underline{\mathbf{x}}^*$ of the primal problem can be got by solving the system $(\mathbf{A}\underline{\mathbf{x}} - \underline{\mathbf{b}})_i = 0$ for $u_i > 0$, i.e., $x_2 = 2$, $-x_1 + 2x_3 = 2$, $2x_1 - 3x_2 + 2x_3 = 2$, therefore: $\underline{\mathbf{x}}^* = (2, 2, 2)$ with $f(\underline{\mathbf{x}}) = 18$.

18.1.4 Special Linear Programming Problems

18.1.4.1 Transportation Problem

1. Modeling

A certain product, produced by m producers $E_1, E_2, \ldots, E_m$ in quantities $a_1, a_2, \ldots, a_m$, is to be transported to n consumers $V_1, V_2, \ldots, V_n$ with demands $b_1, b_2, \ldots, b_n$. Transportation cost of a unit product of producer E_i to consumer V_j is c_{ij}. The amount of the product transported from E_i to V_j is x_{ij} units. An optimal transportation plan is to be determined with minimum total transportation cost. The system is supposed to be balanced, i.e., supply equals demand:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{18.24}$$

The matrix of costs $\mathbf{C}$ and the distribution matrix $\mathbf{X}$ are constructed:

$$\mathbf{C} = \begin{array}{c} \\ V: \end{array} \begin{array}{c} \begin{array}{ccc} V_1 & \cdots & V_n \end{array} \\ \begin{pmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & & \vdots \\ c_{m,1} & \cdots & c_{m,n} \end{pmatrix} \end{array} \begin{array}{c} E: \\ E_1 \\ \vdots \\ E_m \end{array}, \tag{18.25a}$$

$$\mathbf{X} = \begin{array}{c} \\ \sum: \end{array} \begin{array}{c} \begin{array}{ccc} b_1 & \cdots & b_n \end{array} \\ \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{pmatrix} \end{array} \begin{array}{c} \sum: \\ a_1 \\ \vdots \\ a_m \end{array}. \tag{18.25b}$$

If condition (18.24) is not fulfilled, then two cases are distinguished:

a) If $\sum a_i > \sum b_j$, then a fictitious consumer V_{n+1} is introduced with demand $b_{n+1} = \sum a_i - \sum b_j$ and with transportation costs $c_{i,n+1} = 0$.

b) If $\sum a_i < \sum b_j$, then introduce a fictitious producer E_{m+1} is introduced with capacity $a_{m+1} =$

$\sum b_j - \sum a_i$ and with transportation costs $c_{m+1,j} = 0$.

In order to determine an optimal program, the following programming problem should be solved:

$$\mathbf{OF}: \quad f(\mathbf{X}) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \min! \tag{18.26a}$$

$$\mathbf{CT}: \quad \sum_{j=1}^{n} x_{ij} = a_i \;\; (i = 1, \ldots, m), \quad \sum_{i=1}^{m} x_{ij} = b_j \;\; (j = 1, \ldots, n), \;\; x_{ij} \geq 0). \tag{18.26b}$$

The minimum of the problem occurs at a vertex of the feasible set. There are $m + n - 1$ linearly independent constraints among the $m + n$ original constraints, so, in the non-degenerate case, the solution contains $m + n - 1$ positive components x_{ij}. To determine an optimal solution the following algorithm is used, which is called the transportation algorithm.

2. Determination of a Basic Feasible Solution

With the *Northwest corner rule* an initial basic feasible solution can be determined:

a) Choose $x_{11} = \min(a_1, b_1)$. (18.27a)

b) If $a_1 < b_1$, the first row of $\mathbf{X}$ is omitted. (18.27b)

If $a_1 > b_1$, the first column of $\mathbf{X}$ is omitted. (18.27c)

If $a_1 = b_1$, either the first row or the first remaining column of $\mathbf{X}$ is omitted. (18.27d)

If there are only one row but several columns, then one column is cancelled. The same applies for the rows.

c) a_1 is replaced by $a_1 - x_{11}$ and b_1 by $b_1 - x_{11}$ and the procedure in repeated in the left upper vertex of the reduced distribution matrix $\mathbf{X}$.

The variables obtained in step **a)** are the basic variables, all the others are non-basic variables with zero values.

■

$$\mathbf{C} = \begin{array}{c} \\ E: \\ \left(\begin{array}{cccc} 5 & 3 & 2 & 7 \\ 8 & 2 & 1 & 1 \\ 9 & 2 & 6 & 3 \end{array}\right) \begin{array}{c} E_1 \\ E_2 \\ E_3 \end{array} \\ V: \quad V_1 \; V_2 \; V_3 \; V_4 \end{array}, \qquad \mathbf{X} = \begin{array}{c} \\ \sum: \\ \left(\begin{array}{cccc} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{array}\right) \begin{array}{c} a_1 = 9 \\ a_2 = 10 \\ a_3 = 3 \end{array} \\ \sum: \quad b_1 = 4 \; b_2 = 6 \; b_3 = 5 \; b_4 = 7 \end{array}.$$

The determination of an initial extreme point with the Northwest corner rule gives

first step

$$\mathbf{X} = \begin{pmatrix} 4 & & \\ & & \\ & & \end{pmatrix} \begin{array}{c} \not{9} \;\; 5 \\ 10 \\ 3 \end{array}, \qquad \begin{array}{c} \not{4} \; 6 \; 5 \; 7 \\ 0 \end{array}$$

second step

$$\mathbf{X} = \begin{pmatrix} 4 & 5 & \text{—} \\ & & \\ & & \end{pmatrix} \begin{array}{c} \not{5} \;\; 0 \\ 10 \\ 3 \end{array}, \qquad \begin{array}{c} 0 \; \not{6} \; 5 \; 7 \\ \not{1} \end{array}$$

further steps

$$\mathbf{X} = \begin{pmatrix} 4 & 5 & \text{—} & \\ & & & \\ & & & \end{pmatrix} \begin{array}{c} 0 \\ 10 \;\; \not{9} \;\; \not{4} \; 0 \\ 3 \end{array}. \qquad \begin{array}{c} 0 \quad \not{5} \; \not{7} \\ \not{1} \; 0 \; 3 \\ 0 \end{array}$$

There are alternative methods to find an initial basic solution which also takes the transportation costs into consideration (see, e.g., the Vogel approximation method in [18.13]) and they usually result in a better initial solution.

3. Solution of the Transportation Problem with the Simplex Method

If the usual simplex tableau is prepared for this problem, then it results in a huge tableau $((m+n) \times (m \cdot n))$ with a large number of zeros: In each column, only two elements are equal to 1. So, a reduced tableau is constructed, and the following steps correspond to the simplex steps working only with the non-zero elements of the theoretical simplex tableau. The matrix of the cost data contains the coefficients of

the objective function. The basic variables are exchanged for non-basic variables iteratively, while the corresponding elements of the cost matrix are modified in each step. The procedure is explained by an example.

a) Determination of the modified cost matrix $\tilde{\mathbf{C}}$ from $\mathbf{C}$ by

$$\tilde{c}_{ij} = c_{ij} + p_i + q_j \quad (i = 1, \ldots, m, \;\; j = 1, \ldots, n), \tag{18.28a}$$

with the conditions

$$\tilde{c}_{ij} = 0 \;\text{ for }\; (i, j) \;\text{ if } x_{ij} \text{ is an actual basic variable.} \tag{18.28b}$$

The elements of $\mathbf{C}$ belonging to basic variables are marked and $p_1 = 0$ is substituted. The other quantities p_i and q_j, also called potentials or simplex multiplicators, are determined so that the sum of p_i, q_j and the marked costs c_{ij} should be 0:

■

$$\mathbf{C} = \begin{pmatrix} (5) & (3) & 2 & 7 \\ 8 & (2) & (1) & (1) \\ 9 & 2 & 6 & (3) \end{pmatrix} \begin{matrix} p_1 = 0 \\ p_2 = 1 \\ p_3 = -1 \end{matrix} \quad \Longrightarrow \quad \tilde{\mathbf{C}} = \begin{pmatrix} 0 & 0 & 0 & 5 \\ 4 & 0 & 0 & 0 \\ 3 & \boxed{-2} & 3 & 0 \end{pmatrix}. \tag{18.28c}$$

$$q_1 = -5 \;\; q_2 = -3 \;\; q_3 = -2 \;\; q_4 = -2$$

b) The value

$$\tilde{c}_{pq} = \min_{i,j}\{\tilde{c}_{ij}\} \tag{18.28d}$$

must be determined. If $\tilde{c}_{pq} \geq 0$, then the given distribution $\mathbf{X}$ is optimal; otherwise x_{pq} is chosen as a new basic variable. In our example: $\tilde{c}_{pq} = \tilde{c}_{32} = -2$.

c) In $\tilde{\mathbf{C}}$, $\tilde{c}_{pq}$ and the costs associated to the basic variables are marked. If $\tilde{\mathbf{C}}$ contains rows or columns with at most one marked element, then these rows or columns will be omitted. This procedure is repeated with the remaining matrix, until no further cancellation is possible.

$$\tilde{\mathbf{C}} = \begin{pmatrix} \cancel{(0)} & \cancel{(0)} & \cancel{0} & \cancel{5} \\ 4 & (0) & (0) & (0) \\ 3 & (-2) & 3 & (0) \end{pmatrix}. \tag{18.28e}$$

d) The elements x_{ij} associated to the remaining marked elements $\tilde{c}_{ij}$ form a cycle. The new basic variable $\tilde{x}_{pq}$ is to be set to a positive value δ. The other variables $\tilde{x}_{ij}$ associated to the marked elements $\tilde{c}_{ij}$ are determined by the constraints. In practice, δ is subtracted and added from or to every second element of the cycle. To keep the variables non-negative, the amount δ must be chosen as

$$\delta = x_{rs} = \min\{x_{ij} \colon \;\; \tilde{x}_{ij} = x_{ij} - \delta\}, \tag{18.28f}$$

where x_{rs} will be the non-basic variable. In the example $\delta = \min\{1, 3\} = 1$.

$$\tilde{\mathbf{X}} = \begin{matrix} \\ \\ \\ \\ \sum \end{matrix} \begin{matrix} \begin{matrix} \sum \end{matrix} \\ \begin{pmatrix} 4 & 5 & & \\ & \longleftarrow & & \\ 1-\delta & 5 & 4+\delta & \\ \downarrow & & \uparrow & \\ \delta & \longrightarrow & 3-\delta & \end{pmatrix} \\ \begin{matrix} 4 & 6 & 5 & 7 \end{matrix} \end{matrix} \begin{matrix} 9 \\ 10 \\ 3 \end{matrix} \quad \Longrightarrow \quad \tilde{\mathbf{X}} = \begin{pmatrix} 4 & 5 & \\ & 5 & 5 \\ 1 & 2 & \end{pmatrix}, \quad f(\underline{\mathbf{x}}) = 53. \tag{18.28g}$$

Then, this procedure is repeated with $\mathbf{X} = \tilde{\mathbf{X}}$.

$$\mathbf{C} = \begin{pmatrix} (5) & (3) & 2 & 7 \\ 8 & 2 & (1) & (1) \\ 9 & (2) & 6 & (3) \end{pmatrix} \begin{matrix} p_1 = 0 \\ p_2 = 3 \\ p_3 = 1 \end{matrix} \quad \Longrightarrow \quad \tilde{\mathbf{C}} = \begin{pmatrix} (0) & (0) & (\underline{-2}) & 3 \\ 6 & 2 & (0) & (0) \\ 5 & (0) & 3 & (0) \end{pmatrix}, \tag{18.28h}$$

$$q_1 = -5 \;\; q_2 = -3 \;\; q_3 = -4 \;\; q_4 = -4$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} 4 & 5-\delta \leftarrow & \delta \\ & \uparrow & \\ & \downarrow & 5-\delta \leftarrow 5+\delta \\ & & \uparrow \\ & 1+\delta \longrightarrow & 2-\delta \end{pmatrix} \qquad \begin{matrix} \delta = 2 \\ \Longrightarrow \end{matrix} \quad \tilde{\mathbf{X}} = \begin{pmatrix} 4 & 3 & 2 \\ & & 3 & 7 \\ & 3 & \end{pmatrix}, \quad f(\mathbf{X}) = 49. \tag{18.28i}$$

The next matrix $\tilde{\mathbf{C}}$ does not contain any negative element. So, $\tilde{\mathbf{X}}$ is an optimal solution.

18.1.4.2 Assignment Problem

The representation is made by an example.

■ n shipping contracts should be given to n shipping companies so that each company receives exactly one contract. The assignment has to be determined which minimizes the total costs, if the i-th company charges c_{ij} for the j-th contract.
An assignment problem is a special transportation problem with $m = n$ and $a_i = b_j = 1$ for all i, j:

$$\mathbf{OF}: \quad f(\underline{\mathbf{x}}) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} = \min! \tag{18.29a}$$

$$\mathbf{CT}: \quad \sum_{j=1}^{n} x_{ij} = 1 \ \ (i = 1, \ldots, n), \qquad \sum_{i=1}^{n} x_{ij} = 1 \ \ (j = 1, \ldots, n), \qquad x_{ij} \in \{0, 1\}. \tag{18.29b}$$

Every feasible distribution matrix contains exactly one 1 in every row and every column, all other elements are equal to zero. In a general transportation problem of this dimension, however, a non-degenerate basic solution would have $2n - 1$ positive variables. Thus, basic feasible solutions to the assignment problem are highly degenerate, with $n - 1$ basic variables equal to zero. Starting with a feasible distribution matrix $\mathbf{X}$, the assignment problem can be solved by the general transportation algorithm. It is time consuming to do so. However, because of the highly degenerate nature of the basic feasible solutions, the assignment problem can be solved with the highly efficient *Hungarian method* (see [18.9]).

18.1.4.3 Distribution Problem

The problem is represented by an example.

■ m products $E_1, E_2, \ldots, E_m$ should be produced in quantities $a_1, a_2, \ldots, a_m$. Every product can be produced on any of n machines $M_1, M_2, \ldots, M_n$. The production of a unit of product E_i on machine M_j needs processing time b_{ij} and cost c_{ij}. The time capacity of machine M_j is b_j. Denote the quantity produced by machine M_j from product E_i by x_{ij}. The total production costs should be minimized.
This distribution problem has the following general model:

$$\mathbf{OF}: \quad f(\underline{\mathbf{x}}) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} = \min! \tag{18.30a}$$

$$\mathbf{CT}: \quad \sum_{j=1}^{m} x_{ij} = a_i \ \ (i = 1, \ldots, m), \quad \sum_{i=1}^{n} b_{ij} x_{ij} \leq b_j \ \ (j = 1, \ldots, n), \quad x_{ij} \geq 0 \ \text{ for all } \ i, j. \tag{18.30b}$$

The distribution problem is a generalization of the transportation problem and it can be solved by the simplex method. If all $b_{ij} = 1$, then the more effective transportation algorithm can be used (see 18.1.4.1, p. 921) after introducing a fictitious product E_{m+1} (see 18.1.4.1, p. 920).

18.1.4.4 Travelling Salesman

Suppose there are n places $O_1, O_2, \ldots, O_n$. The travelling time from O_i to O_j is c_{ij}. Here, $c_{ij} \neq c_{ji}$ is possible.
One wants to determine the shortest route such that the traveller passes through every place exactly once, and returns to the starting point.
Similarly to the assignment problem, exactly one element is chosen in every row and column of the time

matrix $\mathbf{C}$ so that the sum of the chosen elements is minimal. The difficulty of the numerical solution of this problem is the restriction that the marked elements c_{ij} should be arranged in order of the following form:

$$c_{i_1,i_2}, c_{i_2,i_3}, \ldots, c_{i_n,i_{n+1}} \quad \text{with } i_k \neq i_l \text{ for } k \neq l \text{ and } i_{n+1} = i_1. \tag{18.31}$$

The travelling salesman problem can be solved by the branch and bound methods.

18.1.4.5 Scheduling Problem

n different products are processed on m different machines in a product-dependent order. At any time only one product can be processed on a machine. The processing time of each product on each machine is assumed to be known. Waiting times, when a given product is not in process, and machine idle times are also possible.

An optimal scheduling of the processing jobs is determined where the objective function is selected as the time when all jobs are finished, or the total waiting time of jobs, or total machine idle time. Sometimes the sum of the finishing times for all jobs is chosen as the objective function when no waiting time or idle time is allowed.

18.2 Non-linear Optimization

18.2.1 Formulation of the Problem, Theoretical Basis

18.2.1.1 Formulation of the Problem

1. Non-linear Optimization Problem

A non-linear optimization problem has the general form

$$f(\underline{\mathbf{x}}) = \min! \quad \text{subject to} \quad \underline{\mathbf{x}} \in \mathbb{R}^n \quad \text{with} \tag{18.32a}$$

$$g_i(\underline{\mathbf{x}}) \leq 0, \qquad i \in I = \{1, \ldots, m\}, \quad h_j(\underline{\mathbf{x}}) = 0, \quad j \in J = \{1, \ldots, r\} \tag{18.32b}$$

where at least one of the functions f, g_i, h_j is non-linear. The set of feasible solutions is denoted by

$$M = \{\underline{\mathbf{x}} \in \mathbb{R}^n : \; g_i(\underline{\mathbf{x}}) \leq 0, \quad i \in I, \quad h_j(\underline{\mathbf{x}}) = 0, \quad j \in J\}. \tag{18.33}$$

The problem is to determine the minimum points.

2. Minimum Points

A point $\underline{\mathbf{x}}^* \in M$ is called the *global minimum point* if $f(\underline{\mathbf{x}}^*) \leq f(\underline{\mathbf{x}})$ holds for every $\underline{\mathbf{x}} \in M$. If this relation holds for only the points $\underline{\mathbf{x}}$ of a neighborhood U of $\underline{\mathbf{x}}^*$, then $\underline{\mathbf{x}}^*$ is called a *local minimum point*. Since the equality constraints $h_j(\underline{\mathbf{x}}) = 0$ can be expressed by two inequalities,

$$-h_j(\underline{\mathbf{x}}) \leq 0, \quad h_j(\underline{\mathbf{x}}) \leq 0, \tag{18.34}$$

it can be supposed that the set J is empty, $J = \emptyset$.

18.2.1.2 Optimality Conditions

1. Special Directions

a) The Cone of the Feasible Directions at $\underline{\mathbf{x}} \in M$ is defined by

$$Z(\underline{\mathbf{x}}) = \{\underline{\mathbf{d}} \in \mathbb{R}^n : \; \exists \bar{\alpha} > 0 : \; \underline{\mathbf{x}} + \alpha \underline{\mathbf{d}} \in M, \quad 0 \leq \alpha \leq \bar{\alpha}\}, \quad \underline{\mathbf{x}} \in M, \tag{18.35}$$

where the directions are denoted by $\underline{\mathbf{d}}$. If $\underline{\mathbf{d}} \in Z(\underline{\mathbf{x}})$, then every point of the ray $\underline{\mathbf{x}} + \alpha \underline{\mathbf{d}}$ belongs to M for sufficient small values of α.

b) A Descent Direction at a point $\underline{\mathbf{x}}$ is a vector $\underline{\mathbf{d}} \in \mathbb{R}^n$ for which there exists an $\bar{\alpha} > 0$ such that

$$f(\underline{\mathbf{x}} + \alpha \underline{\mathbf{d}}) < f(\underline{\mathbf{x}}) \quad \forall \alpha \in (0, \bar{\alpha}). \tag{18.36}$$

There exists no feasible descent direction at a minimum point.

If f is differentiable, then $\underline{\mathbf{d}}$ is a descent direction when $\nabla f(\underline{\mathbf{x}})^{\mathrm{T}} \underline{\mathbf{d}} < 0$. Here, ∇ denotes the nabla operator, so $\nabla f(\underline{\mathbf{x}})$ represents the gradient of the scalar-valued function f at $\underline{\mathbf{x}}$.

2. Necessary Optimality Conditions

If f is differentiable and $\underline{\mathbf{x}}^*$ is a local minimum point, then

$$\nabla f(\underline{\mathbf{x}}^*)^{\mathrm{T}} \underline{\mathbf{d}} \geq 0 \quad \text{for every} \quad \underline{\mathbf{d}} \in \overline{Z}(\underline{\mathbf{x}}^*). \tag{18.37a}$$

In particular, if $\underline{\mathbf{x}}^*$ is an interior point of M, then

$$\nabla f(\underline{\mathbf{x}}^*) = \underline{\mathbf{0}}. \tag{18.37b}$$

3. Lagrange Function and Saddle Point

Optimality conditions (18.37a,b) should be transformed into a more practical form including the constraints. The so-called Lagrange function or *Lagrangian* is constructed:

$$L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = f(\underline{\mathbf{x}}) + \sum_{i=1}^{m} u_i g_i(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) + \underline{\mathbf{u}}^{\mathrm{T}} g(\underline{\mathbf{x}}), \quad \underline{\mathbf{x}} \in \mathbb{R},\ \underline{\mathbf{u}} \in \mathbb{R}^m_+, \tag{18.38}$$

according to the *Lagrange multiplier method* (see 6.2.5.6, p. 456) for problems with equality constraints. A point $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*) \in \mathbb{R}^n \times \mathbb{R}^m_+$ is called a *saddle point* of L, if

$$L(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}) \le L(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*) \le L(\underline{\mathbf{x}}, \underline{\mathbf{u}}^*) \quad \text{for every } \underline{\mathbf{x}} \in \mathbb{R}^n,\ \underline{\mathbf{u}} \in \mathbb{R}^m_+. \tag{18.39}$$

4. Global Kuhn-Tucker Conditions

A point $\underline{\mathbf{x}}^* \in \mathbb{R}^n$ satisfies the *global Kuhn-Tucker conditions* if there is an $\underline{\mathbf{u}}^* \in \mathbb{R}^m_+$, i.e., $\underline{\mathbf{u}}^* \ge 0$ such that $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*)$ is a saddle point of L.
For the proof of the Kuhn-Tucker conditions see 12.5.6, p. 683.

5. Sufficient Optimality Condition

If $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*) \in \mathbb{R}^n \times \mathbb{R}^m_+$ is a saddle point of L, then $\underline{\mathbf{x}}^*$ is a global minimum point of (18.32a,b).
If the functions f and g_i are differentiable, then local optimality conditions can be deduced.

6. Local Kuhn-Tucker Conditions

A point $\underline{\mathbf{x}}^* \in M$ satisfies the local Kuhn–Tucker conditions if there are numbers $u_i \ge 0$, $i \in I_0(\underline{\mathbf{x}}^*)$ such that

$$-\nabla f(\underline{\mathbf{x}}^*) = \sum_{i \in I_0(\underline{\mathbf{x}}^*)} u_i \nabla g_i(\underline{\mathbf{x}}^*), \quad \text{where} \tag{18.40a}$$

$$I_0(\underline{\mathbf{x}}) = \{i \in \{1, \dots, m\} \ :\ g_i(\underline{\mathbf{x}}) = 0\} \tag{18.40b}$$

is the index set of the active constraints at $\underline{\mathbf{x}}$. The point $\underline{\mathbf{x}}^*$ is also called a *Kuhn-Tucker stationary point.*

This means geometrically that a point $\underline{\mathbf{x}}^* \in M$ satisfies the local Kuhn-Tucker conditions, if the negative gradient $-\nabla f(\underline{\mathbf{x}}^*)$ lies in the cone spanned by the gradients $\nabla g_i(\underline{\mathbf{x}}^*)$ $i \in I_0(\underline{\mathbf{x}}^*)$ of the constraints active at $\underline{\mathbf{x}}^*$ **(Fig. 18.5)**.
The following equivalent formulation for (18.40a,b) is also often used: $\underline{\mathbf{x}}^* \in \mathbb{R}^n$ satisfies the local Kuhn-Tucker conditions, if there is a $\underline{\mathbf{u}}^* \in \mathbb{R}^m_+$ such that

$$g(\underline{\mathbf{x}}^*) \le 0, \tag{18.41a}$$

$$u_i g_i(\underline{\mathbf{x}}^*) = 0, \quad i = 1, \dots, m, \tag{18.41b}$$

$$\nabla f(\underline{\mathbf{x}}^*) + \sum_{i=1}^{m} u_i \nabla g_i(\underline{\mathbf{x}}^*) = 0. \tag{18.41c}$$

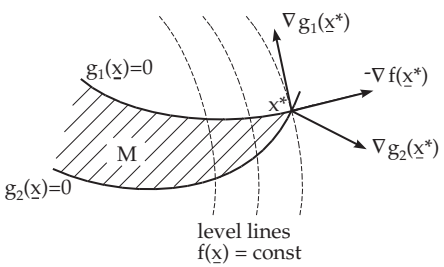

Figure 18.5

7. Necessary Optimality Conditions and Kuhn-Tucker Conditions

If $\underline{\mathbf{x}}^* \in M$ is a local minimum point of (18.32a,b) and the feasible set satisfies the *regularity condition* at $\underline{\mathbf{x}}^*$: $\exists\, \underline{\mathbf{d}} \in \mathbb{R}^n$ such that $\nabla g_i(\underline{\mathbf{x}}^*)^{\mathrm{T}} \underline{\mathbf{d}} < 0$ for every $i \in I_0(\underline{\mathbf{x}}^*)$, then $\underline{\mathbf{x}}^*$ satisfies the local Kuhn-Tucker conditions.

18.2.1.3 Duality in Optimization

1. Dual Problem

With the associated Lagrangian (18.38) the maximum problem is formed, the so-called dual of (18.32a,b):

$$L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \max! \quad \text{subject to} \quad (\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in M^* \quad \text{with} \tag{18.42a}$$

$$M^* = \{(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in \mathbb{R}^n \times \mathbb{R}_+^m \ : \ L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \min_{\underline{\mathbf{z}} \in \mathbb{R}^n} L(\underline{\mathbf{z}}, \underline{\mathbf{u}})\}. \tag{18.42b}$$

2. Duality Theorems

If $\underline{\mathbf{x}}_1 \in M$ and $(\underline{\mathbf{x}}_2, \underline{\mathbf{u}}_2) \in M^*$, then

a) $L(\underline{\mathbf{x}}_2, \underline{\mathbf{u}}_2) \leq f(\underline{\mathbf{x}}_1)$.

b) If $L(\underline{\mathbf{x}}_2, \underline{\mathbf{u}}_2) = f(\underline{\mathbf{x}}_1)$, then $\underline{\mathbf{x}}_1$ is a minimum point of (18.32a,b) and $(\underline{\mathbf{x}}_2, \underline{\mathbf{u}}_2)$ is a maximum point of (18.42a,b).

18.2.2 Special Non-linear Optimization Problems

18.2.2.1 Convex Optimization

1. Convex Problem

The optimization problem

$$f(\underline{\mathbf{x}}) = \min! \quad \text{subject to} \quad g_i(\underline{\mathbf{x}}) \leq 0 \quad (i = 1, \ldots, m) \tag{18.43}$$

is called a *convex problem* if the functions f and g_i are convex. In particular, f and g_i can be linear functions. The following statements are valid for convex problems:

a) Every local minimum of f over M is also a global minimum.

b) If M is not empty and bounded, then there exists at least one solution of (18.43).

c) If f is strictly convex, then there is at most one solution of (18.43).

1. Optimality Conditions

a) If f has continuous partial derivatives, then $\underline{\mathbf{x}}^* \in M$ is a solution of (18.43), if

$$(\underline{\mathbf{x}} - \underline{\mathbf{x}}^*)^{\mathrm{T}} \nabla f(\underline{\mathbf{x}}^*) \geq 0 \quad \text{for every} \quad \underline{\mathbf{x}} \in M. \tag{18.44}$$

b) The *Slater condition* is a regularity condition for the feasible set M. It is satisfied if there exists an $\underline{\mathbf{x}} \in M$ such that $g_i(\underline{\mathbf{x}}) < 0$ for every non-affine linear functions g_i.

c) If the Slater condition is satisfied, then $\underline{\mathbf{x}}^*$ is a minimum point of (18.43) if and only if there exists a $\underline{\mathbf{u}}^* \geq 0$ such that $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*)$ is a saddle point of the Lagrangian. Moreover, if functions f and g_i are differentiable, then $\underline{\mathbf{x}}^*$ is a solution of (18.43) if and only if $\underline{\mathbf{x}}^*$ satisfies the local Kuhn-Tucker conditions.

d) The dual problem (18.42a,b) can be formulated easily for a convex optimization problem with differentiable functions f and g_i:

$$L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \max!, \quad \text{subject to} \quad (\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in M^* \quad \text{with} \tag{18.45a}$$

$$M^* = \{(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in \mathbb{R}^n \times \mathbb{R}_+^m \ : \ \nabla_{\underline{\mathbf{x}}} L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \underline{\mathbf{0}}\}. \tag{18.45b}$$

The gradient of L is calculated here only with respect to $\underline{\mathbf{x}}$.

e) For convex optimization problems, the *strong duality theorem* also holds:

If M satisfies the Slater condition and if $\underline{\mathbf{x}}^* \in M$ is a solution of (18.43), then there exists a $\underline{\mathbf{u}}^* \in \mathbb{R}_+^m$, such that $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*)$ is a solution of the dual problem (18.45a,b), and

$$f(\underline{\mathbf{x}}^*) = \min_{\underline{\mathbf{x}} \in M} f(\underline{\mathbf{x}}) = \max_{(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in M^*} L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = L(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*). \tag{18.46}$$

18.2.2.2 Quadratic Optimization

1. Formulation of the Problem

Quadratic optimization problems have the form

$$f(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}} \mathbf{C} \underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}} \underline{\mathbf{x}} = \min!, \quad \text{subject to} \quad \underline{\mathbf{x}} \in M \subset \mathbb{R}^n \quad \text{with} \tag{18.47a}$$

$$M = M_I: \quad M = \{\underline{\mathbf{x}} \in \mathbb{R}^n \ : \ \mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}}, \ \underline{\mathbf{x}} \geq \underline{\mathbf{0}}\}. \tag{18.47b}$$

Here, $\mathbf{C}$ is a symmetric (n,n) matrix, $\underline{\mathbf{p}} \in \mathbb{R}^n$, $\mathbf{A}$ is an (m,n) matrix, and $\underline{\mathbf{b}} \in \mathbb{R}^m$.
The feasible set M can be written alternatively in the following way:

$$M = M_{II}: \quad M = \{\underline{\mathbf{x}} \,:\, \mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}},\ \underline{\mathbf{x}} \geq \underline{\mathbf{0}}\}, \tag{18.48a}$$

$$M = M_{III}: \quad M = \{\underline{\mathbf{x}} \,:\, \mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}}\}. \tag{18.48b}$$

2. Lagrangian and Kuhn-Tucker Conditions

The Lagrangian to the problem (18.47a,b) is

$$L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}} + \underline{\mathbf{u}}^{\mathrm{T}}(\mathbf{A}\underline{\mathbf{x}} - \underline{\mathbf{b}}). \tag{18.49}$$

By introducing the notation

$$\underline{\mathbf{v}} = \frac{\partial L}{\partial \underline{\mathbf{x}}} = \underline{\mathbf{p}} + 2\mathbf{C}\underline{\mathbf{x}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} \quad \text{and} \quad \underline{\mathbf{y}} = -\frac{\partial L}{\partial \underline{\mathbf{u}}} = -\mathbf{A}\underline{\mathbf{x}} + \underline{\mathbf{b}} \tag{18.50}$$

the Kuhn-Tucker conditions are as follows:

	Case I:	**Case II:**	**Case III:**	
a)	$\mathbf{A}\underline{\mathbf{x}} + \underline{\mathbf{y}} = \underline{\mathbf{b}}$,	$\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$,	$\mathbf{A}\underline{\mathbf{x}} + \underline{\mathbf{y}} = \underline{\mathbf{b}}$,	(18.51a)
b)	$2\mathbf{C}\underline{\mathbf{x}} - \underline{\mathbf{v}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} = -\underline{\mathbf{p}}$,	$2\mathbf{C}\underline{\mathbf{x}} - \underline{\mathbf{v}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} = -\underline{\mathbf{p}}$,	$2\mathbf{C}\underline{\mathbf{x}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} = -\underline{\mathbf{p}}$,	(18.51b)
c)	$\underline{\mathbf{x}} \geq \underline{\mathbf{0}},\ \underline{\mathbf{v}} \geq \underline{\mathbf{0}},\ \underline{\mathbf{y}} \geq \underline{\mathbf{0}},\ \underline{\mathbf{u}} \geq \underline{\mathbf{0}}$,	$\underline{\mathbf{x}} \geq \underline{\mathbf{0}},\ \underline{\mathbf{v}} \geq \underline{\mathbf{0}}$,	$\underline{\mathbf{u}} \geq \underline{\mathbf{0}},\ \underline{\mathbf{y}} \geq \underline{\mathbf{0}}$,	(18.51c)
d)	$\underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{v}} + \underline{\mathbf{y}}^{\mathrm{T}}\underline{\mathbf{u}} = 0$.	$\underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{v}} = 0$.	$\underline{\mathbf{y}}^{\mathrm{T}}\underline{\mathbf{u}} = 0$.	(18.51d)

3. Convexity

The function $f(\underline{\mathbf{x}})$ is convex (strictly convex) if and only if the matrix $\mathbf{C}$ is positive semidefinite (positive definite). Every result on convex optimization problems can be used for quadratic problems with a positive semidefinite matrix $\mathbf{C}$; in particular, the Slater condition always holds, so it is necessary and sufficient for the optimality of a point $\underline{\mathbf{x}}^*$ that there exists a point $(\underline{\mathbf{x}}^*, \underline{\mathbf{y}}, \underline{\mathbf{u}}, \underline{\mathbf{v}})$, which satisfies the corresponding system of local Kuhn-Tucker conditions.

4. Dual Problem

If $\mathbf{C}$ is positive definite, then the dual problem (18.45a,b) of (18.47a,b) can be expressed explicitly:

$$L(\underline{\mathbf{x}}, \underline{\mathbf{u}}) = \max!, \quad \text{subject to} \quad (\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in M^*, \quad \text{where} \tag{18.52a}$$

$$M^* = \{(\underline{\mathbf{x}}, \underline{\mathbf{u}}) \in \mathbb{R}^n \times \mathbb{R}^m_+ \,:\, \underline{\mathbf{x}} = -\frac{1}{2}\mathbf{C}^{-1}(\mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} + \underline{\mathbf{p}})\}. \tag{18.52b}$$

If the expression $\underline{\mathbf{x}} = -\frac{1}{2}\mathbf{C}^{-1}(\mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} + \underline{\mathbf{p}})$ is substituted into the dual objective function $L(\underline{\mathbf{x}}, \underline{\mathbf{u}})$, then the equivalent problem is

$$\varphi(\underline{\mathbf{u}}) = -\frac{1}{4}\underline{\mathbf{u}}^{\mathrm{T}}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} - \left(\frac{1}{2}\mathbf{A}\mathbf{C}^{-1}\underline{\mathbf{p}} + \underline{\mathbf{b}}\right)^{\mathrm{T}}\underline{\mathbf{u}} - \frac{1}{4}\underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{C}}^{-1}\underline{\mathbf{p}} = \max!, \quad \underline{\mathbf{u}} \geq \underline{\mathbf{0}}. \tag{18.53}$$

Hence: If $\underline{\mathbf{x}}^* \in M$ is a solution of (18.47a,b), then (18.53) has a solution $\underline{\mathbf{u}}^* \geq \underline{\mathbf{0}}$, and

$$f(\underline{\mathbf{x}}^*) = \varphi(\underline{\mathbf{u}}^*). \tag{18.54}$$

Problem (18.53) can be replaced by an equivalent formulation:

$$\psi(\underline{\mathbf{u}}) = \underline{\mathbf{u}}^{\mathrm{T}}\mathbf{E}\underline{\mathbf{u}} + \underline{\mathbf{h}}^{\mathrm{T}}\underline{\mathbf{u}} = \min!\,, \quad \text{subject to} \quad \underline{\mathbf{u}} \geq \underline{\mathbf{0}} \quad \text{where} \tag{18.55a}$$

$$\mathbf{E} = \frac{1}{4}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}^{\mathrm{T}} \quad \text{and} \quad \underline{\mathbf{h}} = \frac{1}{2}\mathbf{A}\mathbf{C}^{-1}\underline{\mathbf{p}} + \underline{\mathbf{b}}\,. \tag{18.55b}$$

18.2.3 Solution Methods for Quadratic Optimization Problems

18.2.3.1 Wolfe's Method

1. Formulation of the Problem and Solution Principle

The method of Wolfe is to solve quadratic problems of the special form:

$$f(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}} = \min!\,, \quad \text{subject to} \quad \mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}, \quad \underline{\mathbf{x}} \geq \underline{\mathbf{0}}. \tag{18.56}$$

$\mathbf{C}$ is supposed to be positive definite. The basic idea is the determination of a solution $(\underline{\mathbf{x}}^*, \underline{\mathbf{u}}^*, \underline{\mathbf{v}}^*)$ of the corresponding system of Kuhn-Tucker conditions, associated to problem (18.56):

$$\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}, \tag{18.57a}$$

$$2\mathbf{C}\underline{\mathbf{x}} - \underline{\mathbf{v}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} = -\underline{\mathbf{p}}, \tag{18.57b}$$

$$\underline{\mathbf{x}} \geq \underline{\mathbf{0}}, \quad \underline{\mathbf{v}} \geq \underline{\mathbf{0}}; \tag{18.57c}$$

$$\underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{v}} = 0. \tag{18.58}$$

Relations (18.57a,b,c) represent a linear equation system with $m + n$ equations and $2n + m$ variables. Because of relation (18.58), either $x_i = 0$ or $v_i = 0$ $(i = 1, 2, \ldots, n)$ must hold. Therefore, every solution of (18.57a,b,c), (18.58) contains at most $m+n$ non-zero components. Hence, it must be a basic solution of (18.57a,b,c).

2. Solution Process

First, a feasible basic solution (vertex) $\bar{\underline{\mathbf{x}}}$ of the system $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ is determined. The indices belonging to the basis variables of $\bar{\underline{\mathbf{x}}}$ form the set I_B. In order to find a solution of system (18.57a,b,c), which also satisfies (18.58), the problem is formulated as

$$-\mu = \min!\,, \quad (\mu \in \mathbb{R}); \tag{18.59}$$

$$\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}, \tag{18.60a}$$

$$2\mathbf{C}\underline{\mathbf{x}} - \underline{\mathbf{v}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} - \mu\underline{\mathbf{q}} = -\underline{\mathbf{p}} \quad \text{with} \quad \underline{\mathbf{q}} = 2\mathbf{C}\bar{\underline{\mathbf{x}}} + \underline{\mathbf{p}}, \tag{18.60b}$$

$$\underline{\mathbf{x}} \geq \underline{\mathbf{0}}, \ \ \underline{\mathbf{v}} \geq \underline{\mathbf{0}}, \ \ \mu \geq 0; \tag{18.60c}$$

$$\underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{v}} = 0. \tag{18.61}$$

If $(\underline{\mathbf{x}}, \underline{\mathbf{v}}, \underline{\mathbf{u}}, \mu)$ is a solution of this problem also satisfying (18.57a,b,c) and (18.58), then $\mu = 0$.
The vector $(\underline{\mathbf{x}}, \underline{\mathbf{v}}, \underline{\mathbf{u}}, \mu) = (\bar{\underline{\mathbf{x}}}, \underline{\mathbf{0}}, \underline{\mathbf{0}}, 1)$ is a known feasible solution of the system (18.60a,b,c), and it satisfies the relation (18.61), too. A basis associated to this basic solution is formed from the columns of the coefficient matrix

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} & \underline{\mathbf{0}} \\ 2\mathbf{C} & -\mathbf{I} & \mathbf{A}^{\mathrm{T}} & -\underline{\mathbf{q}} \end{pmatrix}, \quad \begin{array}{l} \mathbf{I} \text{ denotes the unit matrix, } \mathbf{0} \text{ the zero matrix and } \underline{\mathbf{0}} \\ \text{is the zero vector of the corresponding dimension,} \end{array} \tag{18.62}$$

in the following way:
a) m columns belonging to x_i with $i \in I_B$,
b) $n - m$ columns belonging to v_i with $i \notin I_B$,
c) all m columns belonging to u_i,
d) the last column, but then a suitable column determined in **b)** or **c)** will be dropped.
If $\underline{\mathbf{q}} = \underline{\mathbf{0}}$, then the interchange according to **d)** is not possible. Then $\bar{\underline{\mathbf{x}}}$ is already a solution.
Now, a first simplex tableau can be constructed. The minimization of the objective function is performed by the simplex method with an additional rule that guarantees that the relation $\underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{v}} = 0$ is satisfied:
The variables x_i and v_i $(i = 1, 2, \ldots, n)$ must not be simultaneously basic variables.
In the case of a positive definite $\mathbf{C}$, considering this additional rule the simplex method provides a solution of problem (18.59), (18.60a,b,c), (18.61) satisfying $\mu = 0$. For a positive semi-definite matrix $\mathbf{C}$, because of the restricted pivot choice, it may happen that although $\mu > 0$, no more exchange-step can be made without violating the additional rules. In this case μ cannot be reduced any further.

■ $f(\underline{\mathbf{x}}) = x_1^2 + 4x_2^2 - 10x_1 - 32x_2 = \min!$ with $x_1 + 2x_2 + x_3 = 7, \quad 2x_1 + x_2 + x_4 = 8.$

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix}, \quad \underline{\mathbf{b}} = \begin{pmatrix} 7 \\ 8 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \underline{\mathbf{p}} = \begin{pmatrix} -10 \\ -32 \\ 0 \\ 0 \end{pmatrix}.$$

In this case $\mathbf{C}$ is positive semi-definite. A feasible basic solution of $\mathbf{A}\underline{\mathbf{x}} = \underline{\mathbf{b}}$ is $\bar{\underline{\mathbf{x}}} = (0, 0, 7, 8)^{\mathrm{T}}$, $\underline{\mathbf{q}} = 2\mathbf{C}\bar{\underline{\mathbf{x}}} + \underline{\mathbf{p}} = (-10, -32, 0, 0)^{\mathrm{T}}$. The choices for the basis vectors are: **a)** columns 3 and 4 of $\begin{pmatrix} \mathbf{A} \\ 2\mathbf{C} \end{pmatrix}$, **b)** columns 1 and 2 of $\begin{pmatrix} \mathbf{0} \\ -\mathbf{I} \end{pmatrix}$, **c)** the columns of $\begin{pmatrix} \mathbf{0} \\ \mathbf{A}^{\mathrm{T}} \end{pmatrix}$ and **d)** column $\begin{pmatrix} \underline{\mathbf{0}} \\ -\underline{\mathbf{q}} \end{pmatrix}$ instead of the first column of $\begin{pmatrix} \mathbf{0} \\ -\mathbf{I} \end{pmatrix}$. The basis matrix is formed from these columns, and the basis inverse is calculated (see 18.1, p. 909). Multiplying matrix (18.62) and the vectors $\begin{pmatrix} \underline{\mathbf{b}} \\ -\underline{\mathbf{p}} \end{pmatrix}$ by the basis inverse, the first simplex tableau **(Scheme 18.9)** is obtained.
Only x_1 can be interchanged with v_2 in this tableau according to the complementary constraints. After a few steps, we get the solution $\underline{\mathbf{x}}^* = (2, 5/2, 0, 3/2)^{\mathrm{T}}$ is obtained. The last two equations of $2\mathbf{C}\underline{\mathbf{x}} - \underline{\mathbf{v}} + \mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} - \mu\underline{\mathbf{q}} = -\underline{\mathbf{p}}$ are: $v_3 = u_1$, $v_4 = u_2$. Therefore, by eliminating u_1 and u_2 the dimension of the problem can be reduced.

Scheme 18.9

	x_1	x_2	v_1	v_3	v_4	
x_3	1	2	0	0	0	7
x_4	2	1	0	0	0	8
v_2	$\boxed{\frac{64}{10}}$	-8	$-\frac{32}{10}$	$\frac{12}{10}$	$\frac{54}{10}$	0
u_1	0	0	0	-1	0	0
u_2	0	0	0	0	-1	0
μ	$\frac{2}{10}$	0	$-\frac{1}{10}$	$\frac{1}{10}$	$\frac{2}{10}$	1
	$-\frac{2}{10}$	0	$\frac{1}{10}$	$-\frac{1}{10}$	$-\frac{2}{10}$	-1

18.2.3.2 Hildreth-d'Esopo Method

1. Principle

The strictly convex optimization problem

$$f(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}} = \min!, \quad \mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}} \tag{18.63}$$

has the dual problem (see 1., p. 926)

$$\psi(\underline{\mathbf{u}}) = \underline{\mathbf{u}}^{\mathrm{T}}\mathbf{E}\underline{\mathbf{u}} + \underline{\mathbf{h}}^{\mathrm{T}}\underline{\mathbf{u}} = \min! \quad \underline{\mathbf{u}} \geq 0 \quad \text{with} \tag{18.64a}$$

$$\mathbf{E} = \frac{1}{4}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}^{\mathrm{T}}, \quad \underline{\mathbf{h}} = \frac{1}{2}\mathbf{A}\mathbf{C}^{-1}\underline{\mathbf{p}} + \underline{\mathbf{b}}. \tag{18.64b}$$

Matrix $\mathbf{E}$ is positive definite and it has positive diagonal elements $e_{ii} > 0$, $(i = 1, 2, \ldots, m)$. The variables $\underline{\mathbf{x}}$ and $\underline{\mathbf{u}}$ satisfy the following relation:

$$\underline{\mathbf{x}} = -\frac{1}{2}\mathbf{C}^{-1}(\mathbf{A}^{\mathrm{T}}\underline{\mathbf{u}} + \underline{\mathbf{p}}). \tag{18.65}$$

2. Solution by Iteration

The dual problem (18.64a), which contains only the condition $\underline{\mathbf{u}} \geq \underline{\mathbf{0}}$, can be solved by the following simple iteration method:
a) Substitute $\underline{\mathbf{u}}^1 \geq \underline{\mathbf{0}}$, (e.g., $\underline{\mathbf{u}}^1 = \underline{\mathbf{0}}$), $k = 1$.
b) Calculate u_i^{k+1} for $i = 1, 2, \ldots, m$ according to

$$w_i^{k+1} = -\frac{1}{e_{ii}}\left(\sum_{j=1}^{i-1} e_{ij}u_j^{k+1} + \frac{h_i}{2} + \sum_{j=i+1}^{m} e_{ij}u_j^k\right), \tag{18.66a}$$

$$u_i^{k+1} = \max\left\{0, w_i^{k+1}\right\}. \tag{18.66b}$$

c) Repeat step **b)** with $k+1$ instead of k until a stopping rule is satisfied, e.g., $\left|\psi(\underline{\mathbf{u}}^{k+1}) - \psi(\underline{\mathbf{u}}^{k})\right| < \varepsilon$, $\varepsilon > 0$.
Under the assumption that there is an $\underline{\mathbf{x}}$ such that $\mathbf{A}\underline{\mathbf{x}} < \underline{\mathbf{b}}$, the sequence $\{\psi(\underline{\mathbf{u}}^k)\}$ converges to the minimum value $\psi_{\min}$ and sequence $\{\underline{\mathbf{x}}^k\}$ given by (18.65) converges to the solution $\underline{\mathbf{x}}^*$ of the original problem. The sequence $\{\underline{\mathbf{u}}^k\}$ is not always convergent.

18.2.4 Numerical Search Procedures

By using non-linear optimization procedures acceptable approximate solutions can be found with reasonable computing costs for several types of optimization problems. They are based on the principle of comparison of function values.

18.2.4.1 One-Dimensional Search

Several optimization methods contain the subproblem of finding the minimum of a real function $f(x)$ for $x \in [a,b]$. It is often sufficient to find an approximation $\overline{x}$ of the minimum point x^*.

1. Formulation of the Problem

A function $f(x)$, $x \in \mathbb{R}$, is called unimodal in $[a,b]$ if it has exactly one local minimum point on every closed subinterval $J \subseteq [a,b]$. Let f be a unimodal function on $[a,b]$ and x^* the global minimum point. Then an interval $[c,d] \subseteq [a,b]$ should be found with $x^* \in [c,d]$ such that $d - c < \varepsilon$, $\varepsilon > 0$.

2. Uniform Search

A positive integer n is chosen such that $\delta = \dfrac{b-a}{n+1} < \dfrac{\varepsilon}{2}$, and the values $f(x^k)$ for $x^k = a + k\delta$ $(k = 1, \ldots, n)$ are calculated. If $f(x)$ is the smallest value among these function values, then the minimum point x^* is in the interval $[x-\delta, x+\delta]$. The number of required function values for the given accuracy can be estimated by

$$n > \frac{2(b-a)}{\varepsilon} - 1\,. \tag{18.67}$$

3. Golden Section Method, Fibonacci Method

The interval $[a,b] = [a_1,b_1]$ will be reduced step by step so that the new subinterval always contains the minimum point x^*. The points λ_1, μ_1 are determined in the interval $[a_1,b_1]$ as

$$\lambda_1 = a_1 + (1-\tau)(b_1 - a_1), \quad \mu_1 = a_1 + \tau(b_1 - a_1) \quad \text{with} \tag{18.68a}$$

$$\tau = \frac{1}{2}(\sqrt{5} - 1) \approx 0.618. \tag{18.68b}$$

This corresponds to the golden section. Two cases are distinguished:

a) If $f(\lambda_1) < f(\mu_1)$, then $a_2 = a_1$, $b_2 = \mu_1$ and $\mu_2 = \lambda_1$ are substituted. (18.69a)

b) If $f(\lambda_1) \geq f(\mu_1)$, then $a_2 = \lambda_1$, $b_2 = b_1$ and $\lambda_2 = \mu_1$ are substituted. (18.69b)

If $b_2 - a_2 \geq \varepsilon$, then the procedure is repeated with the interval $[a_2,b_2]$, where one value is already known, $f(\lambda_2)$ in case **a)** and $f(\mu_2)$ in case **b)**, from the first step. To determine an interval $[a_n,b_n]$, which contains the minimum point x^*, altogether n function values are calculated. From the requirement

$$\varepsilon > b_n - a_n = \tau^{n-1}(b_1 - a_1) \tag{18.70}$$

the necessary number of steps n can be estimated.
By using the golden section method, at most one more function value should be determined compared to the Fibonacci method. Instead of subdividing the interval according to the golden section, the interval is subdivided according to the *Fibonacci numbers* (see 5.4.1.5, p. 375, and 17.3.2.4, **4.**, p. 908).

18.2.4.2 Minimum Search in n-Dimensional Euclidean Vector Space

The search for an approximation of the minimum point $\underline{\mathbf{x}}^*$ of the problem $f(\underline{\mathbf{x}}) = \min!$, $\underline{\mathbf{x}} \in \mathbb{R}^n$, can be reduced to the solution of a sequence of one-dimensional optimization problems.
One takes

a) $\underline{\mathbf{x}} = \underline{\mathbf{x}}^1$, $k = 1$, where $\underline{\mathbf{x}}^1$ is an appropriate initial approximation of $\underline{\mathbf{x}}^*$. (18.71a)

b) The one-dimensional problems

$$\varphi(\alpha_r) = f(x_1^{k+1}, \ldots, x_{r-1}^{k+1}, x_r^k + \alpha_r, x_{r+1}^k, \ldots, x_n^k) = \min! \text{ with } \alpha_r \in \mathbb{R} \tag{18.71b}$$

are solved for $r = 1, 2, \ldots, n$. If $\bar{\alpha_r}$ is an exact or approximating minimum point of the r-th problem, then $x_r^{k+1} = x_r^k + \bar{\alpha_r}$ are substituted.

c) If two consecutive approximations are close enough to each other, i.e., with some vector norm,

$$||\underline{\mathbf{x}}^{k+1} - \underline{\mathbf{x}}^k|| < \varepsilon_1 \quad \text{or} \quad |f(\underline{\mathbf{x}}^{k+1}) - f(\underline{\mathbf{x}}^k)| < \varepsilon_2, \tag{18.71c}$$

then $\underline{\mathbf{x}}^{k+1}$ is an approximation of $\underline{\mathbf{x}}^*$. Otherwise step b) is repeated with $k+1$ instead of k. The one-dimensional problem in b) can be solved, by using the methods given in 18.2.4.1, p. 930.

18.2.5 Methods for Unconstrained Problems

The general optimization problem

$$f(\underline{\mathbf{x}}) = \min! \quad \text{for} \quad \underline{\mathbf{x}} \in \mathbb{R}^n \tag{18.72}$$

is considered with a continuously differentiable function f. Each method described in this section constructs, in general, an infinite sequence of points $\{\underline{\mathbf{x}}^k\} \in \mathbb{R}^n$, whose accumulation point is a stationary point. The sequence of points will be determined starting with a point $\underline{\mathbf{x}}^1 \in \mathbb{R}^n$ and according to the formula

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k \quad (k = 1, 2, \ldots), \tag{18.73}$$

i.e., first a direction $\underline{\mathbf{d}}^k \in \mathbb{R}^n$ is determined at $\underline{\mathbf{x}}^k$ and the *step size* $\alpha_k \in \mathbb{R}$ indicates how far $\underline{\mathbf{x}}^{k+1}$ is from $\underline{\mathbf{x}}^k$ in the direction $\underline{\mathbf{d}}^k$. Such a method is called a *descent method*, if

$$f(\underline{\mathbf{x}}^{k+1}) < f(\underline{\mathbf{x}}^k) \quad (k = 1, 2, \ldots). \tag{18.74}$$

The equality $\nabla f(\underline{\mathbf{x}}) = 0$, where ∇ is the nabla operator (see 13.2.6.1, p. 715), characterizes a stationary point and can be used as a stopping rule for the iteration method.

18.2.5.1 Method of Steepest Descent

Starting from an actual point $\underline{\mathbf{x}}^k$, the direction $\underline{\mathbf{d}}^k$ in which the function has its steepest descent is

$$\underline{\mathbf{d}}^k = -\nabla f(\underline{\mathbf{x}}^k) \qquad (18.75a) \qquad \text{and consequently} \quad \underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k - \alpha_k \nabla f(\underline{\mathbf{x}}^k). \tag{18.75b}$$

A schematic representation of the *steepest descent method* with level lines $f(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}^i)$ is shown in **Fig. 18.6**.

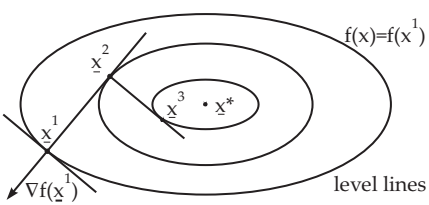

Figure 18.6

The step size α_k is determined by a line search, i.e., α_k is the solution of the one-dimensional problem:

$$f(\underline{\mathbf{x}}^k + \alpha \underline{\mathbf{d}}^k) = \min!, \quad \alpha \geq 0. \tag{18.76}$$

This problem can be solved by the methods given in 18.2.4, p. 930.

The steepest descent method (18.75b) converges relatively slowly. For every accumulation point $\underline{\mathbf{x}}^*$ of the sequence $\{\underline{\mathbf{x}}^k\}$, $\nabla f(\underline{\mathbf{x}}^*) = 0$. In the case of a quadratic objective function, i.e., $f(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}}$, the method has the special form:

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k \qquad (18.77a) \quad \text{with } \underline{\mathbf{d}}^k = -(2\mathbf{C}\underline{\mathbf{x}}^k + \underline{\mathbf{p}}) \text{ and } \alpha_k = \frac{\underline{\mathbf{d}}^{k\mathrm{T}}\underline{\mathbf{d}}^k}{2\underline{\mathbf{d}}^{k\mathrm{T}}\mathbf{C}\underline{\mathbf{d}}^k}. \tag{18.77b}$$

18.2.5.2 Application of the Newton Method

Suppose that at the actual approximation point $\underline{\mathbf{x}}^k$ the function f is approximated by a quadratic function:

$$q(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}^k) + (\underline{\mathbf{x}} - \underline{\mathbf{x}}^k)^{\mathrm{T}} \nabla f(\underline{\mathbf{x}}^k) + \frac{1}{2}(\underline{\mathbf{x}} - \underline{\mathbf{x}}^k)^{\mathrm{T}} \mathbf{H}(\underline{\mathbf{x}}^k)(\underline{\mathbf{x}} - \underline{\mathbf{x}}^k). \tag{18.78}$$

Here $\mathbf{H}(\underline{\mathbf{x}}^k)$ is the Hessian matrix, i.e., the matrix of second partial derivatives of f at the point $\underline{\mathbf{x}}^k$. If $\mathbf{H}(\underline{\mathbf{x}}^k)$ is positive definite, then $q(\underline{\mathbf{x}})$ has an absolute minimum at $\underline{\mathbf{x}}^{k+1}$ with $\nabla q(\underline{\mathbf{x}}^{k+1}) = 0$, therefore one gets the Newton method:

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k - \mathbf{H}^{-1}(\underline{\mathbf{x}}^k)\nabla f(\underline{\mathbf{x}}^k) \quad (k = 1, 2, \ldots), \quad \text{i.e.,} \tag{18.79a}$$

$$\underline{\mathbf{d}}^k = -\mathbf{H}^{-1}(\underline{\mathbf{x}}^k)\nabla f(\underline{\mathbf{x}}^k) \quad \text{and} \quad \alpha_k \text{ in (18.73).} \tag{18.79b}$$

The Newton method converges fast but it has the following disadvantages:
a) The matrix $\mathbf{H}(\underline{\mathbf{x}}^k)$ must be positive definite.
b) The method converges only for sufficiently good initial points.
c) The step size can not influenced.
d) The method is not a descent method.
e) The computational cost of computing the inverse of $\mathbf{H}^{-1}(\underline{\mathbf{x}}^k)$ is fairly high.
Some of these disadvantages can be reduced by the following version of the *damped Newton method* (see also 19.2.2.2, p. 962):

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k - \alpha_k \mathbf{H}^{-1}(\underline{\mathbf{x}}^k)\nabla f(\underline{\mathbf{x}}^k) \quad (k = 1, 2, \ldots). \tag{18.80}$$

The relaxation factor α_k can be determined, for example, by the principle given earlier (see 18.2.5.1, p. 931).

18.2.5.3 Conjugate Gradient Methods

Two vectors $\underline{\mathbf{d}}^1, \underline{\mathbf{d}}^2 \in \mathbb{R}^n$ are called *conjugate vectors* with respect to a symmetric, positive definite matrix $\mathbf{C}$, if

$$\underline{\mathbf{d}}^{1\,\mathrm{T}}\mathbf{C}\underline{\mathbf{d}}^2 = 0. \tag{18.81}$$

If $\underline{\mathbf{d}}^1, \underline{\mathbf{d}}^2, \ldots, \underline{\mathbf{d}}^n$ are pairwise conjugate vectors with respect to a matrix $\mathbf{C}$, then the convex quadratic problem $q(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}}$, $\underline{\mathbf{x}} \in \mathbb{R}^n$, can be solved in n steps if a sequence $\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k$ starting from $\underline{\mathbf{x}}^1$ is constructed, where α_k is the optimal step size. Under the assumption that $f(\underline{\mathbf{x}})$ is approximately quadratic in the neighborhood of $\underline{\mathbf{x}}^*$, i.e., $\mathbf{C} \approx \frac{1}{2}\mathbf{H}(\underline{\mathbf{x}}^*)$, the method developed for quadratic objective functions can also be applied for more general functions $f(\underline{\mathbf{x}})$, without the explicit use of the matrix $\mathbf{H}(\underline{\mathbf{x}}^*)$.
The conjugate gradient method has the following steps:

a) $\underline{\mathbf{x}}^1 \in \mathbb{R}^n$, $\underline{\mathbf{d}}^1 = -\nabla f(\underline{\mathbf{x}}^1)$, (18.82)

where $\underline{\mathbf{x}}^1$ is an appropriate initial approximation for $\underline{\mathbf{x}}^*$.

b) $\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k$ $(k = 1, \ldots, n)$ with $\alpha_k \geq 0$ so that $f(\underline{\mathbf{x}}^k + \alpha \underline{\mathbf{d}}^k)$ will be minimized. (18.83a)

$$\underline{\mathbf{d}}^{k+1} = -\nabla f(\underline{\mathbf{x}}^{k+1}) + \mu_k \underline{\mathbf{d}}^k \quad (k = 1, \ldots, n-1) \text{ with} \tag{18.83b}$$

$$\mu_k = \frac{\nabla f(\underline{\mathbf{x}}^{k+1})^{\mathrm{T}} \nabla f(\underline{\mathbf{x}}^{k+1})}{\nabla f(\underline{\mathbf{x}}^k)^{\mathrm{T}} \nabla f(\underline{\mathbf{x}}^k)} \quad \text{and} \quad \underline{\mathbf{d}}^{n+1} = -\nabla f(\underline{\mathbf{x}}^{n+1}). \tag{18.83c}$$

c) Repeating steps **b)** with $\underline{\mathbf{x}}^{n+1}$ and $\underline{\mathbf{d}}^{n+1}$ instead of $\underline{\mathbf{x}}^1$ and $\underline{\mathbf{d}}^1$.

18.2.5.4 Method of Davidon, Fletcher and Powell (DFP)

With the DFP method, a sequence of points starting from $\underline{\mathbf{x}}^1 \in \mathbb{R}^n$ is determined according to the formula

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k - \alpha_k \mathbf{M}_k \nabla f(\underline{\mathbf{x}}^k) \quad (k = 1, 2, \ldots). \tag{18.84}$$

Here, $\mathbf{M}_k$ is a symmetric, positive definite matrix. The idea of the method is a stepwise approximation of the inverse Hessian matrix by matrices $\mathbf{M}_k$ in the case when $f(\underline{\mathbf{x}})$ is a quadratic function. Starting

with a symmetric, positive definite matrix $\mathbf{M}_1$, e.g., $\mathbf{M}_1 = \mathbf{I}$ ($\mathbf{I}$ is the unit matrix), the matrix $\mathbf{M}_k$ is determined from $\mathbf{M}_{k-1}$ by adding a correction matrix of rank two

$$\mathbf{M}_k = \mathbf{M}_{k-1} + \frac{\underline{\mathbf{v}}^k \underline{\mathbf{v}}^{k\mathrm{T}}}{\underline{\mathbf{v}}^{k\mathrm{T}} \underline{\mathbf{v}}^k} - \frac{(\mathbf{M}_{k-1}\underline{\mathbf{w}}^k)(\mathbf{M}_{k-1}\underline{\mathbf{w}}^k)^{\mathrm{T}}}{\underline{\mathbf{w}}^{k\mathrm{T}} \mathbf{M}_k \underline{\mathbf{w}}^k} \tag{18.85}$$

with $\underline{\mathbf{v}}^k = \underline{\mathbf{x}}^k - \underline{\mathbf{x}}^{k-1}$ and $\underline{\mathbf{w}}^k = \nabla f(\underline{\mathbf{x}}^k) - \nabla f(\underline{\mathbf{x}}^{k-1})$ $(k = 2, 3, \ldots)$. The step size α_k is obtained from

$$f(\underline{\mathbf{x}}^k - \alpha \mathbf{M}_k \nabla f(\underline{\mathbf{x}}^k)) = \min!\,, \quad \alpha \geq 0. \tag{18.86}$$

If $f(\underline{\mathbf{x}})$ is a quadratic function, then the DFP method becomes the conjugate gradient method with $\mathbf{M}_1 = \mathbf{I}$.

18.2.6 Evolution Strategies

18.2.6.1 Evolution Principles

Evolution strategies are examples of stochastic optimization processes imitating natural evolution. They are based on the principles of mutation, recombination and selection.

1. Mutation

From a parent point $\underline{\mathbf{x}}_P$ a offspring (descendant) $\underline{\mathbf{x}}_O = \underline{\mathbf{x}}_P + \underline{\mathbf{d}}$ is formed by applying a random variation $\underline{\mathbf{d}}$. The components of $\underline{\mathbf{d}}$ are $(0, \sigma_i^2)$ normally distributed random variables $Z(0, \sigma_i^2)$ determined newly at every mutation:

$$\underline{\mathbf{d}} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} Z(0,\sigma_1^2) \\ Z(0,\sigma_2^2) \\ \vdots \\ Z(0,\sigma_n^2) \end{pmatrix} = \begin{pmatrix} Z(0,1)\cdot\sigma_1 \\ Z(0,1)\cdot\sigma_2 \\ \vdots \\ Z(0,1)\cdot\sigma_n \end{pmatrix}. \tag{18.87}$$

With a normally distributed $\underline{\mathbf{d}}$ small changes have high probabilities while large changes occur very rarely. The changes are controlled by the standard deviation σ_i.

2. Recombination

From the population of μ parents offspring can be obtained by mixing the information from two or more parents, which are randomly selected. The recombination can follow two types of changes.
At *intermediate recombination* a offspring becomes as weighted average of ϱ randomly chosen parents:

$$\underline{\mathbf{x}}_O = \sum_{i=1}^{\varrho} \alpha_i \underline{\mathbf{x}}_{P_i}, \quad \sum_{i=1}^{\varrho} \alpha_i = 1, \quad 2 \leq \varrho \leq \mu\,. \tag{18.88}$$

At a *discrete* recombination of ϱ parents the i-th component of a offspring $\underline{\mathbf{x}}_O$ is determined by the i-th component of a randomly chosen parent:

$$x_{iO} = x_{iP_j}, \quad j \in \{1, \ldots, \varrho\}, \quad i = 1, \ldots, n. \tag{18.89}$$

3. Selection

By using mutation and recombination, a set of offspring is formed randomly. In a subsequent selection process the *objective function* $f(\underline{\mathbf{x}})$ serves as a measure to compare the fitness of the individuals. The fittest individuals are selected for the next generation. At certain strategies only the offspring take part in the selection. Other strategies consider also the parents (see also [18.12]).

18.2.6.2 Evolution Algorithms

Every evolution strategy is based on the following algorithm:

a) Determination of an appropriate starting population consisting of μ individuals. These are the first generation of parents. $X_P^1 = \{\underline{\mathbf{x}}_{P_1}^1, \ldots, \underline{\mathbf{x}}_{P_\mu}^1\}$.

b) In the k-th step the creation of λ offspring $X_O^k = \{\underline{\mathbf{x}}_{O_1}^k, \ldots, \underline{\mathbf{x}}_{O_\lambda}^k\}$ by mutation and recombination of parents of the actual generation $X_P^k = \{\underline{\mathbf{x}}_{P_1}^k, \ldots, \underline{\mathbf{x}}_{P_\lambda}^k\}$.

c) Application of selection to get the best μ individuals for the next parent generation $X_P^{k+1} = \{\underline{\mathbf{x}}_{P_1}^{k+1}, \ldots, \underline{\mathbf{x}}_{P_\mu}^{k+1}\}$.

d) Repeating steps b) and c) until stopping rule is satisfied. It can be fulfilling an optimal criterium of the optimization problem, or to reach a given number of generations, or exceeding a given computer time, etc.

18.2.6.3 Classification of Evolution Strategies

Every evolution strategy is characterized by a sequence of parameters. Essential parameters are the size of the population μ, the number of the offspring λ, the number of parents ϱ taking part in recombination and rules of making mutation, recombination and selection. To distinguish different types of strategies a special notation is commonly used. For the strategies using only mutation in producing offspring the $(\mu + \lambda)$, or (μ, λ) strategy notation is used. Strategies $(\mu + \lambda)$ and (μ, λ) differ from each other in the type of selection. At strategy (μ, λ) the selection of the new generation is made only among the offspring, while at strategy $(\mu + \lambda)$ the parents are also involved.
For strategies using recombination the number ϱ of the parents, which are involved, is seen in the notation $(\mu/\varrho + \lambda)$- and $(\mu/\varrho, \lambda)$-strategy.

18.2.6.4 Generating Random Numbers

For the numerical evaluation of evolution procedures *uniformly* and *normally* distributed random variables are needed. Values of uniformly distributed variables can be got by the methods given in subchapter 16.3.5.2, p. 843. Normally distributed random variables can be produced from uniform variables in the following way:
Box-Muller Method: If G_1 and G_2 are uniformly distributed random numbers in the interval $[0, 1]$, then the following two equations give two statistically independent normally distributed $(0, \sigma^2)$ random numbers $Z_1(0, \sigma^2)$ and $Z_2(0, \sigma^2)$:

$$Z_1(0, \sigma^2) = \sigma\sqrt{-2 \ln G_1}\cos(2\pi G_2) \quad \text{and} \quad Z_2(0, \sigma^2) = \sigma\sqrt{-2 \ln G_1}\sin(2\pi G_2)\,. \tag{18.90}$$

18.2.6.5 Application of Evolution Strategies

In the practice optimization problems have usually high complexity. Here the conventional optimization processes described in 18.2.5, p. 931 are often not appropriate. Evolution strategies belong to the differentiation-free solution methods, which are based on comparisons of the values of the objective function. They have simple conditions on the structure of the objective function. The objective function does not need to be differentiable or continuous. So the evolution strategies are appropriate for a wide spectrum of optimization problems.

The application of evolution strategies is not restricted to unconstrained continuous optimization problems. Optimization problems with constraints can also be handled, where the constraints are enforced by penalty terms in the objective function (see Penalty and Barrier Methods in 18.2.8, p. 940).
Another field of application is the discrete optimization, where some or all components of $\underline{\mathbf{x}}$ can take their values from a discrete set. One possible mutation mechanism is to replace the value of a discrete component by one of its neighboring values with the same probability.

18.2.6.6 $(1 + 1)$-Mutation-Selection Strategy

This method is similar to the gradient method discussed in 18.2.5, p. 931 with the difference that the direction $\underline{\mathbf{d}}^k$ is a normally distributed random vector. The population consists of a single individual which produces one offspring at every generation.

1. Mutation Step

In generation k a offspring is obtained from a parent by adding a normally distributed random vector:

$$\underline{\mathbf{x}}_O^k = \underline{\mathbf{x}}_P^k + \alpha\underline{\mathbf{d}}^k\,. \tag{18.91}$$

The factor α is a parameter by which the speed of the convergence can be affected. α is considered as the step size of the mutation.

2. Selection Step

The new parent of the next generation $(k+1)$ is selected by comparing the objective function values of both individuals, i.e., from the parent with the formula:

$$\underline{\mathbf{x}}_P^{k+1} = \begin{cases} \underline{\mathbf{x}}_O^k & \text{if } f(\underline{\mathbf{x}}_O^k) < f(\underline{\mathbf{x}}_P^k)\,, \\ \underline{\mathbf{x}}_P^k & \text{otherwise}\,. \end{cases} \tag{18.92}$$

The procedure stops if no better offspring arrives over a given number of generations. The step size α can be increased if the mutation results mostly in improved offspring. At small improvements the value of α should be decreased.

3. Step Size Control

The choice of the mutation step size α is of important influence to the convergence properties of the evolution method. While large step sizes are recommended in order to have fast convergence, small step size is required in the close neighborhood of the optimum or in regions of fast changing or oscillating of the objective function. The optimal step size depends on the problem. Too small steps lead to stagnation, too large steps may result in overshooting of the evolution process.

1. 1/5-Success rule: The ratio of the number of successful mutations and the total number of mutations in the last step defines the rate of success q. If $q > 1/5$, then the step size can be increased. For smaller success rate, α should be decreased:

$$\alpha_{k+1} = \begin{cases} c \cdot \alpha_k, & q < \dfrac{1}{5}\,, \\ 1 & \\ \dfrac{1}{c} \cdot \alpha_k, & q > \dfrac{1}{5} \end{cases} \qquad \text{with} \quad c = 0,8 \ldots 0,85. \tag{18.93}$$

2. Mutative Step Size Determination: The rule of 1/5 is a rough choice, and considering any concrete problem it is not always satisfactory. In an extended model the step size α and the standard deviations σ_i, $i = 1, 2 \ldots, n$ are in correlation. Here α and σ_i are multiplied with equal probability by one of the factors c, 1, $1/c$, where $c = 1, 1 \ldots 1, 5$. Further information see [18.12].

18.2.6.7 Population Strategies

The (1+1) strategy presented in the preceding paragraph reflects the principles of the natural evolution only in a very simplified form. With the extension to population models further properties of the evolution process can be considered. A large number of individuals in an evolution process assures that different regions of the solution space will be searched.

1. $(\mu + \lambda)$-Evolution Strategy

The $(\mu + \lambda)$ strategy is a generalization of the $(1 + 1)$ strategy. From the μ parents of the current generation $X_P^k = \{\underline{\mathbf{x}}_{P_1}^k, \ldots, \underline{\mathbf{x}}_{P_\mu}^k\}$ a set of λ parents is chosen randomly with equal probability. Repeated choices are allowed and even necessary in the case if $\mu < \lambda$. By mutation, λ offspring $X_O^k = \{\underline{\mathbf{x}}_{O_1}^k, \ldots, \underline{\mathbf{x}}_{O_\lambda}^k\}$ are produced. From the selection set $X_O^k \cup X_P^k$ the best μ individuals are chosen to take over into the next generation.

Since the parents are also taken in the selection, the quality of the population from a generation to the next one cannot be worse. The $(\mu + \lambda)$ strategy has the property that it keeps an already found local optimum, since large mutation steps, which are required to leave the optimal point, have very small probability. It means, that an individual can have an infinite life. This behavior can be avoided by adding penalty terms to the objective function values of parents that increase from generation to generation. In this way the aging of individuals can be simulated.

2. $(\mu\,, \lambda)$-Evolution Strategy

In contrary to the $(\mu + \lambda)$ strategy the selection step is now made among the λ offspring, to choose the μ individuals for the next generation, i.e., in this strategy the parents do not survive. Therefore $\lambda > \mu$

must hold. The values of the objective function for the offspring can be larger than that for the parents. This procedure can depart from local optima.

Selection Pressure: The ratio of the individuals taking part in the selection and the size of the population defines the selection pressure S:

$$S = \begin{cases} \dfrac{\mu+\lambda}{\mu} & \text{for } (\mu+\lambda)\text{-strategy}\,, \\ \dfrac{\lambda}{\mu} & \text{for } (\mu\,,\ \lambda)\text{-strategy} \end{cases} \qquad \text{with} \quad 1 \le S < \infty. \tag{18.94}$$

If the selection pressure is close to 1, then the selection step has almost no impact. A large number of offspring $\lambda \gg \mu$ results in a strong selection pressure, since from the set of the present individuals only few will survive into the next generation.

3. $(\mu/\varrho + \lambda)$- and $(\mu/\varrho\,,\ \lambda)$-Evolution Strategies with Recombination

With the concept of recombination some relations are built between the individuals of a population, so the information of several parents are mixed in a offspring.

In order to get a offspring, ϱ parents are chosen from the set of parents X_P^k with the same probability. It is assumed that for every member of the λ offspring a separate choice of ϱ parents is taken. The offspring is a discrete or an intermediate recombination of the chosen parents. The offspring produced in this way will be mutated and enters the selection process.

In the previously described $(\mu\ +\ \lambda)$ and (μ, λ) strategies each individual is the result of a series of mutations applied to one individual of the first generation of parents. So, a wider evolution step is possible only through many generations. Wider evolution steps are possible with recombination. Especially, when the parents have large distances among each other, the offspring are formed with completely new properties.

4. Evolution Strategies with more Populations

The principles of evolution can be expanded formally to the dimension of populations. Instead of individuals now populations are in competition. So, the evolution process has two steps. It is shown in the expanded notation: $[\mu_2/\varrho_2 \dagger\ \lambda_2(\mu_1/\varrho_1 \dagger\ \lambda_1)]$.

From a set of μ_2 parent populations a set of λ_2 offspring populations is created by recombination of ϱ_2 populations that are chosen randomly for each offspring population. In these λ_2 offspring populations the optimization is performed using a $(\mu_1/\varrho_1 + \lambda_1)$ or $(\mu_1/\varrho_1\ ,\ \lambda_1)$ strategy. After a given number of generations the best populations are chosen based on an appropriate criterium. The comparison of populations can be done by considering the values of the objective function of the best individual or by the population mean.

18.2.7 Gradient Method for Problems with Inequality Type Constraints

If the problem

$$f(\underline{\mathbf{x}}) = \min! \quad \text{subject to the constraints} \quad g_i(\underline{\mathbf{x}}) \le 0 \quad (i = 1, \ldots, m) \tag{18.95}$$

has to be solved by an iteration method of the type

$$\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k \quad (k = 1, 2, \ldots) \tag{18.96}$$

then two additional rules must be considered because of the bounded feasible set:

1. The direction $\underline{\mathbf{d}}^k$ must be a feasible descent direction at $\underline{\mathbf{x}}^k$.

2. The step size α_k must be determined so that $\underline{\mathbf{x}}^{k+1}$ is in M.

The different methods based on the formula (18.96) differ from each other only in the construction of the direction $\underline{\mathbf{d}}^k$. To ensure the feasibility of the sequence $\{\underline{\mathbf{x}}^k\} \subset M$, α_k' and α_k'' are determined in the following way:

$$\alpha_k' \quad \text{from} \quad f(\underline{\mathbf{x}}^k + \alpha \underline{\mathbf{d}}^k) = \min!\,, \quad \alpha \ge 0$$

$$\alpha_k'' = \max\{\alpha \in \mathbb{R} : \underline{\mathbf{x}}^k + \alpha \underline{\mathbf{d}}^k \in M\}. \tag{18.97}$$

Then

$$\alpha_k = \min\{\alpha_k', \alpha_k''\}. \tag{18.98}$$

If there is no feasible descent direction $\underline{\mathbf{d}}^k$ in a certain step k, then $\underline{\mathbf{x}}^k$ is a stationary point.

18.2.7.1 Method of Feasible Directions

1. Direction Search Program

A feasible descent direction $\underline{\mathbf{d}}^k$ at point $\underline{\mathbf{x}}^k$ can be determined by the solution of the following optimization problem:

$$\sigma = \min!, \tag{18.99}$$

$$\nabla g_i(\underline{\mathbf{x}}^k)^{\mathrm{T}} \underline{\mathbf{d}} \le \sigma, \ \ i \in I_0(\underline{\mathbf{x}}^k), \ \text{(18.100a)} \qquad \nabla f(\underline{\mathbf{x}}^k)^{\mathrm{T}} \underline{\mathbf{d}} \le \sigma, \text{(18.100b)} \qquad ||\underline{\mathbf{d}}|| \le 1. \tag{18.100c}$$

If $\sigma < 0$ for the result $\underline{\mathbf{d}} = \underline{\mathbf{d}}^k$ of this *direction search program*, then (18.100a) ensures feasibility and (18.100b) ensures the descending property of $\underline{\mathbf{d}}^k$. The feasible set for the direction search program is bounded by the normalizing condition (18.100c). If $\sigma = 0$, then $\underline{\mathbf{x}}^k$ is a stationary point, since there is no feasible descent direction at $\underline{\mathbf{x}}^k$.

A direction search program, defined by (18.100a,b,c), can result in a zig-zag behavior of the sequence $\underline{\mathbf{x}}^k$, which can be avoided if the index set $I_0(\underline{\mathbf{x}}^k)$ is replaced by the index set

$$I_{\varepsilon_k}(\underline{\mathbf{x}}^k) = \{i \in \{1, \ldots, m\} : \ -\varepsilon_k \le g_i(\underline{\mathbf{x}}^k) \le 0\}, \quad \varepsilon_k \ge 0 \tag{18.101}$$

which are the so-called ε_k active constraints in $\underline{\mathbf{x}}^k$. Thus, local directions of descent are excluded which are going from $\underline{\mathbf{x}}^k$ and lead closer to the boundaries of M consisting of the ε_k active constraints **(Fig. 18.7)**.

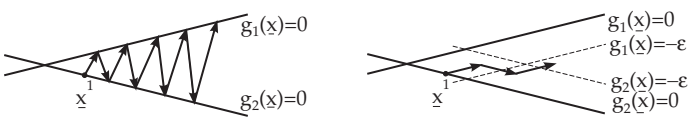

Figure 18.7

If $\sigma = 0$ is a solution of (18.100a,b,c) after these modifications, then $\underline{\mathbf{x}}^k$ is a stationary point only if $I_0(\underline{\mathbf{x}}^k) = I_{\varepsilon_k}(\underline{\mathbf{x}}^k)$. Otherwise ε_k must be decreased and the direction search program must be repeated.

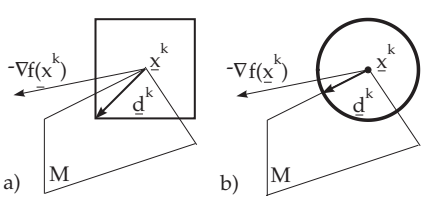

Figure 18.8

2. Special Case of Linear Constraints

If the functions $g_i(\underline{\mathbf{x}})$ are linear, i.e., $g_i(\underline{\mathbf{x}}) = \underline{\mathbf{a}}_i{}^{\mathrm{T}}\underline{\mathbf{x}} - b_i$, then a simpler direction search method can be established:

$$\sigma = \nabla f(\underline{\mathbf{x}}^k)^{\mathrm{T}} \underline{\mathbf{d}} = \min! \quad \text{with} \tag{18.102}$$

$$\underline{\mathbf{a}}_i{}^{\mathrm{T}} \underline{\mathbf{d}} \le 0, \quad i \in I_0(\underline{\mathbf{x}}^k) \ \text{ or } \ i \in I_{\varepsilon_k}(\underline{\mathbf{x}}^k), \text{(18.103a)}$$

$$||\underline{\mathbf{d}}|| \le 1. \tag{18.103b}$$

The effect of the choice of different norms $||\underline{\mathbf{d}}|| = \max\{|d_i|\} \le 1$ or $||\underline{\mathbf{d}}|| = \sqrt{\underline{\mathbf{d}}^{\mathrm{T}}\underline{\mathbf{d}}} \le 1$ is shown in **Fig. 18.8a,b**.

In a certain sense, the best choice is the norm $||\underline{\mathbf{d}}|| = ||\underline{\mathbf{d}}||_2 = \sqrt{\underline{\mathbf{d}}^{\mathrm{T}}\underline{\mathbf{d}}}$, since by the direction search program the direction $\underline{\mathbf{d}}^k$ is obtained, which forms the smallest angle with $-\nabla f(\underline{\mathbf{x}}^k)$. In this case the

direction search program is not linear and requires higher computational costs. With the choice $||\underline{\mathbf{d}}|| = ||\underline{\mathbf{d}}||_\infty = \max\{|d_i|\} \leq 1$ a system of linear constraints $-1 \leq d_i \leq 1$, $(i = 1, \ldots, n)$ is obtained, so the direction search program can be solved, e.g., by the simplex method.
In order to ensure that the method of feasible directions for a quadratic optimization problem $f(\underline{\mathbf{x}}) = \underline{\mathbf{x}}^{\mathrm{T}}\mathbf{C}\underline{\mathbf{x}} + \underline{\mathbf{p}}^{\mathrm{T}}\underline{\mathbf{x}} = \min!$ with $\mathbf{A}\underline{\mathbf{x}} \leq \underline{\mathbf{b}}$ results in a solution in finitely many steps, the direction search program is completed by the following conjugate requirements: If $\alpha_{k-1} = \alpha'_{k-1}$ holds in a step, i.e., $\underline{\mathbf{x}}^k$ is an "interior" point, then the condition

$$\underline{\mathbf{d}}^{k-1\mathrm{T}}\mathbf{C}\underline{\mathbf{d}} = 0 \tag{18.104}$$

is added to the direction search program. Furthermore the corresponding conditions are kept from the previous steps. If in a later step $\alpha_k = \alpha''_k$ then the condition (18.104) is removed.

■ $f(\underline{\mathbf{x}}) = x_1^2 + 4x_2^2 - 10x_1 - 32x_2 = \min!$ $g_1(\underline{\mathbf{x}}) = -x_1 \leq 0$, $g_2(\underline{\mathbf{x}}) = -x_2 \leq 0$, $g_3(\underline{\mathbf{x}}) = x_1 + 2x_2 - 7 \leq 0$, $g_4(\underline{\mathbf{x}}) = 2x_1 + x_2 - 8 \leq 0$.

Step 1: Starting with $\underline{\mathbf{x}}^1 = (3, 0)^{\mathrm{T}}$, $\nabla f(\underline{\mathbf{x}}^1) = (-4, -32)^{\mathrm{T}}$, $I_0(\underline{\mathbf{x}}^1) = \{2\}$.

Direction search program: $\left.\begin{array}{l} -4d_1 - 32d_2 = \min! \\ -d_2 \leq 0,\ ||\underline{\mathbf{d}}||_\infty \leq 1 \end{array}\right\} \Longrightarrow \underline{\mathbf{d}}^1 = (1, 1)^{\mathrm{T}}$.

Minimizing constant: $\alpha'_k = -\dfrac{\underline{\mathbf{d}}^{k\mathrm{T}}\nabla f(\underline{\mathbf{x}}^k)}{2\underline{\mathbf{d}}^{k\mathrm{T}}\mathbf{C}\underline{\mathbf{d}}^k}$ with $\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$.

Maximal feasible step size: $\alpha''_k = \min\left\{\dfrac{-g_i(\underline{\mathbf{x}}^k)}{\underline{\mathbf{a}}_i^{\mathrm{T}}\underline{\mathbf{d}}^k}\text{: for } i \text{ such that } \underline{\mathbf{a}}_i^{\mathrm{T}}\underline{\mathbf{d}}^k > 0\right\}$, $\alpha'_1 = \dfrac{18}{5}$, $\alpha''_1 = \dfrac{2}{3} \Longrightarrow$

$\alpha_1 = \min\left\{\dfrac{18}{5}, \dfrac{2}{3}\right\} = \dfrac{2}{3}$, $\underline{\mathbf{x}}^2 = \left(\dfrac{11}{3}, \dfrac{2}{3}\right)^{\mathrm{T}}$.

Step 2: $\nabla f(\underline{\mathbf{x}}^2) = \left(-\dfrac{8}{3}, -\dfrac{80}{3}\right)^{\mathrm{T}}$, $I_0(\underline{\mathbf{x}}^2) = \{4\}$.

Direction search program: $\left.\begin{array}{l} -\dfrac{8}{3}d_1 - \dfrac{80}{3}d_2 = \min! \\ 2d_1 + d_2 \leq 0,\ ||\underline{\mathbf{d}}||_\infty \leq 1 \end{array}\right\} \Longrightarrow \underline{\mathbf{d}}^2 = \left(-\dfrac{1}{2}, 1\right)^{\mathrm{T}}, \alpha'_2 = \dfrac{152}{51}, \alpha''_2 = \dfrac{4}{3} \Longrightarrow$

$\alpha_2 = \dfrac{4}{3}$, $\underline{\mathbf{x}}^3 = (3, 2)^{\mathrm{T}}$.

Step 3: $\nabla f(\underline{\mathbf{x}}^3) = (-4, -16)^{\mathrm{T}}$, $I_0 = (\underline{\mathbf{x}}^3) = \{3, 4\}$.
Direction search program:

$\left.\begin{array}{l} -4d_1 - 16d_2 = \min! \\ d_1 + 2d_2 \leq 0,\ 2d_1 + d_2 \leq 0,\ ||\underline{\mathbf{d}}||_\infty \leq 1 \end{array}\right\} \Longrightarrow \underline{\mathbf{d}}^3 = \left(-1, \dfrac{1}{2}\right)^{\mathrm{T}}$, $\alpha'_3 = 1$, $\alpha''_3 = 3 \Longrightarrow \alpha^3 = 1$, $\underline{\mathbf{x}}^4 = \left(2, \dfrac{5}{2}\right)^{\mathrm{T}}$.

The next direction search program results in $\sigma = 0$. Here the minimum point is $\underline{\mathbf{x}}^* = \underline{\mathbf{x}}^4$ **(Fig. 18.9)**.

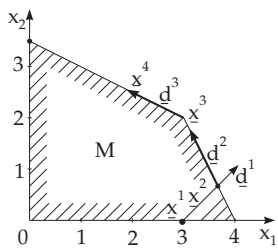

Figure 18.9

18.2.7.2 Gradient Projection Method

1. Formulation of the Problem and Solution Principle

Suppose the convex optimization problem

$$f(\underline{\mathbf{x}}) = \min! \quad \text{with} \quad \underline{\mathbf{a}}_i^{\mathrm{T}}\underline{\mathbf{x}} \leq b_i, \tag{18.105}$$

for $i = 1, \ldots, m$ is given. A feasible descent direction $\underline{\mathbf{d}}^k$ at the point $\underline{\mathbf{x}}^k \in M$ is determined in the following way:
If $-\nabla f(\underline{\mathbf{x}}^k)$ is a feasible direction, then $\underline{\mathbf{d}}^k = -\nabla f(\underline{\mathbf{x}}^k)$ is selected. Otherwise $\underline{\mathbf{x}}^k$ is on the boundary

of M and $-\nabla f(\underline{\mathbf{x}}^k)$ points outward from M. The vector $-\nabla f(\underline{\mathbf{x}}^k)$ is projected by a linear mapping $\mathbf{P}_k$ into a linear submanifold of the boundary of M defined by a subset of active constraints of $\underline{\mathbf{x}}^k$. **Fig. 18.10a** shows a projection into an edge, **Fig. 18.10b** shows a projection into a face. Supposing non-degeneracy, i.e., if the vectors $\underline{\mathbf{a}}_i$, $i \in I_0(\underline{\mathbf{x}})$ are linearly independent for every $\underline{\mathbf{x}} \in \mathbb{R}^n$, such a projection is given by

$$\underline{\mathbf{d}}^k = -\mathbf{P}_k \nabla f(\underline{\mathbf{x}}^k) = -\left(\mathbf{I} - \mathbf{A}_k{}^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k{}^{\mathrm{T}})^{-1}\mathbf{A}_k\right)\nabla f(\underline{\mathbf{x}}^k). \tag{18.106}$$

Here, $\mathbf{A}_k$ consists of all vectors $\underline{\mathbf{a}}_i{}^{\mathrm{T}}$, whose corresponding constraints form the sub-manifold, into which $-\nabla f(\underline{\mathbf{x}}_k)$ should be projected.

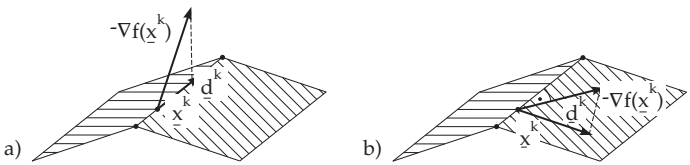

Figure 18.10

2. Algorithm

The gradient projection method consists of the following steps, starting with $\underline{\mathbf{x}}^1 \in M$ and substituting $k = 1$ and proceeding in accordance to the following scheme:

I: If $-\nabla f(\underline{\mathbf{x}}^k)$ is a feasible direction, then $\underline{\mathbf{d}}^k = -\nabla f(\underline{\mathbf{x}}^k)$ is substituted, and continued with **III**. Otherwise $\mathbf{A}_k$ is constructed from the vectors $\underline{\mathbf{a}}_i{}^{\mathrm{T}}$ with $i \in I_0(\underline{\mathbf{x}}^k)$ and continued with **II**.

II: $\underline{\mathbf{d}}^k = -\left(\mathbf{I} - \mathbf{A}_k{}^{\mathrm{T}}(\mathbf{A}_k\mathbf{A}_k{}^{\mathrm{T}})^{-1}\mathbf{A}_k\right)\nabla f(\underline{\mathbf{x}}^k)$ is sunstituted. If $\underline{\mathbf{d}}^k \neq \underline{\mathbf{0}}$, then continued with **III**. If $\underline{\mathbf{d}}^k = 0$ and $\underline{\mathbf{u}} = -(\mathbf{A}_k\mathbf{A}_k{}^{\mathrm{T}})^{-1}\mathbf{A}_k\nabla f(\underline{\mathbf{x}}^k) \geq 0$, then $\underline{\mathbf{x}}^k$ is a minimum point. The local Kuhn-Tucker conditions $-\nabla f(\underline{\mathbf{x}}^k) = \sum_{i \in I_0(\underline{\mathbf{x}}^k)} u_i \underline{\mathbf{a}}_i = \mathbf{A}_k{}^{\mathrm{T}}\underline{\mathbf{u}}$ are obviously satisfied.

If $\underline{\mathbf{u}} \not\geq \underline{\mathbf{0}}$, then an i with $u_i < 0$ is chosen, the i-th row from $\mathbf{A}_k$ is deleted and **II** is repeated.

III: Calculation of α_k and $\underline{\mathbf{x}}^{k+1} = \underline{\mathbf{x}}^k + \alpha_k \underline{\mathbf{d}}^k$ and returning to **I** with $k = k + 1$.

3. Remarks on the Algorithm

If $-\nabla f(\underline{\mathbf{x}}^k)$ is not feasible, then this vector is mapped onto the sub-manifold of the smallest dimension which contains $\underline{\mathbf{x}}^k$. If $\underline{\mathbf{d}}^k = \underline{\mathbf{0}}$, then $-\nabla f(\underline{\mathbf{x}}^k)$ is perpendicular to this sub-manifold. If $\underline{\mathbf{u}} \geq \underline{\mathbf{0}}$ does not hold, then the dimension of the sub-manifold is increased by one by omitting one of the active constraints, so maybe $\underline{\mathbf{d}}^k \neq \underline{\mathbf{0}}$ can occur **(Fig. 18.10b)** (with projection onto a (lateral) face). Since $\mathbf{A}_k$ is often obtained from $\mathbf{A}_{k-1}$ by adding or erasing a row, the calculation of $(\mathbf{A}_k\mathbf{A}_k{}^{\mathrm{T}})^{-1}$ can be simplified by the use of $(\mathbf{A}_{k-1}\mathbf{A}_{k-1}{}^{\mathrm{T}})^{-1}$.

■ Solution of the problem of the previous example on p. 938.

Step 1: $\underline{\mathbf{x}}^1 = (3, 0)^{\mathrm{T}}$,

I: $\nabla f(\underline{\mathbf{x}}^1) = (-4, -32)^{\mathrm{T}}$, $-\nabla f(\underline{\mathbf{x}}^1)$ is feasible, $\underline{\mathbf{d}}^1 = (4, 32)^{\mathrm{T}}$.

III: The step size is determined as in the previous example: $\alpha_1 = \dfrac{1}{20}$, $\underline{\mathbf{x}}^2 = \left(\dfrac{16}{5}, \dfrac{8}{5}\right)^{\mathrm{T}}$.

Step 2:

I: $\nabla f(\underline{\mathbf{x}}^2) = \left(-\dfrac{18}{5}, -\dfrac{96}{5}\right)^{\mathrm{T}}$ (not feasible), $I_0(\underline{\mathbf{x}}^2) = \{4\}$, $\mathbf{A}_2 = (2\ 1)$.

II: $\mathbf{P}_2 = \dfrac{1}{5}\begin{pmatrix} 1 & -2 \\ -2 & 4 \end{pmatrix}$, $\underline{\mathbf{d}}^2 = \left(-\dfrac{8}{25}, \dfrac{16}{25}\right)^{\mathrm{T}} \neq \underline{\mathbf{0}}$.

III: $\alpha_2 = \frac{5}{8}$, $\underline{\mathbf{x}}^3 = (3,2)^{\mathrm{T}}$.

Step 3:

I: $\nabla f(\underline{\mathbf{x}}^3) = (-4,-16)^{\mathrm{T}}$ (not feasible), $I_0(\underline{\mathbf{x}}^3) = \{3,4\}$, $\mathbf{A}_3 = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$.

II: $\mathbf{P}_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\underline{\mathbf{d}}^3 = (0,0)^{\mathrm{T}}, \underline{\mathbf{u}} = \left(\frac{28}{3}, -\frac{8}{3}\right)^{\mathrm{T}}$ $u_2 < 0$: $\mathbf{A}_3 = (1\ 2)$.

II: $\mathbf{P}_3 = \frac{1}{5}\begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix}$, $\underline{\mathbf{d}}^3 = \left(-\frac{16}{5}, \frac{8}{5}\right)^{\mathrm{T}}$.

III: $\alpha_3 = \frac{5}{16}$, $\underline{\mathbf{x}}^4 = \left(2, \frac{5}{2}\right)^{\mathrm{T}}$.

Step 4:

I: $\nabla f(\underline{\mathbf{x}}^4) = (-6,-12)^{\mathrm{T}}$ (not feasible), $I_0(\underline{\mathbf{x}}^4) = \{3\}$, $\mathbf{A}_4 = \mathbf{A}_3$.

II: $\mathbf{P}_4 = \mathbf{P}_3$, $\underline{\mathbf{d}}^4 = (0,0)^{\mathrm{T}}$, $u = 6 \geq 0$.

It follows that $\underline{\mathbf{x}}^4$ is a minimum point.

18.2.8 Penalty Function and Barrier Methods

The basic principle of these methods is that a constrained optimization problem is transformed into a sequence of optimization problems without constraints by modifying the objective function. The modified problem can be solved, e.g., by the methods given in Section 18.2.5. With an appropriate construction of the modified objective functions, every accumulation point of the sequence of the solution points of this modified problem is a solution of the original problem.

18.2.8.1 Penalty Function Method

The problem

$$f(\underline{\mathbf{x}}) = \min! \text{ subject to } g_i(\underline{\mathbf{x}}) \leq 0 \ \ (i = 1,2,\ldots,m) \tag{18.107}$$

is replaced by the sequence of unconstrained problems

$$H(\underline{\mathbf{x}}, p_k) = f(\underline{\mathbf{x}}) + p_k S(\underline{\mathbf{x}}) = \min! \text{ with } \underline{\mathbf{x}} \in \mathbf{R}^n, \ \ p_k > 0 \ \ (k = 1,2,\ldots). \tag{18.108}$$

Here, p_k is a positive parameter, and for $S(\underline{\mathbf{x}})$

$$S(\underline{\mathbf{x}}) = \begin{cases} = 0 & \underline{\mathbf{x}} \in M, \\ > 0 & \underline{\mathbf{x}} \notin M, \end{cases} \tag{18.109}$$

holds, i.e., leaving the feasible set M is punished with a "penalty" term $p_k S(\underline{\mathbf{x}})$. The problem (18.108) is solved with a sequence of penalty parameters p_k tending to ∞. Then

$$\lim_{k\to\infty} H(\underline{\mathbf{x}}, p_k) = f(\underline{\mathbf{x}}), \quad \underline{\mathbf{x}} \in M. \tag{18.110}$$

If $\underline{\mathbf{x}}^k$ is the solution of the k-th penalty problem, then:

$$H(\underline{\mathbf{x}}^k, p_k) \geq H(\underline{\mathbf{x}}^{k-1}, p_{k-1}), \quad f(\underline{\mathbf{x}}^k) \geq f(\underline{\mathbf{x}}^{k-1}), \tag{18.111}$$

and every accumulation point $\underline{\mathbf{x}}^*$ of the sequence $\{\underline{\mathbf{x}}^k\}$ is a solution of (18.107). If $\underline{\mathbf{x}}^k \in M$, then $\underline{\mathbf{x}}^k$ is a solution of the original problem.

For instance, the following functions are appropriate realizations of $S(\underline{\mathbf{x}})$:

$$S(\underline{\mathbf{x}}) = \max{}^r\{0, g_1(\underline{\mathbf{x}}), \ldots, g_m(\underline{\mathbf{x}})\} \ \ (r = 1,2,\ldots) \text{ or} \tag{18.112a}$$

$$S(\underline{\mathbf{x}}) = \sum_{i=1}^{m} \max{}^r\{0, g_i(\underline{\mathbf{x}})\} \ \ (r = 1,2,\ldots). \tag{18.112b}$$

If functions $f(\underline{\mathbf{x}})$ and $g_i(\underline{\mathbf{x}})$ are differentiable, then in the case $r > 1$, the penalty function $H(\underline{\mathbf{x}}, p_k)$ is also differentiable on the boundary of M, so analytic solutions can be used to solve the auxiliary problem (18.108).

Fig. 18.11 shows a representation of the penalty function method.

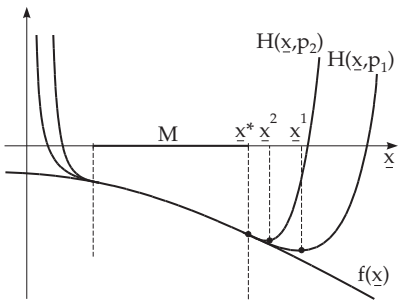

Figure 18.11

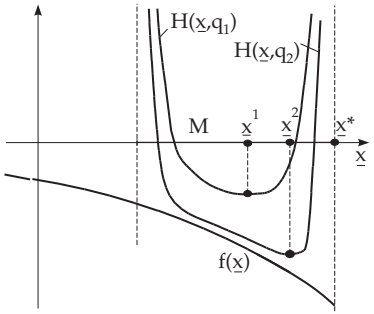

Figure 18.12

■ $f(\underline{\mathbf{x}}) = x_1^2 + x_2^2 = \text{ min! for } x_1 + x_2 \geq 1,\ H(\underline{\mathbf{x}}, p_k) = x_1^2 + x_2^2 + p_k \max^2\{0, 1 - x_1 - x_2\}$.
The necessary optimality condition is:
$\nabla H(\underline{\mathbf{x}}, p_k) = \begin{pmatrix} 2x_1 - 2p_k \max\{0, 1 - x_1 - x_2\} \\ 2x_2 - 2p_k \max\{0, 1 - x_1 - x_2\} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.
The gradient of H is evaluated here with respect to $\underline{\mathbf{x}}$. By subtracting the equations we have $x_1 = x_2$. The equation $2x_1 - 2p_k \max\{0, 1 - 2x_1\} = 0$ has a unique solution $x_1^k = x_2^k = \dfrac{p_k}{1 + 2p_k}$. We get the solution $x_1^* = x_2^* = \lim\limits_{k \to \infty} \dfrac{p_k}{1 + 2p_k} = \dfrac{1}{2}$ by letting $k \to \infty$.

18.2.8.2 Barrier Method

A sequence of modified problems is considered in the form

$$H(\underline{\mathbf{x}}, q_k) = f(\underline{\mathbf{x}}) + q_k B(\underline{\mathbf{x}}) = \text{min!}\,, \quad q_k > 0\,. \tag{18.113}$$

The term $q_k B(\underline{\mathbf{x}})$ prevents the solution leaving the feasible set M, since the objective function increases unboundedly on approaching the boundary of M. The *regularity condition*

$$M^0 = \{\underline{\mathbf{x}} \in M \,:\, g_i(\underline{\mathbf{x}}) < 0 \ \ (i = 1, 2, \ldots, m)\} \neq \emptyset \quad \text{and} \quad \overline{M^0} = M \tag{18.114}$$

must be satisfied, i.e., the interior of M must be non-empty and it is possible to get to any boundary point by approaching it from the interior, i.e., the closure of M^0 is M.
The function $B(\underline{\mathbf{x}})$ is defined to be continuous on M^0. It increases to ∞ at the boundary of M. The modified problem (18.113) is solved by a sequence of barrier parameters q_k tending to zero. For the solution $\underline{\mathbf{x}}^k$ of the k-th problem (18.113)

$$f(\underline{\mathbf{x}}^k) \leq f(\underline{\mathbf{x}}^{k-1}), \tag{18.115}$$

holds and every accumulation point $\underline{\mathbf{x}}^*$ of the sequence $\{\underline{\mathbf{x}}^k\}$ is a solution of (18.107).
Fig. 18.12 shows a representation of the barrier method.
The functions, e.g.,

$$B(\underline{\mathbf{x}}) = -\sum_{i=1}^{m} -\ln(-g_i(\underline{\mathbf{x}})), \quad \underline{\mathbf{x}} \in M^0 \quad \text{or} \tag{18.116a}$$

$$B(\underline{\mathbf{x}}) = \sum_{i=1}^{m} \frac{1}{[-g_i(\underline{\mathbf{x}})]^r} \quad (r = 1, 2, \ldots), \quad \underline{\mathbf{x}} \in M^0 \tag{18.116b}$$

are appropriate realizations of $B(\underline{\mathbf{x}})$.

■ $f(\underline{\mathbf{x}}) = x_1^2 + x_2^2 = \min!$ subject to $x_1 + x_2 \geq 1$, $H(\underline{\mathbf{x}}, q_k) = x_1^2 + x_2^2 + q_k(-\ln(x_1 + x_2 - 1))$, $x_1 + x_2 > 1$, $\nabla H(\underline{\mathbf{x}}, q_k) = \begin{pmatrix} 2x_1 - q_k \dfrac{1}{x_1 - x_2 - 1} \\ 2x_2 - q_k \dfrac{1}{x_1 + x_2 - 1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $x_1 + x_2 > 1$. The gradient of H is given with respect to $\underline{\mathbf{x}}$.

Subtracting the equations results in $x_1 = x_2$, $2x_1 - q_k \dfrac{1}{2x_1 - 1} = 0$, $x_1 > \dfrac{1}{2}$, $\Longrightarrow$ $x_1^2 - \dfrac{x_1}{2} - \dfrac{q_k}{4} = 0$, $x_1 > \dfrac{1}{2}$, $x_1^k = x_2^k = \dfrac{1}{4} + \sqrt{\dfrac{1}{16} + \dfrac{1}{4} q_k}$, $k \to \infty$, $q_k \to 0$: $x_1^* = x_2^* = \dfrac{1}{2}$.

The solutions of problems (18.108) and (18.113) at the k-th step do not depend on the solutions of the previous steps. The application of higher penalty or smaller barrier parameters often leads to convergence problems with numerical solutions of (18.108) and (18.113), e.g., in the method of (18.2.4), in particular, if no good initial approximation is available. Using the result of the k-th problem as the initial solution for the $(k+1)$-th problem the convergence behavior can be improved.

18.2.9 Cutting Plane Methods

1. Formulation of the Problem and Principle of Solution

Let consider the problem

$$f(\underline{\mathbf{x}}) = \underline{\mathbf{c}}^{\mathrm{T}} \underline{\mathbf{x}} = \min!, \quad \underline{\mathbf{c}} \in \mathbb{R}^n \tag{18.117}$$

over the bounded region $M \subset \mathbb{R}^n$ given by convex functions $g_i(\underline{\mathbf{x}})$ $(i = 1, 2, \ldots, m)$ in the form $g_i(\underline{\mathbf{x}}) \leq 0$. A problem with a non-linear but convex objective function $f(\underline{\mathbf{x}})$ is transformed into this form, if

$$f(\underline{\mathbf{x}}) - x_{n+1} \leq 0, \quad x_{n+1} \in \mathbb{R} \tag{18.118}$$

is considered as a further constraint and

$$\overline{f}(\overline{\underline{\mathbf{x}}}) = x_{n+1} = \min! \quad \text{for all} \quad \overline{\underline{\mathbf{x}}} = (\underline{\mathbf{x}}, x_{n+1}) \in \mathbb{R}^{n+1} \tag{18.119}$$

is solved with $\overline{g}_i(\underline{\mathbf{x}}) = g_i(\underline{\mathbf{x}}) \leq 0$.

The basic idea of the method is the iterative linear approximation of M by a convex polyhedron in the neighborhood of the minimum point $\underline{\mathbf{x}}^*$, and therefore the original program is reduced to a sequence of linear programming problems.

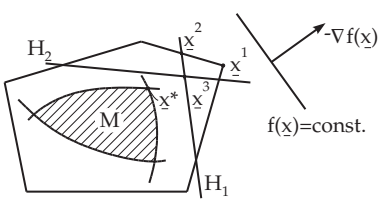

Figure 18.13

First, a polyhedron is determined:

$$P_1 = \{\underline{\mathbf{x}} \in \mathbb{R}^n \ : \ \underline{\mathbf{a}}_i{}^{\mathrm{T}} \underline{\mathbf{x}} \leq b_i, \ \ i = 1, \ldots, s\}. \tag{18.120}$$

From the linear program

$$f(\underline{\mathbf{x}}) = \min! \quad \text{with} \quad \underline{\mathbf{x}} \in P_1 \tag{18.121}$$

an optimal extreme point $\underline{\mathbf{x}}^1$ of P_1 is determined with respect to $f(\underline{\mathbf{x}})$. If $\underline{\mathbf{x}}^1 \in M$ holds, then the optimal solution of the original problem is found. Otherwise, a hyperplane is determined:

$H_1 = \{\underline{\mathbf{x}} : \ \underline{\mathbf{a}}_{s+1}{}^{\mathrm{T}} \underline{\mathbf{x}} = b_{s+1}, \ \underline{\mathbf{a}}_{s+1}{}^{\mathrm{T}} x^1 > b_{s+1}\}$, which separates the point $\underline{\mathbf{x}}^1$ from M, so the new polyhedron contains

$$P_2 = \{\underline{\mathbf{x}} \in P_1 : \ \underline{\mathbf{a}}_{s+1}{}^{\mathrm{T}} \underline{\mathbf{x}} \leq b_{s+1}\}. \tag{18.122}$$

Fig. 18.13 shows a schematic representation of the cutting plane method.

2. Kelley Method

The different methods differ from each other in the choice of the separating planes H_k. In the case of the Kelley method H_k is chosen in the following way: A j_k is chosen so that

$$g_{j_k}(\underline{\mathbf{x}}^k) = \max\{g_i(\underline{\mathbf{x}}^k) \ \ (i = 1, \ldots, m)\}. \tag{18.123}$$

At the point $\underline{\mathbf{x}} = \underline{\mathbf{x}}^k$, the function $g_{jk}(\underline{\mathbf{x}})$ has the tangent plane

$$T(\underline{\mathbf{x}}) = g_{j_k}(\underline{\mathbf{x}}^k) + (\underline{\mathbf{x}} - \underline{\mathbf{x}}^k)^{\mathrm{T}} \nabla g_{j_k}(\underline{\mathbf{x}}^k). \tag{18.124}$$

The hyperplane $H_k = \{\underline{\mathbf{x}} \in \mathbb{R}^n \,:\, T(\underline{\mathbf{x}}) = 0\}$ separates the point $\underline{\mathbf{x}}^k$ from all points $\underline{\mathbf{x}}$ with $g_{j_k}(\underline{\mathbf{x}}) \leq 0$. So, for the $(k+1)$-th linear program, $T(\underline{\mathbf{x}}) \leq 0$ is added as a further constraint. Every accumulation point $\underline{\mathbf{x}}^*$ of the sequence $\{\underline{\mathbf{x}}^k\}$ is a minimum point of the original problem.
In practical applications this method shows a low speed of convergence. Furthermore, the number of constraints is always increasing.

18.3 Discrete Dynamic Programming

18.3.1 Discrete Dynamic Decision Models

A wide class of optimization problems can be solved by the methods of dynamic programming. The problem is considered as a *process* proceeding naturally or formally in time, and it is controlled by time-dependent decisions. If the process can be decomposed into a finite or countably infinite number of steps, then it is called *discrete dynamic programming*, otherwise it is a *continuous dynamic programming*. In this section, only n-stage discrete processes are discussed.

18.3.1.1 n-Stage Decision Processes

An n-stage process P starts at stage 0 with an initial state $\underline{\mathbf{x}}_a = \underline{\mathbf{x}}_0$ and proceeds through the intermediate states $\underline{\mathbf{x}}_1, \underline{\mathbf{x}}_2, \ldots, \underline{\mathbf{x}}_{n-1}$ into a final state $\underline{\mathbf{x}}_n = \underline{\mathbf{x}}_e \in X_e \subseteq \mathbb{R}^m$. The *state vectors* $\underline{\mathbf{x}}_j$ are in the state space $X_j \subseteq \mathbb{R}^m$. To drive a state $\underline{\mathbf{x}}_{j-1}$ into the state $\underline{\mathbf{x}}_j$, a *decision* $\underline{\mathbf{u}}_j$ is required. All possible *decision vectors* $\underline{\mathbf{u}}_j$ in the state $\underline{\mathbf{x}}_{j-1}$ form the *decision space* $U_j(\underline{\mathbf{x}}_{j-1}) \subseteq \mathbb{R}^s$. From $\underline{\mathbf{x}}_{j-1}$ the consecutive state $\underline{\mathbf{x}}_j$ can be obtained by the transformation **(Fig. 18.14)**

$$\underline{\mathbf{x}}_j = g_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \quad j = 1(1)n. \tag{18.125}$$

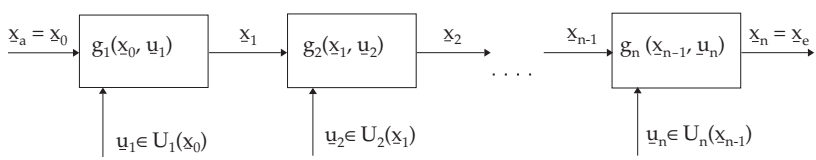

Figure 18.14

18.3.1.2 Dynamic Programming Problem

Our goal is to determine a *policy* $(\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_n)$ which drives the process from the initial state $\underline{\mathbf{x}}_a$ into the state $\underline{\mathbf{x}}_e$ considering all constraints so that it minimizes an objective function or *cost function* $f(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n))$. The functions $f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j)$ are called *stage costs*. The standard form of the dynamic programming problem is

$$\textbf{OF:} \quad f(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) \longrightarrow \min! \tag{18.126a}$$

$$\textbf{CT:} \quad \left.\begin{array}{ll} \underline{\mathbf{x}}_j = g_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), & j = 1(1)n, \\ \underline{\mathbf{x}}_0 = \underline{\mathbf{x}}_a,\ \underline{\mathbf{x}}_n = \underline{\mathbf{x}}_e \in X_e,\ \underline{\mathbf{x}}_j \in X_j \subseteq \mathbb{R}^m, & j = 1(1)n, \\ \underline{\mathbf{u}}_j \in U_j(\underline{\mathbf{x}}_{j-1}) \subseteq \mathbb{R}^m, & j = 1(1)n. \end{array}\right\} \tag{18.126b}$$

The first type of constraints $\underline{\mathbf{x}}_j$ are called *dynamic* and the others $\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_j$ are called *static*. Similarly to (18.126a), a maximum problem can also be considered. A policy $(\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_n)$ satisfying all constraints is called *feasible*. The methods of dynamic programming can be applied if the objective function satisfies certain additional requirements (see 18.3.3, p. 944).

18.3.2 Examples of Discrete Decision Models

18.3.2.1 Purchasing Problem

In the j-th period of a time interval which can be divided into n periods, a workshop needs v_j units of a certain primary material. The available amount of this material at the beginning of period j is denoted by x_{j-1}, in particular, $x_0 = x_a$ is given. The amounts u_j are to be determined for a unit price c_j, which should be purchased by the workshop at the end of each period. The given storage capacity K must not be exceeded, i.e., $x_{j-1} + u_j \leq K$. A purchase policy $(u_1, \ldots, u_n)$ should be determined, which minimizes the total cost. This problem leads to the following dynamic programming problem:

$$\textbf{OF:} \quad f(u_1, \ldots, u_n) = \sum_{j=1}^{n} f_j(u_j) = \sum_{j=1}^{n} c_j u_j \longrightarrow \min! \tag{18.127a}$$

$$\textbf{CT:} \quad \left.\begin{array}{ll} x_j = x_{j-1} + u_j - v_j\,, & j = 1(1)n, \\ x_0 = x_a\,, \quad 0 \leq x_j \leq K\,, & j = 1(1)n, \\ U_j(x_{j-1}) = \{u_j : \max\{0, v_j - x_{j-1}\} \leq u_j \leq K - x_{j-1}\}\,, & j = 1(1)n. \end{array}\right\} \tag{18.127b}$$

In (18.127b) it is ensured that demands are satisfied and the storage capacity is not exceeded. If there is also a storage cost l per unit per period, then intermediate cost in the j-th period is $(x_{j-1} + u_j - v_j/2)l$, and the modified cost function is

$$f(x_0, u_1, \ldots, x_{n-1}, u_n) = \sum_{j=1}^{n} (c_j u_j + (x_{j-1} + u_j - v_j/2) \cdot l). \tag{18.128}$$

18.3.2.2 Knapsack Problem

One has to select some of the items $A_1, \ldots, A_n$ with weights $w_1, \ldots, w_n$ and with values $c_1, \ldots, c_n$ so that the total weight does not exceed a given bound W, and the total value is maximal. This problem does not depend on time. It will be reformulated in the following way: At every stage a decision u_j about the selection of item A_j is made. Here, $u_j = 1$ holds if A_j is selected, otherwise $u_j = 0$. The capacity still available at the beginning of a stage is denoted by x_{j-1}, so the following dynamic problem arises:

$$\textbf{OF:} \quad f(u_1, \ldots, u_n) = \sum_{j=1}^{n} c_j u_j \longrightarrow \max! \tag{18.129a}$$

$$\textbf{CT:} \quad \left.\begin{array}{ll} x_j = x_{j-1} - w_j u_j, & j = 1(1)n, \\ x_0 = W\,, \quad 0 \leq x_j \leq W, & j = 1(1)n, \\ \left.\begin{array}{ll} u_j \in \{0, 1\}, & \text{falls } x_{j-1} \geq w_j, \\ u_j = 0\,, & \text{falls } x_{j-1} < w_j, \end{array}\right\} & j = 1(1)n. \end{array}\right\} \tag{18.129b}$$

18.3.3 Bellman Functional Equations

18.3.3.1 Properties of the Cost Function

In order to state the Bellman functional equations, the cost function must satisfy two requirements:

1. Separability

The function $f(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n))$ is called *separable*, if it can be given by binary functions $H_1, \ldots, H_{n-1}$ and by functions $F_1, \ldots, F_n$ in the following way:

$$\begin{array}{l} f(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) = F_1(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)), \\ F_1(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) = H_1\left(f_1(\underline{\mathbf{x}}_0, \underline{\mathbf{u}}_1), F_2(f_2(\underline{\mathbf{x}}_1, \underline{\mathbf{u}}_2), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n))\right), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ F_{n-1}(f_{n-1}(\underline{\mathbf{x}}_{n-2}, \underline{\mathbf{u}}_{n-1}), f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) = H_{n-1}\left(f_{n-1}(\underline{\mathbf{x}}_{n-2}, \underline{\mathbf{u}}_{n-1}), F_n(f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n))\right), \\ F_n(f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) = f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n). \end{array} \tag{18.130}$$

2. Minimum Interchangeability

A function $H(\tilde{f}(\underline{\mathbf{a}}), \tilde{F}(\underline{\mathbf{b}}))$ is called *minimum interchangeable*, if:

$$\min_{(\underline{\mathbf{a}},\underline{\mathbf{b}})\in A\times B} H\left(\tilde{f}(\underline{\mathbf{a}}), \tilde{F}(\underline{\mathbf{b}})\right) = \min_{\underline{\mathbf{a}}\in A} H\left(\tilde{f}(\underline{\mathbf{a}}), \min_{\underline{\mathbf{b}}\in B} \tilde{F}(\underline{\mathbf{b}})\right). \tag{18.131}$$

This property is satisfied, for example, if H is monotone increasing with respect to its second argument for every $\underline{\mathbf{a}} \in A$, i.e., if for every $\underline{\mathbf{a}} \in A$,

$$H\left(\tilde{f}(\underline{\mathbf{a}}), \tilde{F}(\underline{\mathbf{b}}_1)\right) \le H\left(\tilde{f}(\underline{\mathbf{a}}), \tilde{F}(\underline{\mathbf{b}}_2)\right) \quad \text{for} \quad \tilde{F}(\underline{\mathbf{b}}_1) \le \tilde{F}(\underline{\mathbf{b}}_2). \tag{18.132}$$

Now, for the cost function of the dynamic programming problem, the separability of f and the minimum interchangeability of all functions H_j, $j = 1(1)n-1$ are required. The following often occurring type of cost function satisfies both requirements:

$$f^{sum} = \sum_{j=1}^{n} f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j) \quad \text{or} \quad f^{max} = \max_{j=1(1)n} f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j). \tag{18.133}$$

The functions H_j are

$$H_j^{sum} = f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j) + \sum_{k=j+1}^{n} f_k(\underline{\mathbf{x}}_{k-1}, \underline{\mathbf{u}}_k) \quad \text{and} \tag{18.134}$$

$$H_j^{max} = \max\left\{ f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \max_{k=j+1(1)n} f_k(\underline{\mathbf{x}}_{k-1}, \underline{\mathbf{u}}_k) \right\}. \tag{18.135}$$

18.3.3.2 Formulation of the Functional Equations

The following functions are defined:

$$\phi_j(\underline{\mathbf{x}}_{j-1}) = \min_{\substack{\underline{\mathbf{u}}_k\in U_k(\underline{\mathbf{x}}_{k-1}) \\ k=j(1)n}} F_j(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)), \quad j = 1(1)n, \tag{18.136}$$

$$\phi_{n+1}(\underline{\mathbf{x}}_n) = 0. \tag{18.137}$$

If there is no policy $(\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_n)$ driving the state $\underline{\mathbf{x}}_{j-1}$ into a final state $\underline{\mathbf{x}}_e \in X_e$, then we substitute $\phi_j(\underline{\mathbf{x}}_{j-1}) = \infty$. Using the separability and minimum interchangeability conditions and the dynamic constraints for $j = 1(1)n$ we get:

$$\begin{aligned}
\phi_j(\underline{\mathbf{x}}_{j-1}) &= \min_{\underline{\mathbf{u}}_j\in U_j(\underline{\mathbf{x}}_{j-1})} H_j(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \min_{\substack{\underline{\mathbf{u}}_k\in U_k(\underline{\mathbf{x}}_{k-1}) \\ k=j+1(1)n}} F_{j+1}(f_{j+1}(\underline{\mathbf{x}}_j, \underline{\mathbf{u}}_{j+1}), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n))), \\
&= \min_{\underline{\mathbf{u}}_j\in U_j(\underline{\mathbf{x}}_{j-1})} H_j\left(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \phi_{j+1}(\underline{\mathbf{x}}_j)\right) \\
\phi_j(\underline{\mathbf{x}}_{j-1}) &= \min_{\underline{\mathbf{u}}_j\in U_j(\underline{\mathbf{x}}_{j-1})} H_j\left(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \phi_{j+1}(g_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j))\right).
\end{aligned} \tag{18.138}$$

Equations (18.138) and (18.137) are called the *Bellman functional equations.* $\phi_1(\underline{\mathbf{x}}_0)$ is the optimal value of the cost function f.

18.3.4 Bellman Optimality Principle

The evaluation of the functional equation

$$\phi_j(\underline{\mathbf{x}}_{j-1}) = \min_{\underline{\mathbf{u}}_j\in U_j(\underline{\mathbf{x}}_{j-1})} H_j\left(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \phi_{j+1}(\underline{\mathbf{x}}_j)\right) \tag{18.139}$$

corresponds to the determination of an optimal policy $(\underline{\mathbf{u}}_j^*, \ldots, \underline{\mathbf{u}}_n^*)$ for which the subprocess P_j starting at state $\underline{\mathbf{x}}_{j-1}$ and consisting of the last $n-j+1$ stages of the total process P minimizes the cost function, i.e.,

$$F_j(f_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j), \ldots, f_n(\underline{\mathbf{x}}_{n-1}, \underline{\mathbf{u}}_n)) \longrightarrow \min! \tag{18.140}$$

The optimal policy of the process P_j with the initial state $\underline{\mathbf{x}}_{j-1}$ is independent of the decisions $\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_{j-1}$ of the first $j-1$ stages of P which have driven P to the state $\underline{\mathbf{x}}_{j-1}$. To determine $\phi_j(\underline{\mathbf{x}}_{j-1})$ the value $\phi_{j+1}(\underline{\mathbf{x}}_j)$ is needed to know. Now, if $(\underline{\mathbf{u}}_j^*, \ldots, \underline{\mathbf{u}}_n^*)$ is an optimal policy for P_j, then, obviously, $(\underline{\mathbf{u}}_{j+1}^*, \ldots, \underline{\mathbf{u}}_n^*)$ is an optimal policy for the subprocess P_{j+1} starting at $\underline{\mathbf{x}}_j = g_j(\underline{\mathbf{x}}_{j-1}, \underline{\mathbf{u}}_j^*)$. This statement is generalized in the *Bellman optimality principle*.

Bellman Principle: If $(\underline{\mathbf{u}}_1^*, \ldots, \underline{\mathbf{u}}_n^*)$ is an optimal policy of the process P and $(\underline{\mathbf{x}}_0^*, \ldots, \underline{\mathbf{x}}_n^*)$ is the corresponding sequence of states, then for every subprocess $P_j, j = 1(1)n$, with initial state $\underline{\mathbf{x}}_{j-1}^*$ the policy $(\underline{\mathbf{u}}_j^*, \ldots, \underline{\mathbf{u}}_n^*)$ is also optimal.

18.3.5 Bellman Functional Equation Method

18.3.5.1 Determination of Minimal Costs

With the functional equations (18.137), (18.138) and starting with $\phi_{n+1}(\underline{\mathbf{x}}_n) = 0$ every value $\phi_j(\underline{\mathbf{x}}_{j-1})$ with $\underline{\mathbf{x}}_{j-1} \in X_{j-1}$ is determined in decreasing order of j. It requires the solution of an optimum problem over the decision space $U_j(\underline{\mathbf{x}}_{j-1})$ for every $\underline{\mathbf{x}}_{j-1} \in X_{j-1}$. For every $\underline{\mathbf{x}}_{j-1}$ there is a minimum point $\underline{\mathbf{u}}_j \in U_j$ as an optimal decision for the first stage of a subprocess P_j starting at $\underline{\mathbf{x}}_{j-1}$. If the sets X_j are not finite or they are too large, then the values ϕ_j can be calculated for a set of selected nodes $\underline{\mathbf{x}}_{j-1} \in X_{j-1}$. The intermediate values can be calculated by a certain interpolation method. $\phi_1(\underline{\mathbf{x}}_0)$ is the optimal value of the cost function of process P. The optimal policy $(\underline{\mathbf{u}}_1^*, \ldots, \underline{\mathbf{u}}_n^*)$ and the corresponding states $(\underline{\mathbf{x}}_0^*, \ldots, \underline{\mathbf{x}}_n^*)$ can be determined by one of the following two methods.

18.3.5.2 Determination of the Optimal Policy

1. Variant 1: During the evaluation of the functional equations, the computed $\underline{\mathbf{u}}_j$ is also saved for every $\underline{\mathbf{x}}_{j-1} \in X_{j-1}$. After the calculation of $\phi_1(\underline{\mathbf{x}}_0)$, an optimal policy is got if $\underline{\mathbf{x}}_1^* = g_1(\underline{\mathbf{x}}_0^*, \underline{\mathbf{u}}_1^*)$ is determined from $\underline{\mathbf{x}}_0 = \underline{\mathbf{x}}_0^*$ and the saved $\underline{\mathbf{u}}_1 = \underline{\mathbf{u}}_1^*$. From $\underline{\mathbf{x}}_1^*$ and the saved decision $\underline{\mathbf{u}}_2^*$ follows $\underline{\mathbf{x}}_2^*$, etc.

2. Variant 2: For every $\underline{\mathbf{x}}_{j-1} \in X_{j-1}$ only $\phi_j(\underline{\mathbf{x}}_{j-1})$ is saved. After every $\phi_j(\underline{\mathbf{x}}_{j-1})$ is known, a forward calculation is made. Starting with $j = 1$ and $\underline{\mathbf{x}}_0 = \underline{\mathbf{x}}_0^*$ one determines $\underline{\mathbf{u}}_j^*$ in increasing order of j by the evaluation of the functional equation

$$\phi_j(\underline{\mathbf{x}}_{j-1}^*) = \min_{\underline{\mathbf{u}}_j \in U_j(\underline{\mathbf{x}}_{j-1}^*)} H_j\left(f_j(\underline{\mathbf{x}}_{j-1}^*, \underline{\mathbf{u}}_j), \phi_{j+1}(g_j(\underline{\mathbf{x}}_{j-1}^*, \underline{\mathbf{u}}_j))\right). \tag{18.141}$$

$\underline{\mathbf{x}}_j^* = g_j(\underline{\mathbf{x}}_{j-1}^*, \underline{\mathbf{u}}_j^*)$ is obtained. During the forward calculation, an optimization problem must be solved again at every stage.

3. Comparison of the two Variants: The computation costs of variant **1** are less than variant **2** requires because of the forward calculations. However decision $\underline{\mathbf{u}}_j$ is saved for every state $\underline{\mathbf{x}}_{j-1}$, which may require very large memory in the case of a higher dimensional decision space $U_j(\underline{\mathbf{x}}_{j-1})$, while in the case of variant **2**, only the values $\phi_j(\underline{\mathbf{x}}_{j-1})$ must be saved. Therefore, sometimes variant **2** is used on computers.

18.3.6 Examples for Applications of the Functional Equation Method

18.3.6.1 Optimal Purchasing Policy

1. Formulation of the Problem

The problem from 18.3.2.1, p. 944, to determine an optimal purchasing policy

OF $f(u_1,\ldots,u_n)=\sum_{j=1}^{n} c_j u_j \longrightarrow \min!$ (18.142a)

CT
$$\begin{aligned} &x_j = x_{j-1}+u_j-v_j, \quad j=1(1)n,\\ &x_0 = x_a,\ 0\le x_j\le K, \quad j=1(1)n,\\ &U_j(x_{j-1}) = \{u_j \ :\ \max\{0, v_j - x_{j-1}\} \le u_j \le K - x_{j-1}\}, \quad j=1(1)n \end{aligned} \tag{18.142b}$$

leads to the functional equations

$$\phi_{n+1}(x_n) = 0, \tag{18.143}$$

$$\phi_j(x_{j-1}) = \min_{u_j\in U_j(x_{j-1})} (c_j u_j + \phi_{j+1}(x_{j-1}+u_j-v_j)), \quad j=1(1)n. \tag{18.144}$$

2. Numerical Example

$n=6\,,\quad K=10\,,\quad x_a=2\,.\qquad \begin{matrix} c_1=4, & c_2=3, & c_3=5, & c_4=3, & c_5=4, & c_6=2,\\ v_1=6, & v_2=7, & v_3=4, & v_4=2, & v_5=4, & v_6=3. \end{matrix}$

Backward Calculation: The function values $\phi_j(x_{j-1})$ will be determined for the states $x_{j-1} = 0,1,\ldots,10$. Now, it is enough to make the minimum search only for integer values of u_j.

$j=6$: $\quad \phi_6(x_5) = \min_{u_6\in U_6(x_5)} c_6 u_6 = c_6 \max\{0, v_6 - x_5\} = 2\max\{0, 3-x_5\}$.

According to variant **2** of the Bellman functional equation method, only the values of $\phi_6(x_5)$ are written in the last row. For example, $\phi_4(0)$ is determined.

$$\begin{aligned} \phi_4(0) &= \min_{2\le u_4\le 10}(3u_4+\phi_5(u_4-2))\\ &= \min(28,27,26,25,24,25,26,27,30) = 24. \end{aligned}$$

	x_j=0	1	2	3	4	5	6	7	8	9	10
j=1			75								
2	59	56	53	50	47	44	41	38	35	32	29
3	44	39	34	29	24	21	18	15	12	9	6
4	24	21	18	15	12	9	6	4	2	0	0
5	22	18	14	10	6	4	2	0	0	0	0
6	6	4	2	0	0	0	0	0	0	0	0

Forward Calculation:

$$\phi_1(2) = 75 = \min_{4\le u_1\le 8}(4u_1+\phi_2(u_1-4)).$$

One gets $u_1^*=4$ as the minimum point, therefore $x_1^* = x_0^* + u_1^* - v_1 = 0$. This method is repeated for $\phi_2(0)$ and for all later stages. The optimal policy is:

$$(u_1^*, u_2^*, u_3^*, u_4^*, u_5^*, u_6^*) = (4,10,1,6,0,3).$$

18.3.6.2 Knapsack Problem

1. Formulation of the Problem

Consider the problem given in 18.3.2.2, p. 944

OF $: f(u_1,\ldots,u_n) = \sum_{j=1}^{n} c_j u_j \longrightarrow \max!$ (18.145a)

CT:
$$\left.\begin{aligned} &x_j = x_{j-1} - w_j u_j, && j=1(1)n,\\ &x_0 = W, \quad 0\le x_j\le W, && j=1(1)n,\\ &u_j\in\{0,1\}, \ \text{if } x_{j-1}\ge w_j,\\ &u_j = 0\,, \quad \text{if } x_{j-1} < w_j, \end{aligned}\right\} j=1(1)n. \tag{18.145b}$$

Since this is a maximum problem, the Bellman functional equations are now

$$\phi_{n+1}(x_n) = 0,$$

$$\phi_j(x_{j-1}) = \max_{u_j \in U_j(x_{j-1})} (c_j u_j + \phi_{j+1}(x_{j-1} - w_j u_j)), \quad j = 1(1)n.$$

The decisions can be only 0 and 1, so it is practical to apply variant **1** of the functional equation method. For $j = n, n-1, \ldots, 1$:

$$\phi_j(x_{j-1}) = \begin{cases} c_j + \phi_{j+1}(x_{j-1} - w_j) & \text{if } x_{j-1} \geq w_j \text{ and } c_j + \phi_{j+1}(x_{j-1} - w_j) > \phi_{j+1}(x_{j-1}), \\ \phi_{j+1}(x_{j-1}) & \text{otherwise,} \end{cases}$$

$$u_j(x_{j-1}) = \begin{cases} 1 \text{ if } x_{j-1} \geq w_j \text{ and } c_j + \phi_{j+1}(x_{j-1} - w_j) > \phi_{j+1}(x_{j-1}), \\ 0 \text{ otherwise.} \end{cases}$$

2. Numerical Example

$W = 10, \quad n = 6.$ $\quad c_1 = 1, \quad c_2 = 2, \quad c_3 = 3, \quad c_4 = 1, \quad c_5 = 5, \quad c_6 = 4,$
$\quad w_1 = 2, \quad w_2 = 4, \quad w_3 = 6, \quad w_4 = 3, \quad w_5 = 7, \quad w_6 = 6.$

Since the weights w_j are integers, the possible values for x_j are $x_j \in \{0, 1, \ldots, 10\}$, $j = 1(1)n$, and $x_0 = 10$. The table contains the function values $\phi_j(x_{j-1})$ and the actual decision $u_j(x_{j-1})$ for every stage and for every state x_{j-1}. For example, the values of $\phi_6(x_5)$, $\phi_3(2)$, $\phi_3(6)$, and $\phi_3(8)$ are calculated:

$$\phi_6(x_5) = \begin{cases} 0, & \text{if } x_5 < w_6 = 4, \\ c_6 = 6, & \text{otherwise,} \end{cases} \qquad u_6(x_5) = \begin{cases} 0, \text{ if } x_5 < 4, \\ 0, \text{ otherwise.} \end{cases}$$

$\phi_3(2)$: $\quad x_2 = 2 < w_3 = 3$: $\quad \phi_3(2) = \phi_4(2) = 3, \ u_3(2) = 0.$

$\phi_3(6)$: $\quad x_2 > w_3$ and $c_3 + \phi_3(x_2 - w_3) = 6 + 3 < \phi_4(x_2) = 10$: $\quad \phi_3(6) = 10, \ u_3(6) = 0.$

$\phi_3(8)$: $\quad x_2 > w_3$ and $c_3 + \phi_3(x_2 - w_3) = 6 + 9 > \phi_4(x_2) = 10$: $\quad \phi_3(8) = 15, \ u_3(8) = 1.$

The optimal policy is

$(u_1^*, u_2^*, u_3^*, u_4^*, u_5^*, u_6^*) = (0, 1, 1, 1, 0, 1), \quad \phi_1(10) = 19.$

	$x_j = 0$	1	2	3	4	5	6	7	8	9	10
$j = 1$											19; 0
2	0; 0	3; 0	4; 1	7; 1	9; 0	10; 1	13; 1	13; 1	15; 0	16; 0	19; 1
3	0; 0	3; 0	3; 0	6; 1	9; 1	9; 0	10; 0	12; 1	15; 1	16; 1	16; 0
4	0; 0	3; 1	3; 1	3; 1	6; 0	9; 1	10; 1	10; 1	10; 1	13; 0	16; 1
5	0; 0	0; 0	0; 0	0; 0	6; 0	7; 1	7; 1	7; 1	7; 1	13; 1	13; 1
6	0; 0	0; 0	0; 0	0; 0	6; 1	6; 1	6; 1	6; 1	6; 1	6; 1	6; 1