apl. Prof. Dr. Frank Werner
Fakultät für Mathematik
Institut für Mathematische Optimierung
https://math.uni-magdeburg.de/∼werner/

Lecture Script

(in extracts)

O P E R A T I O N S R E S E A R C H

iMBA Magdeburg

Summer Term 2020

(Date: May 21, 2020)

Literature

1. Werner, F.; Sotskov, Y.N.: Mathematics of Economics and Business, 1st Edition, Routledge, Abingdon (UK) / New York (USA), 2006, 536 p.
   (**FREE** Download of the ebook e.g. under Amazon.de Kindle Store, siehe
   https://www.amazon.de/Mathematics-Economics-Business-English-Werner-ebook/dp/B000Q7ZFKW/)

2. Hillier, F; Lieberman, G: Introduction to Operations Research, 9th Edition, MacGraw Hill, New York (USA), 2010.

3. Taha, H.A.: Operations Research - An Introduction, 7th Edition, Prentice Hall, New York (USA), 2003.

4. **for a refreshment of mathematical preliminaries:**

   Werner, F.: A Refresher Course in Mathematics, Bookboon Publishers, 2016, 284 S.

   (**FREE** Download of the pdf file under:

   https://bookboon.com/en/a-refresher-course-in-mathematics-ebook)

Contents

1 Linear Programming

1.1 Introductory Example

Example 1 *A company produces a mixture consisting of three raw materials denoted as R_1, R_2 and R_3. Raw materials R_1 and R_2 must be contained with a given minimum percentage, and raw material R_3 must not exceed a certain given maximum percentage. Moreover, the price of each raw material per kilogram is known. The data are summarized in Table 1.*

Table 1: *Data for Example 1*

Raw material	*Required percentage*	*Price in EUR per kilogram*
R_1	*at least 10 per cent*	*25*
R_2	*at least 50 per cent*	*17*
R_3	*at most 30 per cent*	*12*

We wish to determine a feasible mixture with the lowest cost. Let $x_i, i \in \{1,2,3\}$, be the percentage of raw material R_i. Then we get the following constraints:

$$x_1 + x_2 + x_3 = 100. \tag{1}$$

Equation (1) states that the sum of the percentages of all raw materials equals 100 per cent. Since the percentage of raw material R_3 should not exceed 30 per cent, we obtain the constraint

$$x_3 \leq 30. \tag{2}$$

The percentage of raw material R_2 is at least 50 per cent, or what is the same, the sum of the percentages of R_1 and R_3 is no more than 50 per cent:

$$x_1 + x_3 \leq 50. \tag{3}$$

Moreover, the percentage of R_1 is at least 10 per cent, or what is the same, the sum of the percentages of R_2 and R_3 should not exceed 90 per cent, i.e.

$$x_2 + x_3 \leq 90. \tag{4}$$

Moreover, all variables should be nonnegative:

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0. \tag{5}$$

The cost of producing the resulting mixture should be minimized, i.e. the objective function is as follows:

$$z = 25x_1 + 17x_2 + 12x_3 \longrightarrow \text{min!} \tag{6}$$

The notation $z \longrightarrow \text{min!}$ indicates that the value of function z should become minimal for the desired solution. So we have formulated a problem consisting of an objective function (6), four constraints (three inequalities (2),(3) and (4) and one equation (1)) and the non-negativity constraints (5) for all three variables.

1.2 Preliminaries

In general, a linear programming problem (abbreviated by LPP) consists of constraints (a system of linear equations or linear inequalities), non-negativity constraints and a linear objective function. The general form of such an LPP can be given as follows.

General form of an LPP:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \longrightarrow \begin{cases} \text{max!} \\ \text{min!} \end{cases}$$

subject to (s.t.)

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ \{\leq, =, \geq\} \ b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ \{\leq, =, \geq\} \ b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ \{\leq, =, \geq\} \ b_m \end{array}$$

$$x_j \geq 0 \, , j \in J \subseteq \{1, 2, \ldots, n\}$$

Alternatively, we can give the following *matrix representation* of an LPP:

$$z = \mathbf{c}^T\mathbf{x} \longrightarrow \begin{cases} \text{max!} \\ \text{min!} \end{cases}$$

s.t.

$$A\mathbf{x}\{\leq, =, \geq\}\mathbf{b} \qquad (7)$$

$$\mathbf{x} \geq \mathbf{0}.$$

Here matrix A is of order $m \times n$. The vector $\mathbf{c} = (c_1, c_2, \ldots, c_n)^T$ is denoted as vector of the coefficients in the objective function, and the vector $\mathbf{b} = (b_1, b_2, \ldots, b_m)^T$ is denoted as the right-hand side vector.

Geometrical interpretation of an LPP with two variables x_1 and x_2

Assume that the constraints are given as inequalities. The constraints $a_{i1}x_1 + a_{i2}x_2 \, R_i \, b_i$, $R_i \in \{\leq, \geq\}$, $i = 1, 2, \ldots m$, are half-planes which are bounded by the lines $a_{i1}x_1 + a_{i2}x_2 = b_i$. Each of these lines can also be written in the form

$$\frac{x_1}{s_1} + \frac{x_2}{s_2} = 1,$$

where $s_1 = b_i/a_{i1}$ and $s_2 = b_i/a_{i2}$ are the intercepts of the line with the x_1 - and x_2- coordinate axes.

For fixed z and $c_2 \neq 0$, the objective function $z = c_1 x_1 + c_2 x_2$ is a line of the form

$$x_2 = -\frac{c_1}{c_2} x_1 + \frac{z}{c_2},$$

i.e. for different values of z we get parallel lines all with slope $-c_1/c_2$. The vector

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

gives the direction in which the objective function increases most. Thus, when maximizing the linear objective function z, we have to shift the line

$$x_2 = -\frac{c_1}{c_2} x_1 + \frac{z}{c_2}$$

into the direction given by vector $\mathbf{c}$, while when minimizing z, we have to shift this line into the opposite direction given by vector $-\mathbf{c}$.

An LPP of the form (7) with two variables can be *graphically* solved as follows:

(1) Determine the feasible region M (i.e. the set of feasible solutions) as the intersection of all feasible half-planes with the first quadrant.

(2) Draw the objective function $z = Z$, where Z is constant and shift it either into the direction given by vector $\mathbf{c}$ (in the case of $z \to$ max!) or into the direction given by vector $-\mathbf{c}$ (in the case of $z \to$ min!) Apply this procedure as long as the line $z = const$ has joint points with the feasible region.

Definition 1 *A feasible solution* $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$*, for which the objective function has an optimum (i.e. maximum or minimum) value is called* **optimal solution.**

1.3 Properties of Linear Programming Problems

Definition 2 *A set* M *is called* **convex***, if for any two vectors* $\mathbf{x^1}, \mathbf{x^2} \in M$*, any convex combination* $\lambda \mathbf{x^1} + (1-\lambda)\mathbf{x^2}$ *with* $0 \leq \lambda \leq 1$ *also belongs to set* M*.*

Definition 3 *A vector (point)* $\mathbf{x} \in M$ *is called an* **extreme point** *of the convex set* M*, if* $\mathbf{x}$ *cannot be written as a proper convex combination of two other vectors of* M*, i.e.* $\mathbf{x}$ *cannot be written as* $\lambda \mathbf{x^1} + (1-\lambda)\mathbf{x^2}$ *with* $\mathbf{x^1}, \mathbf{x^2} \in M$ *and* $0 < \lambda < 1$*.*

Thus, when considering a system of m inequalities with two nonnegative variables, the set of solutions is described by the intersection of m half-planes with the nonnegative quadrant.

Theorem 1 The set M of feasible solutions of system (7) is either empty or a convex set with at most a finite number of extreme points.

Theorem 2 If the set M of feasible solutions of system (7) is bounded, it can be written as the set of all convex combinations of the extreme points $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^s}$ of set M, i.e.:

$$\begin{aligned} M \;=\; \Big\{ & \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \lambda_1 \mathbf{x^1} + \lambda_2 \mathbf{x^2} + \cdots + \lambda_s \mathbf{x^s}; \\ & 0 \le \lambda_i \le 1, \quad i = 1, 2, \ldots, s, \quad \sum_{i=1}^{s} \lambda_i = 1 \Big\} \end{aligned}$$

Let M be the feasible region and consider the maximization of objective function $z = \mathbf{c}^T \mathbf{x}$. There may occur the following three cases:

(a) We have $M = \emptyset$. In this case the constraints are inconsistent, i.e. there does not exist a feasible solution of the LPP.

(b) M is a nonempty bounded subset of the n-space $\mathbb{R}^n$.

(c) M is an unbounded subset of the n-space $\mathbb{R}^n$, i.e. at least one variable may become arbitrarily large, or if some variables are not necessarily nonnegative, at least one of them may become arbitrarily small.

In case (b), set M is called a convex polyhedron, and there always exists a solution of the maximization problem. In case (c), there are again two possibilities:

(c1) The objective function z is bounded from above. Then an optimal solution of the maximization problem under consideration exists.

(c2) The objective function z is not bounded from above. Then there does not exist a (finite) optimal solution for the maximization problem under consideration.

Theorem 3 If an LPP has a (finite) optimal solution, then there exists at least one extreme point, where the objective function has an optimum value.

Theorem 4 Let $P_1, P_2, \ldots, P_r$ described by vectors $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^r}$ be optimal extreme points. Then every convex combination

$$\mathbf{x^0} = \lambda_1 \mathbf{x^1} + \lambda_2 \mathbf{x^2} + \ldots + \lambda_r \mathbf{x^r}, \quad \lambda_i \geq 0, \ i = 1, 2, \ldots, r, \quad \sum_{i=1}^{r} \lambda_i = 1$$

is also an optimal solution.

1.4 Standard Form of a Linear Programming Problem

Let $r(A)$ be the rank of matrix A, i.e., the maximum number of linearly independent column (or equivalently, row) vectors of matrix A.

Definition 4 *A system $A\mathbf{x} = \mathbf{b}$ of $p = r(A)$ linear equations, where in each equation one variable occurs only in this equation and it has the coefficient +1, is called system of linear equations in* **canonical form**. *These eliminated variables are called the* **basic variables** *(bv), while the remaining variables are called the* **nonbasic variables** *(nbv).*

Hence the number of basic variables of a system of linear equations in canonical form is equal to the rank of matrix A. As a consequence of Definition 4, if a system of linear equations $A\mathbf{x} = \mathbf{b}$ is given in canonical form, the coefficient matrix A always contains an identity matrix. If $r(A) = p = n$, the identity matrix I is of order $n \times n$, i.e the system has the form

$$I\mathbf{x^B} = \mathbf{b},$$

where $\mathbf{x^B}$ is the vector of the basic variables (note that columns may have been interchanged in matrix A to get the identity matrix which means that the order of the variables in vector $\mathbf{x^B}$ is different from that in vector $\mathbf{x}$). If $r(A) = p < n$, the order of the identity submatrix is $p \times p$. In the latter case, the system can be written as

$$I\mathbf{x^B} + A_N \mathbf{x^N} = \mathbf{b},$$

where $\mathbf{x^B}$ is the p-vector of the basic variables, $\mathbf{x^N}$ is the $(n-p)$-vector of the nonbasic variables and A_N is the submatrix of A formed by the column vectors belonging to the nonbasic variables (again column interchanges in matrix A may have been applied).

Definition 5 *A solution* $\mathbf{x}$ *of a system of equations* $A\mathbf{x} = \mathbf{b}$ *in canonical form, where each nonbasic variable has the value zero, is called a* **basic solution**.

Definition 6 *An LPP of the form*

$$z = \mathbf{c}^T\mathbf{x} \longrightarrow \max!$$

$$s.t. \qquad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

where $A = (A_N, I)$ *and* $\mathbf{b} \geq \mathbf{0}$*, is called the* **standard form** *of an LPP.*

According to Definition 6, matrix A can be partitioned into some matrix A_N and an identity submatrix I. Thus, the standard form of an LPP is characterized by the following properties:

- the LPP is a maximization problem;
- the constraints are given as a system of linear equations in canonical form with nonnegative right-hand sides and
- all variables have to be nonnegative.

Any LPP can formally be transformed into the standard form by the following rules. We consider the possible violations of the standard form according to Definition 6.

(a) Some variable x_j is not necessarily nonnegative, i.e. x_j may take arbitrary values. Then variable x_j is replaced by the difference of two nonnegative variables, i.e. we set:

$$x_j = x_j^* - x_j^{**} \qquad \text{with} \quad x_j^* \geq 0 \quad \text{and} \quad x_j^{**} \geq 0.$$

Then we get:

$$\begin{aligned} x_j^* > x_j^{**} &\iff x_j > 0 \\ x_j^* = x_j^{**} &\iff x_j = 0 \\ x_j^* < x_j^{**} &\iff x_j < 0. \end{aligned}$$

(b) The given objective function has to be minimized:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \rightarrow \min!$$

The determination of a minimum of function z is equivalent to the determination of a maximum of function $\bar{z} = -z$:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \rightarrow \min! \iff \bar{z} = -z = -c_1x_1 - c_2x_2 - \cdots - c_nx_n \rightarrow \max!$$

(c) For some right-hand side, we have $b_i < 0$:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i < 0.$$

In this case, we multiply this constraint by -1 and obtain:

$$-a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n = -b_i > 0.$$

(d) Let some constraints be inequalities:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

or

$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \geq b_k.$$

Then by introducing a slack variable u_i and a surplus variable u_k, respectively, we obtain an equation:

$$\begin{array}{lll} a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + u_i = b_i & \text{with} & u_i \geq 0 \\ \text{or} & & \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n - u_k = b_k & \text{with} & u_k \geq 0\ . \end{array}$$

(e) Let the given system of linear equations be not in canonical form, i.e. the constraints are given e.g. as follows:

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

with $\quad b_i \geq 0,\ i = 1, 2, \ldots, m;\ x_j \geq 0,\ j = 1, 2, \ldots, n.$

In the above situation, there is no constraint that contains an eliminated variable with coefficient +1 (provided that the column vectors of matrix A belonging to variables $x_1, x_2, \ldots, x_n$ are different from the unit vector). Then we introduce in each equation an artificial variable x_{Ai} as basic variable and obtain:

$$\begin{array}{lllll} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{A1} & & & = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & + x_{A2} & & = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & & & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & & + x_{Am} & = b_m \end{array}$$

with $\quad b_i \geq 0,\ i = 1, 2, \ldots, m;\ x_j \geq 0,\ j = 1, 2, \ldots, n$, and $x_{Ai} \geq 0, i = 1, 2, \ldots, m.$

Example 2 *Given is the following LPP:*

$$
\begin{array}{rrcrcrcrcr}
 & & & z = -x_1 + 3x_2 + x_4 \to \min! & & & & & & \\
s.t. & x_1 & - & x_2 & + & 3x_3 & - & x_4 & \geq & 8 \\
 & & & x_2 & - & 5x_3 & + & 2x_4 & \leq & -4 \\
 & & & & & x_3 & + & x_4 & \leq & 3 \\
 & & & & & x_2, x_3, x_4 \geq 0. & & & &
\end{array}
$$

First, we substitute variable x_1 by the difference of two nonnegative variables x_1^* and x_1^{**}, i.e. $x_1 = x_1^* - x_1^{**}$ with $x_1^* \geq 0, x_2^{**} \geq 0$. Further, we multiply the objective function z by -1 and obtain:

$$
\begin{array}{rrcrcrcrcrcr}
 & & & \overline{z} = -z = x_1^* - x_1^{**} - 3x_2 - x_4 \to \max! & & & & & & & & \\
s.t. & x_1^* & - & x_1^{**} & - & x_2 & + & 3x_3 & - & x_4 & \geq & 8 \\
 & & & & & x_2 & - & 5x_3 & + & 2x_4 & \leq & -4 \\
 & & & & & & & x_3 & + & x_4 & \leq & 3 \\
 & & & & & x_1^*, x_1^{**}, x_2, x_3, x_4 \geq 0. & & & & & &
\end{array}
$$

Multiplying the second constraint by -1 and introducing the slack variable x_7 in the third constraint as well as the surplus variables x_5 and x_6 in the first and second constraints, we obtain all constraints as equations with nonnegative right-hand sides:

$$
\begin{array}{rrcrcrcrcrcrcrcrcr}
 & & & \overline{z} = -z = x_1^* - x_2^{**} - 3x_2 - x_4 \to \max! & & & & & & & & & & & & & & \\
s.t. & x_1^* & - & x_1^{**} & - & x_2 & + & 3x_3 & - & x_4 & - & x_5 & & & & & = & 8 \\
 & & & & - & x_2 & + & 5x_3 & - & 2x_4 & & & - & x_6 & & & = & 4 \\
 & & & & & & & x_3 & + & x_4 & & & & & + & x_7 & = & 3 \\
 & & & & & x_1^*, x_1^{**}, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0. & & & & & & & & & & & &
\end{array}
$$

Now we can choose variable x_1^* as eliminated variable in the first constraint and variable x_7 as the eliminated variable in the third constraint, but there is no variable that occurs only in the second constraint having coefficient +1. Therefore, we introduce the artificial variable x_{A1} in the second constraint and obtain:

$$
\begin{array}{rrcrcrcrcrcrcrcrcrcr}
 & & & \overline{z} = -z = x_1^* - x_2^{**} - 3x_2 - x_4 \to \max! & & & & & & & & & & & & & & & & \\
s.t. & -x_1^{**} & - & x_2 & + & 3x_3 & - & x_4 & - & x_5 & & & + & \mathbf{x_1^*} & & & & & = & 8 \\
 & & - & x_2 & + & 5x_3 & - & 2x_4 & & & - & x_6 & & & + & \mathbf{x_{A1}} & & & = & 4 \\
 & & & & & x_3 & + & x_4 & & & & & & & & & + & \mathbf{x_7} & = & 3 \\
 & & & & & x_1^*, x_1^{**}, x_2, x_3, x_4, x_5, x_6, x_7, x_{A1} \geq 0. & & & & & & & & & & & & & &
\end{array}
$$

Notice that we have written the variables in such a way that the identity submatrix (column vectors of variables x_1^*, x_{A1}, x_7) occurs at the end. So in the standard form, the problem has now $n = 9$ variables. A vector satisfying all constraints is only a feasible solution for the original problem if the artificial variable x_{A1} has value zero (otherwise the original second constraint would be violated).

1.5 The Simplex Algorithm

Basic idea:

Starting with some initial extreme point (represented by a basic feasible solution resulting from the standard form of an LPP), we compute the value of the objective function and check whether the latter can be improved upon by moving to an adjacent extreme point (by applying the pivoting procedure). If so, we perform this move to the next extreme point and seek then whether further improvement is possible by a subsequent move. When finally an extreme point is attained that does not admit any further improvement, it will constitute an optimal solution.

In order to apply such an approach, a criterion to decide whether a move to an adjacent extreme point improves the objective function value is required which we will derive in the following. In the following, we assume that the rank of matrix A equals $m : r(A) = m$, i.e., in the canonical form there are m basic variables among the n variables, and the number of nonbasic variables equals $n' = n - m$. Consider a *feasible canonical form* with the basic variables x_{Bi} and the nonbasic variables x_{Nj}:

$$x_{Bi} = \hat{b}_i - \sum_{j=1}^{n'} \hat{a}_{ij} x_{Nj} \, , \quad i = 1, 2, \ldots, m \quad (n' = n - m). \tag{8}$$

Then the objective function z can be written as follows:

$$\begin{aligned} z &= c_1x_1 + c_2x_2 + \ldots + c_nx_n \\ &= \underbrace{c_{B1}x_{B1} + c_{B2}x_{B2} + \ldots + c_{Bm}x_{Bm}}_{(basic \quad variables)} + \underbrace{c_{N1}x_{N1} + c_{N2}x_{N2} + \ldots + c_{Nn'}x_{Nn'}}_{(nonbasic \quad variables)} \\ &= \sum_{i=1}^{m} c_{Bi}x_{Bi} + \sum_{j=1}^{n'} c_{Nj}x_{Nj}. \end{aligned}$$

Using equations (8), we can replace the basic variables and write the objective function only in dependence on the nonbasic variables. We obtain

$$\begin{aligned} z &= \sum_{i=1}^{m} c_{Bi} \left(\hat{b}_i - \sum_{j=1}^{n'} \hat{a}_{ij} x_{Nj} \right) + \sum_{j=1}^{n'} c_{Nj} x_{Nj} \\ &= \sum_{i=1}^{m} c_{Bi} \hat{b}_i - \sum_{j=1}^{n'} \left(\sum_{i=1}^{m} c_{Bi} \hat{a}_{ij} - c_{Nj} \right) x_{Nj}. \end{aligned}$$

We denote the latter row, where the objective function is written in terms of the current nonbasic variables, as the *objective row.* Moreover, we define the following values:

$$z_0 = \sum_{i=1}^{m} c_{Bi} \hat{b}_i \quad \text{(value of the objective function of the basic solution);} \tag{9}$$

$$g_j = \sum_{i=1}^{m} c_{Bi}\hat{a}_{ij} - c_{Nj} \qquad \text{(coefficient of variable } x_{Nj} \text{ in the objective row).} \qquad (10)$$

Concerning the calculation of value z_0 according to formula (9), we remind that in a basic solution, all nonbasic variables are equal to zero.

Then we get the following representation of the objective function in dependence on the nonbasic variables x_{Nj}:

$$z = z_0 - g_1 x_{N1} - g_2 x_{N2} - \ldots - g_{n'} x_{Nn'}.$$

Here each coefficient g_j gives the change in the objective function value if the nonbasic variable x_{Nj} is included into the set of basic variables (replacing some other basic variable) and if its value would increase by one unit. By means of the coefficients in the objective row we can give the following optimality criterion.

Theorem 5 (Optimality or simplex criterion)
If we have $g_j \geq 0,\ j = 1, 2, \ldots, n'$, for all coefficients of the nonbasic variables in the objective row, the corresponding solution is optimal.

Corollary 1 *If there exists a column l with $g_l < 0$ in a basic feasible solution, the value of the objective function can be increased by inserting the column vector belonging to the nonbasic variable x_{Nl} into the set of basis vectors, i.e variable x_{Nl} becomes basic variable in the next tableau.*

Starting with an initial basic feasible solution, we apply the *short tableau* of the pivoting procedure. An additional row contains the coefficients g_j together with the objective function value z_0 (i.e. the objective row) calculated as given above:

	nbv	x_{N1}	x_{N2}	$\cdots$	$\mathbf{x_{Nl}}$	$\cdots$	$x_{Nn'}$		
bv	-1	c_{N1}	c_{N2}	$\cdots$	c_{Nl}	$\cdots$	$c_{Nn'}$	0	Q
x_{B1}	c_{B1}	$\hat{a}_{11}$	$\hat{a}_{12}$	$\cdots$	$\hat{a}_{1l}$	$\cdots$	$\hat{a}_{1n'}$	$\hat{b}_1$	
x_{B2}	c_{B2}	$\hat{a}_{21}$	$\hat{a}_{22}$	$\cdots$	$\hat{a}_{2l}$	$\cdots$	$\hat{a}_{2n'}$	$\hat{b}_2$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	
$\mathbf{x_{Bk}}$	c_{Bk}	$\hat{a}_{k1}$	$\hat{a}_{k2}$	$\cdots$	$\mathbf{\hat{a}_{kl}}$	$\cdots$	$\hat{a}_{kn'}$	$\hat{b}_k$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	
x_{Bm}	c_{Bm}	$\hat{a}_{m1}$	$\hat{a}_{m2}$	$\cdots$	$\hat{a}_{ml}$	$\cdots$	$\hat{a}_{mn'}$	$\hat{b}_m$	
z		g_1	g_2	$\cdots$	g_l	$\cdots$	$g_{n'}$	z_0	

Determination of the pivot column l

Choose some column $l, 1 \leq l \leq n'$, such that $g_l < 0$. Often, a column l is used with

$$g_l = \min\{g_j \mid g_j < 0,\ j = 1, 2, \ldots, n'\}.$$

It is worth noting that the selection of the smallest negative coefficient g_l does not guarantee that the algorithm terminates after the smallest possible number of iterations. It only guarantees that there is the biggest increase in the objective function value when going towards the resulting next extreme point.

Determination of the pivot row k

We remind that after the pivoting step, feasibility of the basic solution must be maintained. Therefore, we choose row k with $1 \leq k \leq m$ such that

$$\frac{\hat{b}_k}{\hat{a}_{kl}} = \min\left\{\frac{\hat{b}_i}{\hat{a}_{il}} \mid \hat{a}_{il} > 0,\ i = 1, 2, \ldots, m\right\}.$$

To determine the above quotients, we added the last column Q in the tableau above, where we enter the quotient in each row in which the corresponding element in the chosen pivot column is positive.

If column l is chosen as pivot column, the corresponding variable x_{Nl} becomes basic variable in the next step. We also say that x_{Nl} is the *entering variable*, and the column of the initial matrix A belonging to variable x_{Nl} is entering the basis. Using row k as pivot row, the corresponding variable x_{Bk} becomes nonbasic variable in the next step. In this case, we say that x_{Bk} is the *leaving variable*, and the column vector of matrix A belonging to variable x_{Nl} is leaving the basis. Element $\hat{a}_{kl}$ is denoted as *pivot* or *pivot element*. It has been printed in bold face in the tableau together with the leaving and the entering variable.

The following two theorems characterize situations when either an optimal solution does not exist or when an existing optimal solution is not uniquely determined.

Theorem 6 If we have $g_l < 0$ for a coefficient of a nonbasic variable in the objective row and $\hat{a}_{il} \leq 0$ for all coefficients in column l, then the LPP does not have a (finite) optimal solution.

Theorem 7 If there exists a coefficient $g_l = 0$ in the objective row of an optimal solution such that $\hat{a}_{il} > 0$ for at least one coefficient in column l, then there exists another optimal basic feasible solution, where x_{Nl} is a basic variable.

Simplex Algorithm

Step 1: Transform the LPP into the standard form, where the constraints are given in canonical form as follows (we remind that it is assumed that no artificial variables are necessary to transform the given problem into standard form):

$$A_N \mathbf{x_N} + I\mathbf{x_B} = \mathbf{b}, \qquad \mathbf{x_N} \geq \mathbf{0}, \qquad \mathbf{x_B} \geq \mathbf{0}, \qquad \mathbf{b} \geq \mathbf{0}.$$

The initial basic feasible solution is

$$\mathbf{x} = \begin{pmatrix} \mathbf{x_N} \\ \mathbf{x_B} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}$$

with the objective function value $z_0 = \mathbf{c}^T\mathbf{x}$. Establish the corresponding initial tableau.

Step 2: Consider the coefficients $g_j,\ j = 1, 2, \ldots, n'$, of the nonbasic variables x_{Nj} in the objective row.
If $g_j \geq 0$ for $j = 1, 2, \ldots, n'$, then the current basic feasible solution is optimal, STOP. Otherwise, there is a coefficient $g_j < 0$ in the objective row.

Step 3: Determine column l with

$$g_l = \min\{g_j \mid g_j < 0,\ j = 1, 2, \ldots, n'\}$$

as pivot column.

Step 4: If $a_{il} \leq 0$ for $i = 1, 2, \ldots, m$, then STOP (in this case, there does not exist an optimal solution for the problem). Otherwise, there is at least one element $a_{il} > 0$.

Step 5: Determine the pivot row k such that

$$\frac{b_k}{a_{kl}} = \min\left\{\frac{b_i}{a_{il}} \mid a_{il} > 0,\ i = 1, 2, \ldots, m\right\}.$$

Step 6: Exchange the basic variable x_{Bk} of row k with the nonbasic variable x_{Nl} of column l and calculate the following values of the new tableau:

$$\begin{aligned}
\hat{a}_{kl} &= \frac{1}{a_{kl}}; \\
\hat{a}_{kj} &= \frac{a_{kj}}{a_{kl}}; \qquad \hat{b}_k = \frac{b_k}{a_{kl}}; \qquad j = 1, 2, \ldots, n',\ j \neq l; \\
\hat{a}_{il} &= -\frac{a_{il}}{a_{kl}}; \qquad i = 1, 2, \ldots, m,\ i \neq k; \\
\hat{a}_{ij} &= a_{ij} - \frac{a_{il}}{a_{kl}} \cdot a_{kj}; \qquad \hat{b}_i = b_i - \frac{a_{il}}{a_{kl}} \cdot b_k;
\end{aligned}$$

$$i = 1, 2, \ldots, m,\ i \neq k; \quad j = 1, 2, \ldots, n',\ j \neq l.$$

Moreover, we obtain for the values of the last row in the new tableau:

$$\begin{aligned}
\hat{g}_l &= -\frac{g_l}{a_{kl}}; \\
\hat{g}_j &= g_j - \frac{g_l}{a_{kl}} \cdot a_{kj}; \qquad j = 1, 2, \ldots, n',\ j \neq l; \\
\hat{z}_0 &= z_0 - \frac{g_l}{a_{kl}} \cdot b_k.
\end{aligned}$$

Consider the tableau obtained as new starting solution and go to step 2.

Example 3 *A firms intends to manufacture three types of products P_1, P_2 and P_3 so that the total production cost does not exceed 32,000 EUR. There are 420 working hours possible and 30 units of raw materials may be used. Additionally, the data presented in Table 2 are given.*

Table 2: *Data for Example 3*

Product	P_1	P_2	P_3
Selling price (EUR/piece)	*1,600*	*3,000*	*5,200*
Production cost (EUR/piece)	*1,000*	*2,000*	*4,000*
Required raw material per piece	*3*	*2*	*2*
Working time in hours per piece	*20*	*10*	*20*

The objective is to determine the quantities of each product so that the profit is maximized. Let x_i be the number of produced pieces of P_i, $i \in \{1, 2, 3\}$. We can formulate the above problem as an LPP as follows:

$$z = 6x_1 + 10x_2 + 12x_3 \to \max!$$

$$\begin{array}{lrcrcrcl}
s.t. & x_1 & + & 2x_2 & + & 4x_3 & \leq & 32 \\
 & 3x_1 & + & 2x_2 & + & 2x_3 & \leq & 30 \\
 & 2x_1 & + & x_2 & + & 2x_3 & \leq & 42 \\
 & & & x_1, x_2, x_3 \geq 0. & & & &
\end{array}$$

The objective function has been obtained by subtracting the production cost from the selling price and dividing the resulting profit by 100 for each product. Moreover, the constraint on the production cost has been divided by 1,000, and the constraint on the working time by 10.

Introducing now in the ith constraint the slack variable $x_{3+i} \geq 0$, we obtain the standard form together with the following initial tableau:

	nbv	x_1	x_2	$\mathbf{x_3}$		
bv	-1	6	10	12	0	Q
$\mathbf{x_4}$	0	1	2	**4**	32	8
x_5	0	3	2	2	30	15
x_6	0	2	1	2	42	21
		-6	-10	-12	0	

Choosing x_3 now as the entering variable (since it has the smallest negative coefficient in the objective row), variable x_4 becomes the leaving variable due to the quotient rule. We obtain:

	nbv	x_1	$\mathbf{x_2}$	x_4		
bv	-1	6	10	0	0	Q
x_3	12	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	8	16
$\mathbf{x_5}$	0	$\frac{5}{2}$	**1**	$-\frac{1}{2}$	14	14
x_6	0	$\frac{3}{2}$	0	$-\frac{1}{2}$	26	–
		-3	-4	3	96	

Choosing now x_2 as entering variable, x_5 becomes the leaving variable. We obtain the tableau:

	nbv	x_1	x_5	x_4		
bv	-1	6	0	0	0	Q
x_3	12	-1	$-\frac{1}{2}$	$\frac{1}{2}$	1	
x_2	10	$\frac{5}{2}$	1	$-\frac{1}{2}$	14	
x_6	0	$\frac{3}{2}$	0	$-\frac{1}{2}$	26	
		7	4	1	152	

Since now all coefficients g_j are positive, we get the following optimal solution from the latter tableau:

$$x_1 = 0, \quad x_2 = 14, \quad x_3 = 1, \quad x_4 = 0, \quad x_5 = 0, \quad x_6 = 26.$$

That means, the optimal solution is to produce no piece of product P_1, 14 pieces of product P_2 and one piece of product P_3. Taking into a account that the coefficients of the objective function were divided by 100, we get a total profit of 15,200 EUR.

Example 4 *We consider the following LPP:*

$$\begin{array}{rrcrcr} & z = -2x_1 - 2x_2 \to \text{min!} \\ s.t. & x_1 & - & x_2 & \geq & -1 \\ & -x_1 & + & 2x_2 & \leq & 4 \\ & & x_1, x_2 \geq 0. \end{array}$$

First, we transform the given problem into the standard form, i.e. we multiply the objective function and the first constraint by -1 and introduce the slack variables x_3 and x_4. We obtain:

$$\begin{array}{rrcrcrcrcr} & \overline{z} = 2x_1 + 2x_2 \to \text{max!} \\ s.t. & -x_1 & + & x_2 & + & x_3 & & & = & 1 \\ & -x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\ & & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

Now we can establish the first tableau:

	nbv	x_1	$\mathbf{x_2}$		
bv	-1	2	2	0	Q
$\mathbf{x_3}$	0	-1	**1**	1	1
x_4	0	-1	2	4	2
		-2	-2	0	

Since there are only negative elements in the column of variable x_1, only variable x_2 can be the entering variable. In this case, we get the quotients given in the last column of the latter tableau and therefore, variable x_3 is the leaving variable. We obtain the following tableau:

	nbv	$\mathbf{x_1}$	x_3		
bv	-1	2	0	0	Q
x_2	2	-1	1	1	
$\mathbf{x_4}$	0	**1**	-2	2	2
		-4	2	2	

In the latter tableau, there is only one negative coefficient of a nonbasic variable in the objective row, therefore, variable x_1 becomes the entering variable. Since there is only one positive element in the column belonging to x_1, variable x_4 becomes the leaving variable. We obtain the following tableau:

	nbv	x_4	x_3		
bv	-1	0	0	0	Q
x_2	2	1	-1	3	
x_1	2	1	-2	2	
		4	-6	10	

Since there is only one negative coefficient of a nonbasic variable in the objective row, variable x_3 should be chosen as entering variable. However, there are only negative elements in the column belonging to x_3. It means that we cannot perform a further pivoting step and there does not exist a finite solution of the maximization problem considered (i.e. the objective function value can become arbitrarily large, see Theorem 9.5).

Example 5 *Given is the following LPP:*

$$z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \to \min!$$

$$\begin{array}{llllllllllll} s.t. & 2x_1 & + & x_2 & + & x_3 & & & & & & \geq 4,000 \\ & & & x_2 & & & + & 2x_4 & + & x_5 & & \geq 5,000 \\ & & & & & x_3 & & & + & 2x_5 & + \; 3x_6 & \geq 3,000 \\ & & & & & & & & & & & \end{array}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.$$

To get the standard form, we notice that in each constraint there is one variable that occurs only in one constraint (variable x_1 occurs only in the first constraint, variable x_4 only in the second constraint and variable x_6 only in the third constraint). Therefore, we divide the first constraint *by the coefficient two of variable x_1, the second constraint by two and the third constraint by three. Then, we introduce a surplus variable in each of the constraints, multiply the objective function by -1 and obtain the standard form (again the variables are written in such a way that the identity submatrix of the coefficient matrix occurs now at the end):*

$$\overline{z} = -z = -x_1 - x_2 - x_3 - x_4 - x_5 - x_6 \rightarrow \max!$$

$$\begin{array}{llllllllll} s.t. & \frac{1}{2}x_2 + \frac{1}{2}x_3 & & - x_7 & & & + x_1 & & & = 2,000 \\ & \frac{1}{2}x_2 & + \frac{1}{2}x_5 & & - x_8 & & & + x_4 & & = 2,500 \\ & & \frac{1}{3}x_3 + \frac{2}{3}x_5 & & & - x_9 & & & + x_6 & = 1,000 \end{array}$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \geq 0.$$

This yields the following initial tableau:

	nbv	x_2	x_3	$\mathbf{x_5}$	x_7	x_8	x_9		
bv	-1	-1	-1	-1	0	0	0	0	Q
x_1	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	-1	0	0	$2,000$	$-$
x_4	-1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	-1	0	$2,500$	$5,000$
$\mathbf{x_6}$	-1	0	$\frac{1}{3}$	$\mathbf{\frac{2}{3}}$	0	0	-1	$1,000$	$1,500$
		0	$\frac{1}{6}$	$-\frac{1}{6}$	1	1	1	$-5,500$	

Choosing now x_5 as entering variable, we obtain the quotients given in the last column of the above tableau and therefore, x_6 is chosen as leaving variable. We obtain the following tableau:

	nbv	x_2	x_3	x_6	x_7	x_8	x_9		
bv	-1	-1	-1	-1	0	0	0	0	Q
x_1	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	-1	0	0	$2,000$	
x_4	-1	$\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{3}{4}$	0	-1	$\frac{3}{4}$	$1,750$	
x_5	-1	0	$\frac{1}{2}$	$\frac{3}{2}$	0	0	$-\frac{3}{2}$	$1,500$	
		0	$\frac{1}{4}$	$\frac{1}{4}$	1	1	$\frac{3}{4}$	$-5,250$	

Now all coefficients of the nonbasic variables in the objective row are nonnegative and from the latter tableau we obtain the following optimal solution:

$$x_1 = 2,000, \quad x_2 = x_3 = 0, \quad x_4 = 1,750, \quad x_5 = 1,500, \quad x_6 = 0$$

with the optimal objective function value $\overline{z}_0^{max} = -5,250$, which corresponds to $z_0^{min} = 5,250$ (for the original minimization problem). Notice that the optimal solution is not uniquely determined. In the last tableau, there is one coefficient in the objective row equal to zero. Taking x_2 as the entering variable, the quotient rule determines x_4 as the leaving variable, and the following basic feasible solution with the same objective function value is obtained:

$$x_1 = 250, \quad x_2 = 3,500, \quad x_3 = x_4 = 0, \quad x_5 = 1,500, \quad x_6 = 0.$$

1.6 The 2-Phase Simplex Algorithm

The introduction of artificial variables is necessary when at least one constraint is an equation with no eliminated variable that has coefficient +1, e.g. the constraints may have the following form:

$$\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{array}$$

$$x_j \geq 0,\ i = 1, 2, \ldots, n; \quad b_i \geq 0,\ i = 1, 2, \ldots, m.$$

As discussed in case (e) of generating the standard form, we introduce an artificial variable x_{Ai} in each equation. Additionally we replace the original objective function z by an objective function z_I minimizing the sum of all artificial variables (or equivalently, maximizing the negative sum of all artificial variables) since it is our goal that all artificial variables will get value zero to ensure feasibility for the original problem. This gives the following linear programming problem to be considered in phase I:

$$z_I = -x_{A1} - x_{A2} - \cdots - x_{Am} \longrightarrow \text{max!}$$

$$\begin{array}{lllllll} \text{s.t.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & + x_{A1} & & & = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & & + x_{A2} & & = b_2 \\ \cdots & \cdots\cdots\cdots\cdots\cdots\cdots & & & & \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & & & + x_{Am} & = b_m \end{array} \tag{11}$$

$$x_j \geq 0,\ j = 1, 2, \ldots, n \quad \text{and} \quad x_{Ai} \geq 0,\ i = 1, 2, \ldots, m.$$

Assume that we have determined an optimal solution of the auxiliary problem (11) by the simplex method, i.e. the procedure stops with $g_j \geq 0$ for all coefficients of the nonbasic variables in the objective row (for the auxiliary objection function z_I).
Then, at the end of phase I, the following cases are possible:

(1) $z_I^{\max} < 0 \iff$ The initial problem does not have a feasible solution.

(2) We have $z_I^{\max} = 0$ and one of the following cases:

  (2a) All artificial variables are nonbasic variables. Then the basic solution obtained represents a feasible canonical form for the initial problem, and we can start with phase II of the simplex algorithm described in Section 1.5.

  (2b) Among the basic variables, there is still an artificial variable: $x_{Bk} = x_{Al} = 0$ (degeneration case). Then we have again one of the two possibilities:

i) In the row belonging to the basic variable $x_{Al} = 0$, all coefficients are also equal to zero. In this case, the corresponding equation is superfluous and can be omitted.

ii) In the row belonging to the basic variable $x_{Al} = 0$, we have $\hat{a}_{kj} \neq 0$ for at least one coefficient. Then we can choose $\hat{a}_{kj}$ as pivot element and replace the artificial variable x_{Al} by the nonbasic variable x_{Nj}.

Example 6 *Given is the following LPP:*

$$
\begin{array}{lrcrcl}
 & z = x_1 - 2x_2 \to \max! & & & & \\
s.t. & x_1 & + & x_2 & \leq & 4 \\
 & 2x_1 & - & x_2 & \geq & 1 \\
 & x_1, x_2 \geq 0. & & & &
\end{array}
$$

We transform the given problem into standard form by introducing a surplus variable (x_3) in the second constraint, a slack variable (x_4) in the first constraint and an artificial variable (x_{A1}) in the second constraint. Now we replace the objective function z by the auxiliary function z_I. Thus, in phase I of the simplex method, we consider the following LPP:

$$
\begin{array}{lrcrcrcrcrcl}
 & & & z_I = -x_{A1} \to \max! & & & & & & & & \\
s.t. & x_1 & + & x_2 & & & + & x_4 & & & = & 4 \\
 & 2x_1 & - & x_2 & - & x_3 & & & + & x_{A1} & = & 1 \\
 & & & x_1, x_2, x_3, x_4, x_{A1} \geq 0. & & & & & & & &
\end{array}
$$

We start with the following tableau:

	nbv	$\mathbf{x_1}$	x_2	x_3		
bv	-1	0	0	0	0	Q
x_4	0	1	1	0	4	4
$\mathbf{x_{A1}}$	-1	**2**	-1	-1	1	$\frac{1}{2}$
		-2	1	1	-1	

Choosing x_1 as entering variable gives the quotients presented in the last column of the tableau above, and variable x_{A1} becomes the leaving variable. This leads to the following tableau:

	nbv	x_{A1}	x_2	x_3		
bv	-1	-1	0	0	0	Q
x_4	0	$-\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{7}{2}$	
x_1	0	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	–
		1	0	0	0	

Now phase I is finished, we drop variable x_{A1} and the corresponding column, use the original objective function and determine the coefficients g_j of the objective row. This yields the following *tableau:*

	nbv	x_2	$\mathbf{x_3}$		
bv	-1	-2	0	0	Q
$\mathbf{x_4}$	0	$\frac{3}{2}$	$\mathbf{\frac{1}{2}}$	$\frac{7}{2}$	7
x_1	1	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-$
		$\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	

Due to the negative coefficient in the objective row, we choose x_3 as the entering variable in the next step and variable x_4 becomes the leaving variable. Then we obtain the following tableau:

	nbv	x_2	x_4		
bv	-1	-2	0	0	Q
x_3	0	3	2	7	
x_1	1	1	1	4	
		3	1	4	

Since all coefficients g_j are nonnegative, the obtained solution is optimal: $x_1 = 4, x_2 = 0$. The introduced surplus variable x_3 is equal to seven while the introduced slack variable x_4 is equal to zero. The optimal objective function value is $z_0^{max} = 4$.

Let us consider another LPP assuming that the objective function changes now to

$$\tilde{z} = -x_1 + 3x_2 \longrightarrow \min!$$

Can we easily decide whether the optimal solution for the former objective function is also optimal for the new one? We replace only the coefficients c_1 and c_2 of the objective function in the last tableau (again for the maximization version of the problem), recompute the coefficients g_j of the objective row and obtain the following tableau:

	nbv	x_2	x_4		
bv	-1	-3	0	0	Q
x_3	0	3	2	7	
x_1	1	1	1	4	
		4	1	4	

Since also in this case all coefficients g_j in the objective row are nonnegative, the solution $x_1 = 4, x_2 = 0$ is optimal for $\tilde{z} = -x_1 + 3x_2 \longrightarrow \min$ with a function value $\tilde{z}_0^{min} = -4$.

Example 7 *We consider the data given in Example 1 and apply the 2-phase simplex method. Transforming the given problem into standard form we obtain:*

$$\overline{z} = -25x_1 - 17x_2 - 12x_3 \to \max!$$

$$\begin{array}{llllllllllllllll} s.t. & x_1 & + & x_2 & + & x_3 & + & x_{A1} & & & & & & & = & 100 \\ & & & & & x_3 & & & + & x_4 & & & & & = & 30 \\ & x_1 & & & + & x_3 & & & & & + & x_5 & & & = & 50 \\ & & & x_2 & + & x_3 & & & & & & & + & x_6 & = & 90 \\ \end{array}$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_{A1} \geq 0.$$

Starting with phase I of the simplex method, we replace function z by the auxiliary objective function

$$z_I = -x_{A1} \longrightarrow \max!,$$

and we obtain the following initial tableau:

	nbv	$\mathbf{x_1}$	x_2	x_3		
bv	-1	0	0	0	0	Q
x_{A1}	-1	1	1	1	100	100
x_4	0	0	0	1	30	–
$\mathbf{x_5}$	0	**1**	0	1	50	50
x_6	0	0	1	1	90	–
		-1	-1	-1	-100	

Choosing x_1 as the entering variable, we get the quotients given above and select x_5 as the leaving variable. This leads to the following tableau:

	nbv	x_5	$\mathbf{x_2}$	x_3		
bv	-1	0	0	0	0	Q
$\mathbf{x_{A1}}$	-1	-1	**1**	0	50	50
x_4	0	0	0	1	30	–
x_1	0	1	0	1	50	–
x_6	0	0	1	1	90	90
		1	-1	0	-50	

Now x_2 becomes the entering variable and the artificial variable x_{A1} is the leaving variable. We get the following tableau, where the superfluous column belonging to x_{A1} is dropped:

	nbv	x_5	x_3		
bv	-1	0	0	0	Q
x_2	-1	-1	0	50	
x_4	0	0	1	30	
x_1	0	1	1	50	
x_6	0	1	1	40	
		0	0	0	

Now, phase I is finished, and we can consider the objective function

$$\overline{z} = -25x_1 - 17x_2 - 12x_3 \longrightarrow \max!$$

We recompute the coefficients in the objective row and obtain the following tableau:

	nbv	x_5	$\mathbf{x_3}$		
bv	-1	0	-12	0	Q
x_2	-17	-1	0	50	–
$\mathbf{x_4}$	0	0	**1**	30	30
x_1	-25	1	1	50	50
x_6	0	1	1	40	40
		-8	-13	$-2,100$	

We choose x_3 as entering variable and based on the quotients given in the last column, x_4 is the leaving variable. After this pivoting step, we get the following tableau:

	nbv	$\mathbf{x_5}$	x_4		
bv	-1	0	0	0	Q
x_2	-17	-1	0	50	–
x_3	-12	0	1	30	–
x_1	-25	1	-1	20	20
$\mathbf{x_6}$	0	**1**	-1	10	10
		-8	13	$-1,710$	

We choose x_5 as entering variable and x_6 as leaving variable which gives the following tableau:

	nbv	x_6	x_4		
bv	-1	0	0	0	Q
x_2	-17	1	-1	60	
x_3	-12	0	1	30	
x_1	-25	-1	0	10	
x_5	0	1	-1	10	
		8	5	$-1,630$	

The last tableau gives the following optimal solution:

$$x_1 = 10, \quad x_2 = 60, \quad x_3 = 30, \quad x_4 = 0, \quad x_5 = 10, \quad x_6 = 0$$

with the objective function value $z_0^{min} = 1,630$ for the minimization problem.

Example 8 *Consider the following LPP:*

$$\begin{aligned} & z = x_1 + 2x_2 \to \max! \\ s.t. \quad & \begin{array}{rcrcl} x_1 & - & x_2 & \geq & 1 \\ 5x_1 & - & 2x_2 & \leq & 3 \end{array} \\ & x_1, x_2 \geq 0. \end{aligned}$$

Transforming the above problem into standard form, we obtain

$$z = x_1 + 2x_2 \to \max!$$

$$\begin{array}{rcrcrcrcrcl} x_1 & - & x_2 & - & x_3 & + & x_{A1} & & & = & 1 \\ 5x_1 & - & 2x_2 & & & & & + & x_4 & = & 3 \end{array}$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0.$$

This leads to the following starting tableau for phase I with the auxiliary objective function $z_I = -x_{A1} \to \max!$

	nbv	$\mathbf{x_1}$	x_2	x_3		
bv	-1	0	0	0	0	Q
x_{A1}	-1	1	-1	-1	1	1
$\mathbf{x_4}$	0	**5**	-2	0	3	$\frac{3}{5}$
		-1	1	1	-1	

We choose now variable x_1 as entering variable, which gives the leaving variable x_4. This yields the following tableau:

	nbv	x_4	x_2	x_3		
bv	-1	0	0	0	0	Q
x_{A1}	-1	$-\frac{1}{5}$	$-\frac{3}{5}$	-1	$\frac{2}{5}$	
x_1	0	$\frac{1}{5}$	$-\frac{2}{5}$	0	$\frac{3}{5}$	
		$\frac{1}{5}$	$\frac{3}{5}$	1	$-\frac{2}{5}$	

So, we finish with case (1) described earlier, i.e. $z_I^{max} < 0$. Consequently, the above LPP does not have a feasible solution. In fact, in the final tableau, variable x_{A1} is still positive (so the original constraint $x_1 - x_2 - x_3 = 1$ is violated).

2 Discrete Optimization

2.1 Preliminaries

Discrete Optimization Problem:

$$f(\mathbf{x}) \rightarrow \min! \quad (\max!)$$
$$\mathbf{x} \in S$$

Special case: S finite

$\rightarrow$ Often S is described by linear inequalities / equations.

Integer (Linear) Optimization Problem:

$$f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x} \rightarrow \min! \quad (\max!)$$

s.t.

$$A \cdot \mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{Z}_+^n$$

Parameters A, **b**, **c** integer

$\mathbb{Z}_+^n$ - Set of integer, non-negative, n-dimensional vectors

Mixed Integer (Linear) Optimization Problem:

replace $\mathbf{x} \in \mathbb{Z}_+^n$ by

$$x_1, x_2, \ldots, x_r \in \mathbb{Z}_+$$
$$x_{r+1}, x_{r+2}, \ldots, x_n \in \mathbb{R}_+$$

Binary Optimization Problem:

replace $\mathbf{x} \in \mathbb{Z}_+^n$ by

$$x_1, x_2, \ldots, x_n \in \{0, 1\}$$
$$\text{i.e.,} \quad \mathbf{x} \in \{0, 1\}^n$$

Mixed Binary Optimization Problem:

replace $\mathbf{x} \in \mathbb{Z}_+^n$ by

$$x_1, x_2, \ldots, x_r \in \{0, 1\}$$
$$x_{r+1}, x_{r+2}, \ldots, x_n \in \mathbb{R}_+$$

Combinatorial Optimization Problem (COP):

The set S is finite and non-empty.

Example 9 (Investment planning) *An enterprise may realize 5 projects with the following expenditures (in Mill. EUR) for the next three years.*

Project	*year 1*	*year 2*	*year 3*	*profit*
1	*5*	*1*	*8*	*20*
2	*4*	*7*	*10*	*40*
3	*3*	*9*	*2*	*20*
4	*7*	*4*	*1*	*15*
5	*8*	*6*	*10*	*30*
Available budgets	*25*	*25*	*25*	

Which projects should be realized in order to maximize the profit?

2.2 Branch and Bound Algorithms (B&B)

- Exact procedure
- Method of implicit enumeration: Exclude successively subsets of S which cannot contain an optimal solution.
- Basic idea for minimization problems:
  - Branching: Partition the set of solutions at least into two (disjoint) subsets.
  - Bounding: Determine for each subset $S(i)$ a lower bound $LB(i)$
  - Let UB be a known upper bound and $LB(i) \geq UB$ for $S(i)$, then $S(i)$ does not need to be considered further.

First we consider a *binary optimization problem*:

$$f(\mathbf{x}) \to \min!$$

s.t.

$$\mathbf{x} \in S \subseteq \{0,1\}^n$$

Remark: In the case of a complete enumeration for $n = 50$, we would already obtain

$$| \{0,1\}^{50} | = 2^{50} \approx 10^{15}$$

possible combinations.

States of variables

Variable u_j describes the state of x_j as follows:

State of x_j	Value of x_j	Value of u_j
fixed "settled"	1	1
fixed "locked"	0	0
free	$0 \vee 1$	-1

- Vector $\mathbf{u} \in U := \{-1,0,1\}^n$ is identified with node u in the branching tree. Node u restricts the set of solutions as follows:

$$S(u) = \{\mathbf{x} \in S \mid x_j = u_j,\ x_j \text{ fixed}\}, \quad j \in \{1,\dots,n\}$$

- To node u, there corresponds the following optimization problem:

$$\left.\begin{array}{rl} & f(\mathbf{x}) \to \min! \\ \text{s.t.} & \\ & \mathbf{x} \in S(u) \end{array}\right\} \quad P(u)$$

Let $f^*(u) := \min\{f(\mathbf{x}) \mid \mathbf{x} \in S(u)\}$.

Introduction of bound functions

Definition 7 *A function* $LB: U \to \mathbb{R} \cup \{\infty\}$ *is called a* lower bound function*, if*

(a) $LB(u) \leq \min\{f(\mathbf{x}) \mid \mathbf{x} \in S(u)\} = f^*(u)$

(b) $S(u) = \{\mathbf{x}\} \Rightarrow LB(u) = f(\mathbf{x})$

(c) $S(u) \subseteq S(v) \Rightarrow LB(u) \geq LB(v)$

Definition 8 $UB \in \mathbb{R}$ *is called an* upper bound *on the optimal objective function value, if* $UB \geq \min\{f(\mathbf{x}) \mid \mathbf{x} \in S\}$.

$\bar{\mathbf{x}}$ represents the best solution found so far.

At the beginning of a B&B procedure, we set $UB := f(\bar{\mathbf{x}})$, if $\bar{\mathbf{x}}$ a heuristic solution, or we set $UB := \infty$.

Generation of the branching tree

active node: a node, which has *not* been investigated yet

At the beginning, the branching tree contains only the root $u = (-1, -1, \ldots, -1)^T$ as active node.

Investigation of an active node u

- *Case 1:* $LB(u) \geq UB$

  Node u is removed from the branching tree, since according to Definition 5 (a)

  $$\min\{f(\mathbf{x}) \mid \mathbf{x} \in S(u)\} \geq LB(u) \geq UB$$

  holds.

  → Problem can be excluded.

- *Case 2a:* $LB(u) < UB$ with $u_j \in \{0, 1\}$ for $j = 1, 2, \ldots, n$

  solution $x = u$ is uniquely determined

If $\mathbf{x} \in S \quad \Rightarrow \qquad$ due to

$$f(\mathbf{x}) = LB(u) < UB = f(\bar{\mathbf{x}}),$$

we have found a new best solution. Set $\bar{\mathbf{x}} := \mathbf{x}$ and $UB := f(\bar{\mathbf{x}})$. (Node u is no longer active.)

$\rightarrow$ Problem can be excluded.

- *Case 2b:* $LB(u) < UB$ with $u_j = -1$ for (at least) one $j \in \{1, 2, \ldots, n\}$

  Generate the successor nodes w^i of node u by fixing one (or several) free variables. (Node u is no longer active, but the successors w^i of u are active.)

  $\rightarrow$ Problem is branched.

Search strategies - Selection of the next active node to be selected for investigation

(a) *FIFO strategy* (first in, first out)

Newly generated nodes are added to the end of the queue and the node at the beginning of the queue is investigated next.

$\rightarrow$ Breadth first search

(b) *LIFO strategy* (last in, first out)

Newly generated nodes are added to the end of the queue and the node at the end of the queue is investigated first.

$\rightarrow$ Depth first search

(c) *LLB strategy* (least lower bound)

The node with the smallest $LB(u)$ is investigated next.
(If $LB(u) \geq UB$, then stop.)

The LIFO strategy delivers often quickly feasible solutions. During the course of the search, it is often recommendable to switch to the FIFO or LLB strategy.

On the bound function $LB(u)$

One or several constraints of $P(u)$ are "relaxed" or removed.

$\Rightarrow$ One obtains an easier problem $P^*(u)$ with $S^*(u) \supseteq S(u)$.

$P^*(u) \;\rightarrow\;$ *Relaxation* of $P(u)$

Set $LB(u) := f(\mathbf{x}^*(u))$, where $\mathbf{x}^*(u)$ is an optimal solution of $P^*(u)$.

Binary problem: For the free variables, replace $x_i \in \{0, 1\}$ by $0 \le x_i \le 1$.

(*LP relaxation*)

B&B algorithm for binary optimization problems (minimization)

Step 1:

- If a feasible solution $\mathbf{x} \in S$ is known, set

$$\overline{\mathbf{x}} := \mathbf{x} \quad \text{and} \quad UB := f(\mathbf{x}),$$

  otherwise set

$$UB := \infty.$$

- Set $u^0 := (-1, -1, \ldots, -1)^T$ and $U_a := \{u^0\}$ (u^0 is the root).

Step 2:

- If $U_a = \emptyset$, go to Step 4.

  Otherwise, select by means of a search strategy a node $u \in U_a$, remove u from U_a and calculate $LB(u)$.

Step 3:

- If $LB(u) \ge UB$, go to Step 4 in the case of the LLB strategy.

  Otherwise, eliminate u from the branching tree.

- If $LB(u) < UB$ and all variables are fixed, set in the case of $\mathbf{x} \in S$:

$$\overline{\mathbf{x}} := \mathbf{x} \quad \text{and} \quad UB := f(\mathbf{x}).$$

- If $LB(u) < UB$ and at least one variable is free, generate by fixing one (or several) free variables the successors of node u. Add the successors of u to U_a and to the branching tree.

- Go to Step 2.

Schritt 4: (Stop)

- If $UB < \infty$, then $\overline{\mathbf{x}}$ is an optimal solution with $f(\mathbf{x}) = UB$. Otherwise, the problem has no feasible solution.

This procedure can be generalized to mixed binary problems of the form

$$f(\mathbf{x}, \mathbf{y}) \to \min!$$

s.t.

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in S$$

$$\mathbf{x} \in \mathbb{R}^n, \ \mathbf{y} \in \{0, 1\}^k.$$

If all binary variables are fixed, we have an LPP in the variables $x_1, x_2, \dots, x_n$.

Modifications for integer programming problems

Use as relaxation the resulting LPP, where $x_i \in \mathbb{Z}_+$ is replaced by $x_i \geq 0$ (*LP Relaxation*).

The optimal solution (OS) gives a lower bound $LB(u)$ for node u (we have $S^*(u) \supseteq S(u)$).

Algorithm by Dakin: (branching strategy)

If in the OS of the LPP at least one variable x_i^* is not integer, generate *two* successor nodes v^k und v^l by adding the following constraints:

$$x_i \leq [x_i^*] \text{ in } S(v^k) \qquad \text{and}$$

$$x_i \geq [x_i^*] + 1 \text{ in } S(v^l)$$

2.3 Knapsack Problem

Problem: A climber can use n items $1, 2, \dots, n$, where

c_i - value of item i

a_i - volume of item i

V - volume of the knapsack

Goal: Determine a knapsack filling with maximal total value such that the volume V is not exceeded.

$\Rightarrow$ Introduce binary variables x_i as follows:

$$x_i = \begin{cases} 1, \text{ if item } i \text{ is put into the knapsack} \\ 0, \text{ otherwise} \end{cases}$$

$$\text{for } i = 1, 2, \ldots, n$$

$\Rightarrow$ mathematical model:

$$\boxed{\begin{array}{ll} & \displaystyle\sum_{i=1}^{n} c_i x_i \to \max! \\ \text{s.t.} & \left.\begin{array}{l} \displaystyle\sum_{i=1}^{n} a_i x_i \leq V \\ x_1, x_2, \ldots, x_n \in \{0,1\} \end{array}\right\} =: S \end{array}} \qquad (K)$$

Problem (K) is a binary optimization problem with only one constraint.

Greedy Algorithm (heuristic algorithm)

Step 1:

Number the n items according to non-increasing weights $\frac{c_i}{a_i}$ and set $f_G := 0$.

Step 2:

For $j = 1, 2, \ldots, n$ do:

If $a_j > V$, set $k := j$, $x_j^G := 0$ and go to Step 3, otherwise set $x_j^G := 1$, $f_G := f_G + c_j$ and $V := V - a_j$.

Step 3:

For $j = k+1, k+2, \ldots, n$ do:

If $a_j > V$, set $x_j^G := 1$, $f_G := f_G + c_j$ and $V := V - a_j$.

k - critical index

Some remarks on the B&B algorithm for the knapsack problem

1. See also B&B algorithm for binary optimization problems, but: maximization problem.

2. Determine by means of the greedy algorithm a feasible solution $\mathbf{x}^G$ and set $\bar{\mathbf{x}} := \mathbf{x}^G$ and $LB = f_G$.

3. For calculating upper bounds $UB(u)$, use the LP relaxation, i.e., replace the free variables $x_j \in \{0,1\}$ by $0 \leq x_j \leq 1$.

4. Possibly, the dimension of the initial problem can be reduced.

Theorem 8 **Let f_{LP} be the optimal objective function value of the LP relaxation and f_G the objective function value obtained by the greedy algorithm for the knapsack problem (K) and k be the critical index.**

Then there exists an optimal solution $\mathbf{x}^*$ of problem (K) with the following properties:

- **If for $j \in \{1, 2, \ldots, k-1\}$**

$$f_{LP} - f_G \leq c_j - \frac{a_j\ c_k}{a_k}\ ,$$

  then $x_j^* = 1$.

- **If for $j \in \{k+1, k+2, \ldots, n\}$**

$$f_{LP} - f_G < \frac{a_j\ c_k}{a_k} - c_j\ ,$$

  then $x_j^* = 0$.

Remark: Theorem 1 reduces problem (K) to a *core problem.*

3 Metaheuristics

3.1 Local Search, Preliminaries

Introduce a neighborhood structure as follows:

$$N \; : \; S \rightarrow 2^S$$
$$\mathbf{x} \in S \;\Rightarrow\; N(\mathbf{x}) \subseteq 2^S$$

S - Set of feasible solutions

$N(\mathbf{x})$ - Set of neighbors of a feasible solution $\mathbf{x} \in S$

Algorithm *ITERATIVE IMPROVEMENT*

1. determine an initial solution $\mathbf{x} \in S$;

   REPEAT

2. determine the best solution $\mathbf{x}' \in N(\mathbf{x})$;
3. **IF** $f(\mathbf{x}') < f(\mathbf{x})$ **THEN** $\mathbf{x} := \mathbf{x}'$;

   UNTIL $f(\mathbf{x}') \geq f(\mathbf{x})$ for all $\mathbf{x}' \in N(\mathbf{x})$.

$\mathbf{x}'$ - local minimal point w.r.t. neighborhood N

$\rightarrow$ The algorithm works with "largest improvement" *(best-fit)*.

Modification:

Use "first improvement" *(first-fit)*, i.e., search the neighborhood in a systematic way and accept a neighbor with a better objective function value than the current starting solution immediately for the next iteration.

(Stop, if a complete cycle with all neighbors has been checked without getting a better objective function value.)

$|$ **N(x)** $|$ **very large** $\Rightarrow$ Generate the neighbors randomly.

$\Rightarrow$ Replace row 2 in algorithm "Iterative Improvement" by

$$2^*\text{: Determine a solution } \mathbf{x}' \in N(\mathbf{x})$$

Stop, if

- a settled time limit is elapsed or
- a settled number of feasible solutions has been generated or
- a settled number of solutions after the last objective function value improvement has been generated without improving the objective function value further.

We consider

$$f(\mathbf{x}) \rightarrow \min! \quad (\max!)$$

s.t.

$$\mathbf{x} \in S \subseteq \{0,1\}^n$$

Neighborhood $N_k(\mathbf{x})$**:**

$$N_k(\mathbf{x}) = \{\mathbf{x}' \in S \mid \sum_{i=1}^{n} \mid x_i - x_i' \mid \leq k\}$$

($\mathbf{x}' \in N_k(\mathbf{x}') \;\Leftrightarrow\; \mathbf{x}'$is feasible and differs in at most k components from $\mathbf{x}$)

$$\Rightarrow \mid N_1(\mathbf{x}) \mid \quad \leq n$$

$$\mid N_2(\mathbf{x}) \mid \quad \leq n + \binom{n}{2} = n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2}$$

For the systematic generation of neighbors, change component 1,2,... etc.

3.2 Simulated Annealing

randomized procedure, since

- $\mathbf{x}' \in N(\mathbf{x})$ is randomly selected

- in the i-th iteration, $\mathbf{x}'$ is accepted with probability

$$\min\left\{1, exp\Big(- \frac{f(\mathbf{x}') - f(\mathbf{x})}{t_i}\Big)\right\}$$

  as new starting solution ($\{t_i\}$ is a sequence of positive control parameters known as the temperature).

Algorithm *SIMULATED ANNEALING*

1. $i := 0$; choose t_0;
2. determine an initial solution $\mathbf{x} \in S$;
3. $best := f(\mathbf{x})$;
4. $\mathbf{x}^* := \mathbf{x}$;

   REPEAT

5. generate randomly a solution $\mathbf{x}' \in N(\mathbf{x})$;
6. **IF** $rand[0,1] < \min\left\{1, exp\left(-\frac{f(\mathbf{x}')-f(\mathbf{x})}{t_i}\right)\right\}$ **THEN** $\mathbf{x} := \mathbf{x}'$;
7. **IF** $f(\mathbf{x}') < best$ **THEN**

   BEGIN $\mathbf{x}^* := \mathbf{x}'$; $best := f(\mathbf{x}')$ **END**;

8. $t_{i+1} := g(t_i)$;
9. $i := i + 1$;

   UNTIL stopping criterion is satisfied.

Modification:

Threshold Accepting (deterministic variant of Simulated Annealing)

- accept $\mathbf{x}' \in N(\mathbf{x})$ if

$$f(\mathbf{x}') - f(\mathbf{x}) \leq t_i$$

  t_i – *Threshold* in the i-th iteration

3.3 Tabu Search

Goal: Avoidance of ‘short cycles’

$\Rightarrow$ use attributes to characterize the solutions attended recently and forbid the returnal to such solutions for a specified number of iterations

Notations:

$Cand(\mathbf{x})$	–	contains all neighbors $\mathbf{x}' \in N(\mathbf{x})$, to which a transition (‘move’) is allowed
TL	–	tabu list
t	–	length of the tabu list

Algorithm *TABU SEARCH*

1. determine an initial solution $\mathbf{x} \in S$;
2. $best := f(\mathbf{x})$;
3. $\mathbf{x}^* := \mathbf{x}$;
4. $TL := \emptyset$;

   REPEAT

5. determine $Cand(\mathbf{x}) = \{ \mathbf{x}' \in N(\mathbf{x}) \mid \text{the move from } \mathbf{x} \text{ to } \mathbf{x}' \text{ is not tabu} \}$;
6. select a solution $\overline{\mathbf{x}} \in Cand(\mathbf{x})$;
7. update TL (such that maximal t attributes are contained in TL);
8. $\mathbf{x} := \overline{\mathbf{x}}$;
9. **IF** $f(\overline{\mathbf{x}}) < best$ **THEN**

   BEGIN $\mathbf{x}^* := \overline{\mathbf{x}}$; $best := f(\overline{\mathbf{x}})$ **END**;

   UNTIL stopping criterion is satisfied.

3.4 Genetic Algorithms

- Use of **Darwin's evolution theory** (survival of the fittest)

- Genetic algorithms work with a **population of individuals** (chromosomes), which are characterized by their fitness

- Generation of offspring by **genetic operators** (crossover, mutation)

Fitness and Encoding of an Individual

e.g. fitness(ch)=$f(\mathbf{x})$ for $f \to$ max!

fitness(ch)=$\frac{1}{f(\mathbf{x})}$ for $f \to$ min! and $f(\mathbf{x}) > 0$,

where ch denotes the encoding of solution $\mathbf{x} \in S$

$$\mathbf{x} = (0,1,1,1,0,1,0,1)^T \in \{0,1\}^8$$

ch:

0	1	1	1	0	1	0	1

Genetic Operators for Generating Offspring

<u>Mutation:</u>

"Mutate" the genes of an individual.

parent chromosome

0	1	1	1	0	1	0	1

(3,5)-Inversion

0	1	**0**	**1**	**1**	1	0	1

2-Mutation

0	**0**	1	1	0	1	0	1

(1,4,7)-Mutation

1	1	1	**0**	0	1	**1**	1

<u>Crossover:</u>

Combine the genetic structures of two individuals and generate two offspring.

1-Point-Crossover e.g. (4,8)-Crossover

P_1 1 0 1 | 0 0 1 0 1 → O_1 1 0 1 | **1 0 0 1 1**

P_2 0 1 1 | 1 0 0 1 1 → O_2 0 1 1 | **0 0 1 0 1**

2-Point-Crossover e.g. (3,5)-Crossover

P_1 1 0 | 1 0 0 | 1 0 1 → O_1 1 0 | **1 1 0** | 1 0 1

P_2 0 1 | 1 1 0 | 0 1 1 → O_2 0 1 | **1 0 0** | 0 1 1

Algorithm *GEN-ALG*

1. set the parameters population size $POPSIZE$, maximal number of generations $MAXGEN$, probability P_{CO} for the application of a crossover and probability P_{MU} for the application of a mutation;
2. generate the initial population POP_0 with $POPSIZE$ individuals (chromosomes);
3. determine the fitness of all individuals;
4. $k := 0$;

   WHILE $k < MAXGEN$ **DO**

   BEGIN
5. $h := 0$;

   WHILE $h < POPSIZE$ **DO**

   BEGIN
6. select two parents from POP_k (e.g. randomly proportional to their fitness values or according to roulette wheel selection);
7. apply with probability P_{CO} a crossover to the selected parents;
8. apply with probability P_{MU} a mutation to each of the individuals;
9. $h := h + 2$;

   END;
10. $k := k + 1$;
11. select from the generated offspring (and possibly also from the parents) $POPSIZE$ individuals of the k-th generation POP_k (e.g. proportional to their fitness values);

    END

4 Dynamic Programming

→ Problems are considered, which can be partitioned into particular stages so that the overall optimization can be replaced by a 'stepwise optimization' over the stages.

→ Dynamic programming is often applied to an optimal control of economic processes, where the stages correspond to time periods.

4.1 Introductory Examples

(a) Inventory Problem

Problem Formulation:

- A good is stored during a finite planning horizon consisting of n periods.
- In each period, a delivery to the inventory is possible at the beginning.
- There is a demand in each period, which has to be satisfied after a potential delivery.

Notations:

$u_j \geq 0$ - the amount delivered at the beginning of period j

$r_j \geq 0$ - demand in period j

x_j - stock immediately before the delivery in period j $(j = 1, 2, \ldots, n)$

Optimization problem:

$$\sum_{j=1}^{n} \left[K\delta(u_j) + hx_{j+1} \right] \to \min!$$

s.t.

$$\begin{aligned} x_{j+1} &= x_j + u_j - r_j, \quad && j = 1, 2, \ldots, n \\ x_1 &= x_{n+1} = 0 && \\ x_j &\geq 0, && j = 2, 3, \ldots, n \\ u_j &\geq 0, && j = 1, 2, \ldots, n \end{aligned} \tag{12}$$

Remark:

$$x_1 = x_{n+1} = 0 \quad \text{and} \quad (12)$$

$\Rightarrow$ Replace in the objective function hx_{j+1} by hx_j such that each term in the sum has the form $g_j(x_j, u_j)$.

$$x_j = x_{j+1} - u_j + r_j \geq 0 \quad \Rightarrow \quad u_j \leq x_{j+1} + r_j$$

The constraints can be formulated as follows:

$$\begin{aligned} x_1 &= x_{n+1} = 0 \\ x_j &= x_{j+1} - u_j + r_j, \quad && j = 1, 2, \ldots, n \\ x_j &\geq 0, && j = 1, 2, \ldots, n \\ 0 \leq u_j &\leq x_{j+1} + r_j, && j = 1, 2, \ldots, n \end{aligned}$$

(b) Knapsack Problem

$$u_j := \begin{cases} 1, & \text{if item } j \text{ is put into the knapsack} \\ 0, & \text{otherwise} \end{cases}$$

Optimization problem:

$$\sum_{j=1}^{n} c_j u_j \ \rightarrow \max!$$

s.t.

$$\sum_{j=1}^{n} a_j u_j \qquad \leq V$$

$$u_1, u_2, \ldots, u_n \in \{0, 1\}$$

$\rightarrow$ Here the states are no time periods. The decisions which of the items $1, 2, \ldots, n$ are put into the knapsack is interpreted as decisions in n successive stages.

x_j - remaining volume of the knapsack for the items $j,\ j+1, \ldots, n$

$\Rightarrow x_1 = V$ and $x_{j+1} = x_j - a_j u_j$ for all $j = 1, 2, \ldots, n$

Reformulated optimization problem:

$$\sum_{j=1}^{n} c_j u_j \rightarrow \max!$$

u.d.N.

$$
\begin{array}{lll}
x_{j+1} = x_j - a_j u_j, & & j = 1, 2, \ldots, n \\
x_1 = V & & \\
0 \leq x_{j+1} \leq V, & & j = 1, 2, \ldots, n \\
u_j \in \{0, 1\}, & \text{if } x_j \geq a_j, & j = 1, 2, \ldots, n \\
u_j = 0, & \text{if } x_j < a_j, & j = 1, 2, \ldots, n
\end{array}
$$

4.2 Problem Formulation

Dynamic programming problems consider a finite planning horizon, which is partitioned into n periods or stages.

State variable x_j:

$\rightarrow$ describes the state of the system at the beginning of period j (and at the end of period $j-1$, respectively)

$\rightarrow x_1 := x_a$ - given initial state of the system

Decision variable u_j:

$\rightarrow$ In period 1 the decision u_1 is made, which transforms die system into the state x_2, i.e.,

$$x_2 = f_1(x_1, u_1),$$

where, from the decision u_1, the cost $g_1(x_1, u_1)$ results.

in general:

$x_{j+1} = f_j(x_j, u_j)$	resultant *state*
$g_j(x_j, u_j)$	*stage cost*
$X_{j+1} \neq \emptyset$	*State region*, which contains possible states at the end of period j, where $X_1 = \{x_1\}$
$U_j(x_j) \neq \emptyset$	*Control region*, which contains possible decisions in period j (depends on state x_j at the beginning of period j)

Optimization problem:

$$\sum_{j=1}^{n} g_j(x_j, u_j) \ \to \min!$$

u.d.N.

$$\begin{aligned} & x_{j+1} = f_j(x_j, u_j), && j = 1, 2, \dots, n \\ & x_1 = x_a, && \\ & x_{j+1} \in X_{j+1}, && j = 1, 2, \dots, n \\ & u_j \in U_j(x_j), && j = 1, 2, \dots, n \end{aligned} \tag{13}$$

Remark: In general, the time complexity increases exponentially with the dimension of the state and decision variables

Definition 9 *A sequence of decisions* $(u_1, u_2, \dots, u_n)$ *is called* **policy** *or* **control**. *The sequence of decisions* $(x_1, x_2, \dots, x_n, x_{n+1})$ *corresponding to a given policy* $(u_1, u_2, \dots, u_n)$ *according to*

$$x_1 = x_a \text{ and } x_{j+1} = f_j(x_j, u_j) \qquad \text{for all } j = 1, 2, \dots, n$$

is called the corresponding **state sequence**. *A policy or state sequence satisfying the constraints (13) is called* **feasible**.

4.3 Bellman Equations and Bellman's Principle of Optimality

Given are g_j, f_j, X_{j+1} and U_j for all $j = 1, 2, \ldots, n$.

$\Rightarrow$ Optimization problem depends on x_1, i.e., $P_1(x_1)$.

analogously: $P_j(x_j)$ - problem for the periods $j, j+1, \ldots, n$ with the initial state x_j

Theorem 9 (Bellman's Principle of Optimality)
Let $(u_1^*, \ldots, u_j^*, \ldots, u_n^*)$ be an optimal policy for the problem $P_1(x_1)$ and x_j^* be the state at the beginning of period j, then $(u_j^*, \ldots, u_n^*)$ is an optimal policy for the problem $P_j(x_j^*)$, i.e.:
The decisions in the periods $j, \ldots, n$ of the n-period problem $P_1(x_1)$ are (for a given state x_j^*) independent of the decisions in the periods $1, \ldots, j-1$.

Bellman Equations:

1. Let $v_j^*(x_j)$ be the minimal cost for the problem $P_j(x_j)$. For $j = 1, 2, \ldots, n$, the relationships

$$\begin{aligned} v_j^*(x_j) &= g_j(x_j, u_j^*) + v_{j+1}^*(x_{j+1}^*) \\ &= \min_{u_j \in U_j(x_j)} \left\{ g_j(x_j, u_j) + v_{j+1}^*[f_j(x_j, u_j)] \right\} \\ & x_j \in X_j \end{aligned} \tag{14}$$

   are called the *Bellman equations* (BE), where

$$v_{n+1}^*(x_{n+1}) = 0$$

   for $x_{n+1} \in X_{n+1}$.

   $\Rightarrow$ Function v_j^* can be determined provided that v_{j+1}^* is known.

2. BE can also be determined for the following cases:

(a) $\sum_{i=1}^{n} g_j(x_j, u_j) \ \to \max!$

$\Rightarrow$ Replace in (14) min! by max!

(b) $\prod_{i=1}^{n} g_j(x_j, u_j) \ \to \min!$

$\Rightarrow$ BE:

$$v_j^*(x_j) = \min_{u_j \in U_j(x_j)} \left\{ g_j(x_j, u_j) \cdot v_{j+1}^* \big[f_j(x_j, u_j) \big] \right\}$$

where $v_{n+1}^*(x_{n+1}) := 1$ and

$g_j(x_j, u_j) > 0$ for all $x_j \in X_j, u_j \in U_j(x_j), j = 1, 2, \dots, n$

(c) $\max_{1 \le j \le n} \big\{ g_j(x_j, u_j) \big\} \ \to \min!$

$\Rightarrow$ BE:

$$v_j^*(x_j) = \min_{u_j \in U_j(x_j)} \left\{ \max\big\{ g_j(x_j, u_j); v_{j+1}^* \big[f_j(x_j, u_j) \big] \big\} \right\}$$

where $v_{n+1}^*(x_{n+1}) = 0$

4.4 Bellman Method

$\Rightarrow$ successive evaluation of (14) for $j = n, n-1, \dots, 1$ to determine $v_j^*(x_j)$

Algorithm DO

Phase 1: Backward Calculation

(a) Set $v_{n+1}^*(x_{n+1}) := 0$ for all $x_{n+1} \in X_{n+1}$.

(b) For $j = n, n-1, \dots, 1$ do:
For all $x_j \in X_j$, determine $z_j^*(x_j)$ as the minimum point of function

$$w_j(x_j, u_j) := g_j(x_j, u_j) + v_{j+1}^* \big[f_j(x_j, u_j) \big]$$

on $U_j(x_j)$, i.e.,

$$w_j(x_j, z_j^*(x_j)) = \min_{u_j \in U_j(x_j)} w_j(x_j, u_j) = v_j^*(x_j) \text{ for } x_j \in X_j$$

Phase 2: Forward Calculation

(a) Set $x_1^* := x_a$.

(b) For $j = 1, 2, \ldots, n$ do:

$$u_j^* := z_j^*(x_j),\ x_{j+1}^* := f_j(x_j^*, u_j^*)$$

$\Rightarrow$ $(u_1^*, u_2^*, \ldots, u_n^*)$ optimal policy

$\Rightarrow$ $(x_1^*, x_2^*, \ldots, x_{n+1}^*)$ optimal state sequence for problem $P_1(x_1^* = x_a)$

Summary: DP (Dynamic Programming)

Phase 1: *Decomposition*

Phase 2: *Backward calculation*

Phase 3: *Forward calculation*

Remark: If all equations

$$x_{j+1} = f_j(x_j, u_j), \quad j = 1, 2, \ldots, n$$

can be uniquely solved for x_j, one can also execute first a forward calculation and then a backward calculation (e.g. for the inventory problem from 4.1.).

4.5 Examples and Applications

(a) Knapsack Problem

Assumption: V, a_j, c_j - integer

$$g_j(x_j, u_j) = c_j u_j, \qquad j = 1, 2, \ldots, n$$

$$f_j(x_j, u_j) = x_j - a_j u_j, \qquad j = 1, 2, \ldots, n$$

$$X_{j+1} = \{0, 1, \ldots, V\}$$

$$U_j(x_j) = \begin{cases} \{0, 1\} & \text{for } x_j \geq a_j \\ 0 & \text{for } x_j < a_j \end{cases}, j = 1, 2, \ldots, n$$

BE:

$$v_j^*(x_j) = \max_{u_j \in U_j(x_j)} \big\{ c_j u_j + v_{j+1}^*(x_j - a_j u_j) \big\}, \quad 1 \le j \le n$$

Backward Calculation:

$$v_n^*(x_n) = \begin{cases} c_n, & \text{if } x_n \ge a_n \\ 0, & \text{otherwise} \end{cases}$$

$$z_n^*(x_n) = \begin{cases} 1, & \text{if } x_n \ge a_n \\ 0, & \text{otherwise} \end{cases}$$

For $j = n-1, n-2, \ldots, 1$:

$$v_j^*(x_j) = \begin{cases} \max\big\{ v_{j+1}^*(x_j);\ c_j + v_{j+1}^*(x_j - a_j) \big\}, & \text{if } x_j \ge a_j \\ v_{j+1}^*(x_j), & \text{otherwise} \end{cases}$$

$$z_j^*(x_j) = \begin{cases} 1, & \text{if } v_j^*(x_j) > v_{j+1}^*(x_j) \\ 0, & \text{otherwise} \end{cases}$$

$\Rightarrow v_1^*(V)$ - maximal value of the knapsack filling

Forward Calculation:

$$\begin{aligned} x_1^* &:= V \\ u_j^* &:= z_j^*(x_j^*), \qquad j = 1, 2, \ldots, n \\ x_{j+1}^* &:= x_j^* - a_j u_j^*, \quad j = 1, 2, \ldots, n \end{aligned}$$

(b) Determination of a Shortest (Longest) Path in a Graph

Goal: Determine a shortest path from vertex (city) x_1 to vertex (city) x_{n+1}.

Let:

$X_j = \{x_j^1, x_j^2, \ldots, x_j^k\}$ - set of all vertices of stage j, $\quad 2 \le j \le n$

$X_1 = \{x_1\}, \quad X_{n+1} = \{x_{n+1}\}$

$U_j(x_j) = \{x_{j+1} \in X_{j+1} \mid \exists \text{ a vertex from } x_j \text{ to } x_{j+1}\}, \quad j = 1, 2, \ldots, n$

$v_j^*(x_j)$ - length of a shortest path from vertex $x_j \in X_j$ to vertex x_{n+1}

$g_j(x_j, u_j) = c_{x_j,u_j}$

$f_{j+1}(x_j, u_j) = u_j = x_{j+1}$

$z_j^*(x_j) = u_j = x_{j+1}$ if x_{j+1} is the next vertex after vertex x_j on a shortest path from vertex x_j to vertex x_{n+1}

BE:

$$v_n^*(x_n) = c_{x_n,x_{n+1}} \quad \text{for} \quad x_n \in X_n$$

For $j = n-1, n-2, \ldots, 1$:

$$v_j^*(x_j) = \min\Big\{c_{x_j,x_{j+1}} + v_{j+1}^*(x_{j+1}) \mid x_{j+1} \in X_{j+1} \text{ such that the arc}(x_j, x_{j+1}) \text{ exists}\Big\}$$