CHAPTER I

Basic concepts of linear optimization

Overview

In 1827, the French mathematician JEAN-BAPTISTE JOSEPH FOURIER (1768–1830) published a method for solving systems of linear inequalities. This publication is usually seen as the first account on linear optimization. In 1930, the Russian mathematician LEONID V. KANTOROVICH (1912–1986) gave linear optimization formulations of resource allocation problems. Around the same time, the Dutch economist TALLING C. KOOPMANS (1910–1985) formulated linear optimization models for problems arising in classical, Walrasian (Léon Walras, 1834–1910), economics. In 1975, both KANTOROVICH and KOOPMANS received the Nobel Prize in economic sciences for their work. During World War II, linear optimization models were designed and solved for military planning problems. In 1947, GEORGE B. DANTZIG (1914–2005) invented, what he called, the simplex algorithm. The discovery of the simplex algorithm coincided with the rise of the computer, making it possible to computerize the calculations, and to use the method for solving large-scale real life problems. Since then, linear optimization has developed rapidly, both in theory and in application. At the end of the 1960’s, the first software packages appeared on the market. Nowadays linear optimization problems with millions of variables and constraints can readily be solved.

Linear optimization is presently used in almost all industrial and academic areas of quantitative decision making. For an extensive – but not exhaustive – list of fields of applications of linear optimization, we refer to Section 1.6 and the case studies in Chapters 10–11. Moreover, the theory behind linear optimization forms the basis for more advanced nonlinear optimization.

In this chapter, the basic concepts of linear optimization are discussed. We start with a simple example of a so-called linear optimization model (abbreviated to LO-model) containing two decision variables. An optimal solution of the model is determined by means of the ‘graphical method’. This simple example is used as a warming up exercise for more realistic cases, and the general form of an LO-model. We present a few LO-models that illustrate
the use of linear optimization, and that introduce some standard modeling techniques. We also describe how to use an online linear optimization package to solve an LO-model.

1.1 The company Dovetail

We start this chapter with a prototype problem that will be used throughout the book. The problem setting is as follows. The company Dovetail produces two kinds of matches: long and short ones. The company makes a profit of 3 (×$1,000) for every 100,000 boxes of long matches, and 2 (×$1,000) for every 100,000 boxes of short matches. The company has one machine that can produce both long and short matches, with a total of at most 9 (×100,000) boxes per year. For the production of matches the company needs wood and boxes: three cubic meters of wood are needed for 100,000 boxes of long matches, and one cubic meter of wood is needed for 100,000 boxes of short matches. The company has 18 cubic meters of wood available for the next year. Moreover, Dovetail has 7 (×100,000) boxes for long matches, and 6 (×100,000) for short matches available at its production site. The company wants to maximize its profit in the next year. It is assumed that Dovetail can sell any amount it produces.

1.1.1 Formulating Model Dovetail

In order to write the production problem that company Dovetail faces in mathematical terms, we introduce the decision variables $x_1$ and $x_2$:

\begin{align*}
x_1 &= \text{the number of boxes (×100,000) of long matches to be made the next year}, \\
x_2 &= \text{the number of boxes (×100,000) of short matches to be made the next year}.
\end{align*}

The company makes a profit of 3 (×$1,000) for every 100,000 boxes of long matches, which means that for $x_1$ (×100,000) boxes of long matches, the profit is $3x_1$ (×$1,000). Similarly, for $x_2$ (×100,000) boxes of short matches the profit is $2x_2$ (×$1,000). Since Dovetail aims at maximizing its profit, and it is assumed that Dovetail can sell its full production, the objective of Dovetail is:

\[ \text{maximize } 3x_1 + 2x_2. \]

The function $3x_1 + 2x_2$ is called the objective function of the problem. It is a function of the decision variables $x_1$ and $x_2$. If we only consider the objective function, it is obvious that the production of matches should be taken as high as possible. However, the company also has to take into account a number of constraints. First, the machine capacity is 9 (×100,000) boxes per year. This yields the constraint:

\[ x_1 + x_2 \leq 9. \quad (1.1) \]

Second, the limited amount of wood yields the constraint:

\[ 3x_1 + x_2 \leq 18. \quad (1.2) \]
Third, the numbers of available boxes for long and short matches is restricted, which means that $x_1$ and $x_2$ have to satisfy:

$$x_1 \leq 7,$$

and $x_2 \leq 6$.  

(1.3)  

The inequalities (1.1) – (1.4) are called *technology constraints*. Finally, we assume that only nonnegative amounts can be produced, i.e.,

$$x_1, x_2 \geq 0.$$  

The inequalities $x_1 \geq 0$ and $x_2 \geq 0$ are called *nonnegativity constraints*. Taking together the six expressions formulated above, we obtain Model Dovetail:

**Model Dovetail**

$$\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 \\
\text{s.t.} & \quad x_1 + x_2 \leq 9 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (1.1) \\
& \quad 3x_1 + x_2 \leq 18 \quad \quad \quad \quad (1.2) \\
& \quad x_1 \leq 7 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (1.3) \\
& \quad x_2 \leq 6 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (1.4) \\
& \quad x_1, x_2 \geq 0.
\end{align*}$$

In this model ‘s.t.’ means ‘subject to’. Model Dovetail is an example of a *linear optimization model*. We will abbreviate ‘linear optimization model’ as ‘LO-model’. The term ‘linear’ refers to the fact that the objective function and the constraints are linear functions of the decision variables $x_1$ and $x_2$. In the next section we will determine an *optimal solution* (also called *optimal point*) of Model Dovetail, which means that we will determine values of $x_1$ and $x_2$ satisfying the constraints of the model, and such that the value of the objective function is maximum for these values.

LO-models are often called ‘LP-models’, where ‘LP’ stands for linear programming. The word ‘programming’ in this context is an old-fashioned word for optimization, and has nothing to do with the modern meaning of programming (as in ‘computer programming’). We therefore prefer to use the word ‘optimization’ to avoid confusion.

### 1.1.2 The graphical solution method

Model Dovetail has two decision variables, which allows us to determine a solution graphically. To that end, we first draw the constraints in a rectangular coordinate system. We start with the nonnegativity constraints (see Figure 1.1). In Figure 1.1, the values of the decision variables $x_1$ and $x_2$ are nonnegative in the shaded area above the line $x_1 = 0$ and to the right of the line $x_2 = 0$. Next, we draw the line $x_1 + x_2 = 9$ corresponding to constraint (1.1) and determine on which side of this line the values of the decision variables satisfying
$x_1 + x_2 \leq 9$ are located. Figure 1.2 is obtained by doing this for all constraints. We end up with the region $0v_1, v_2, v_3, v_4$, which is called the feasible region of the model; it contains the points $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ that satisfy the constraints of the model. The points $0, v_1, v_2, v_3, v_4$ are called the vertices of the feasible region. It can easily be calculated that:

$v_1 = \begin{bmatrix} 6 \\ 0 \end{bmatrix}$, $v_2 = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$, $v_3 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$, and $v_4 = \begin{bmatrix} 0 \\ 6 \end{bmatrix}$.

In Figure 1.2, we also see that constraint (1.3) can be deleted without changing the feasible region. Such a constraint is called redundant with respect to the feasible region. On the other hand, there are reasons for keeping this constraint in the model. For example, when the right hand side of constraint (1.2) is sufficiently increased (thereby moving the line in Figure 1.2 corresponding to (1.3) to the right), constraint (1.3) becomes nonredundant again. See Chapter 5.

Next, we determine the points in the feasible region that attain the maximum value of the objective function. To that end, we draw in Figure 1.2 a number of so-called level lines. A level line is a line for which all points $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ on it have the same value of the objective function. In Figure 1.3, five level lines are drawn, namely $3x_1 + 2x_2 = 0, 6, 12, 18,$ and $24$. The arrows in Figure 1.3 point in the direction of increasing values of the objective function $3x_1 + 2x_2$. These arrows are in fact perpendicular to the level lines.

In order to find an optimal solution using Figure 1.3, we start with a level line corresponding to a small objective value (e.g., 6) and then (virtually) 'move' it in the direction of the arrows, so that the values of the objective function increase. We stop moving the level line when it reaches the boundary of the feasible region, so that moving the level line any further would mean that no point of it would lie in the region $0v_1, v_2, v_3, v_4$. This happens for the level line $3x_1 + 2x_2 = 4\frac{1}{2}$. This level line intersects the feasible region at exactly one point, namely $\begin{bmatrix} 4\frac{1}{2} \\ 4\frac{1}{2} \end{bmatrix}$. Hence, the optimal solution is $x_1^* = 4\frac{1}{2}, x_2^* = 4\frac{1}{2}$, and the optimal objective value
is \(22\frac{1}{2}\). Note that the optimal point is a vertex of the feasible region. This fact plays a crucial role in linear optimization; see Section 2.1.2. Also note that this is the only optimal point.

In Figure 1.4, the same model is depicted in three-dimensional space. The values of \(z = 3x_1 + 2x_2\) on the region \(0v_1v_2v_3v_4\) form the region \(0v_1'v_2'v_3'v_4'\). From Figure 1.4 it is obvious that the point \(v_2\) with coordinate values \(x_1 = 4\frac{1}{2}\) and \(x_2 = 4\frac{1}{2}\) is the optimal solution. At \(v_2\), the value of the objective function is \(z^* = 22\frac{1}{2}\), which means that the maximum profit is $22,500. This profit is achieved by producing 450,000 boxes of long matches and 450,000 boxes of short matches.

### 1.2 Definition of an LO-model

In this section we define the standard form of an LO-model. Actually, the literature on linear optimization contains a wide range of definitions. They all consist of an objective together with a number of constraints. All such standard forms are equivalent; see Section 1.3.

#### 1.2.1 The standard form of an LO-model

We first formulate the various parts of a general LO-model. It is always assumed that for any positive integer \(d\), \(\mathbb{R}^d\) is a Euclidean vector space of dimension \(d\); see Appendix B. Any LO-model consists of the following four parts:

- **Decision variables.** An LO-model contains real-valued decision variables, denoted by \(x_1, \ldots, x_n\), where \(n\) is a finite positive integer. In Model Dovetail the decision variables are \(x_1\) and \(x_2\).

- **Objective function.** The objective function \(c_1x_1 + \ldots + c_nx_n\) is a linear function of the \(n\) decision variables. The constants \(c_1, \ldots, c_n\) are real numbers that are called the _objective coefficients_. Depending on whether the objective function has to be maximized
or minimized, the objective of the model is written as:

\[ \max c_1x_1 + \ldots + c_nx_n, \quad \text{or} \quad \min c_1x_1 + \ldots + c_nx_n, \]

respectively. In the case of Model Dovetail the objective is \( \max 3x_1 + 2x_2 \) and the objective function is \( 3x_1 + 2x_2 \). The value of the objective function at a point \( \mathbf{x} \) is called the objective value at \( \mathbf{x} \).

**Technology constraints.** A technology constraint of an LO-model is either a ‘\( \leq \)’, a ‘\( \geq \)’, or an ‘\( = \)’ expression of the form:

\[ a_{i1}x_1 + \ldots + a_{in}x_n \ (\leq, \geq, =) \ b_i, \]

where \( (\leq, \geq, =) \) means that either the sign ‘\( \leq \)’, or ‘\( \geq \)’, or ‘\( = \)’ holds. The entry \( a_{ij} \) is the coefficient of the \( j \)th decision variable \( x_j \) in the \( i \)th technology constraint. Let \( m \) be the number of technology constraints. All (left hand sides of the) technology constraints are linear functions of the decision variables \( x_1, \ldots, x_n \).

**Nonnegativity and nonpositivity constraints.** A nonnegativity constraint of an LO-model is an inequality of the form \( x_i \geq 0 \); similarly, a nonpositivity constraint is of the form \( x_i \leq 0 \). It may also happen that a variable \( x_i \) is not restricted by a nonnegativity constraint or a nonpositivity constraint. In that case, we say that \( x_i \) is a free or unrestricted variable. Although nonnegativity and nonpositivity constraints can be written in the form of a technology constraint, we will usually write them down separately.

For \( i \in \{1, \ldots, m\} \) and \( j \in \{1, \ldots, n\} \), the real-valued entries \( a_{ij}, b_i, \) and \( c_j \) are called the parameters of the model. The technology constraints, nonnegativity and nonpositivity constraints together are referred to as the constraints (or restrictions) of the model.

A vector \( \mathbf{x} \in \mathbb{R}^n \) that satisfies all constraints is called a feasible point or feasible solution of the model. The set of all feasible points is called the feasible region of the model. An LO-model is called feasible if its feasible region is nonempty; otherwise, it is called infeasible. An optimal solution of a maximizing (minimizing) LO-model is a point in the feasible region with maximum (minimum) objective value, i.e., a point such that there is no other point with a larger (smaller) objective value. Note that there may be more than one optimal solution, or none at all. The objective value at an optimal solution is called the optimal objective value.

Let \( \mathbf{x} \in \mathbb{R}^n \). A constraint is called binding at the point \( \mathbf{x} \) if it holds with equality at \( \mathbf{x} \). For example, in Figure 1.2, the constraints (1.1) and (1.1) are binding at the point \( \mathbf{v}_2 \), and the other constraints are not binding. A constraint is called violated at the point \( \mathbf{x} \) if it does not hold at \( \mathbf{x} \). So, if one or more constraints are violated at \( \mathbf{x} \), then \( \mathbf{x} \) does not lie in the feasible region.
A maximizing LO-model with only ‘≤’ technology constraints and nonnegativity constraints can be written as follows:

\[
\begin{align*}
\text{max} & \quad c_1 x_1 + \ldots + c_n x_n \\
\text{s.t.} & \quad a_{11} x_1 + \ldots + a_{1n} x_n \leq b_1 \\
& \quad \vdots \\
& \quad a_{m1} x_1 + \ldots + a_{mn} x_n \leq b_m \\
& \quad x_1, \ldots, x_n \geq 0.
\end{align*}
\]

Using the summation sign ‘\(\sum\)’, this can also be written as:

\[
\text{max} \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, \ldots, m \\
x_1, \ldots, x_n \geq 0.
\]

In terms of matrices there is an even shorter notation. The superscript ‘\(T\)’ transposes a row vector into a column vector, and an \((m,n)\) matrix into an \((n,m)\) matrix \((m,n \geq 1)\). Let

\[
\mathbf{c} = [c_1 \ldots c_n]^T \in \mathbb{R}^n, \quad \mathbf{b} = [b_1 \ldots b_m]^T \in \mathbb{R}^m, \\
\mathbf{x} = [x_1 \ldots x_n]^T \in \mathbb{R}^n, \quad \text{and } \mathbf{A} = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}.
\]

The matrix \(\mathbf{A}\) is called the technology matrix (or coefficients matrix), \(\mathbf{c}\) is the objective vector, and \(\mathbf{b}\) is the right hand side vector of the model. The LO-model can now be written as:

\[
\text{max} \{\mathbf{c}^T \mathbf{x} \mid \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\},
\]

where \(\mathbf{0} \in \mathbb{R}^n\) is the \(n\)-dimensional all-zero vector. We call this form the standard form of an LO-model (see also Section 1.3). It is a maximizing model with ‘≤’ technology constraints, and nonnegativity constraints. The feasible region \(\mathcal{F}\) of the standard LO-model satisfies:

\[
\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0\}.
\]

In the case of Model Dovetail, we have that:

\[
\mathbf{c} = [3 \ 2]^T, \quad \mathbf{b} = [9 \ 18 \ 7 \ 6]^T, \quad \mathbf{x} = [x_1 \ x_2]^T, \quad \text{and } \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.
\]
Figure 1.5: The feasible region of Model Dovetail*.

So Model Dovetail in standard form reads:

\[
\begin{align*}
\text{max} \ & \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
\text{s.t.} \ & 1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 9 \\ 7 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{align*}
\]

Suppose that we want to add the following additional constraint to Model Dovetail. The manager of Dovetail has an agreement with retailers to deliver a total of at least 500,000 boxes of matches next year. Using our decision variables, this yields the new constraint:

\[
x_1 + x_2 \geq 5
\]  

(1.5)

Instead of a ‘\(\leq\)’ sign, this inequality contains a ‘\(\geq\)’ sign. With this additional constraint, Figure 1.3 changes into Figure 1.5. From Figure 1.5 one can graphically derive that the optimal solution \((x_1^* = x_2^* = 4 \frac{1}{2})\) is not affected by adding the new constraint (1.5).

Including constraint (1.5) in Model Dovetail, yields Model Dovetail*.

\[
\begin{align*}
\text{Model Dovetail}^*: \\
\text{max} \ & \ 3x_1 + 2x_2 \\
\text{s.t.} \ & x_1 + x_2 \leq 9 \quad (1.1) \\
\ & 3x_1 + x_2 \leq 18 \quad (1.2) \\
\ & x_1 \leq 7 \quad (1.3) \\
\ & x_2 \leq 6 \quad (1.4) \\
\ & x_1 + x_2 \geq 5 \quad (1.5) \\
\ & x_1, x_2 \geq 0.
\end{align*}
\]

In order to write this model in the standard form, the ‘\(\geq\)’ constraint has to be transformed into a ‘\(\leq\)’ constraint. This can be done by multiplying both sides of it by \(-1\). Hence, \(x_1 + x_2 \geq 5\) then becomes \(-x_1 - x_2 \leq -5\). Therefore, the standard form of Model
Dovetail* is:

$$\max \left\{ \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \middle| \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 9 \\ 18 \\ 7 \\ 6 \\ -5 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}.$$ 

Similarly, a constraint with ‘=’ can be put into standard form by replacing it by two ‘≤’ constraints. For instance, $3x_1 - 8x_2 = 11$ can be replaced by $3x_1 - 8x_2 \leq 11$ and $-3x_1 + 8x_2 \leq -11$. Also, the minimizing LO-model $\min \{ c^T x \mid Ax \leq b, x \geq 0 \}$ can be written in standard form, since

$$\min \{ c^T x \mid Ax \leq b, x \geq 0 \} = -\max \{ -c^T x \mid Ax \leq b, x \geq 0 \};$$

see also Section 1.3.

1.2.2 Slack variables and binding constraints

Model Dovetail in Section 1.1 contains the following machine capacity constraint:

$$x_1 + x_2 \leq 9. \quad (1.1)$$

This constraint expresses the fact that the machine can produce at most 9 ($\times 100,000$) boxes per year. We may wonder whether there is excess machine capacity (overcapacity) in the case of the optimal solution. For that purpose, we introduce an additional nonnegative variable $x_3$ in the following way:

$$x_1 + x_2 + x_3 = 9.$$ 

The variable $x_3$ is called the slack variable of constraint (1.1). Its optimal value, called the slack, measures the unused capacity of the machine. By requiring that $x_3$ is nonnegative, we can avoid the situation that $x_1 + x_2 > 9$, which would mean that the machine capacity is exceeded and the constraint $x_1 + x_2 \leq 9$ is violated. If, at the optimal solution, the value of $x_3$ is zero, then the machine capacity is completely used. In that case, the constraint is binding at the optimal solution.

Introducing slack variables for all constraints of Model Dovetail, we obtain the following model:
Model Dovetail with slack variables.

\[
\begin{align*}
\text{max} & \quad 3x_1 + 2x_2 \\
\text{s.t.} & \quad x_1 + x_2 + x_3 = 9 \\
& \quad 3x_1 + x_2 + x_4 = 18 \\
& \quad x_1 + x_5 = 7 \\
& \quad x_2 + x_6 = 6 \\
& \quad x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{align*}
\]

In this model, \(x_3, x_4, x_5,\) and \(x_6\) are the nonnegative slack variables of the constraints (1.1), (1.2), (1.3), and (1.4), respectively. The number of slack variables is therefore equal to the number of inequality constraints of the model. In matrix notation the model becomes:

\[
\begin{align*}
\text{max} \quad & \begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
\text{s.t.} \quad & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 9 \\ 18 \\ 7 \\ 6 \end{bmatrix},
\end{align*}
\]

If \(I_m\) denotes the identity matrix with \(m\) rows and \(m\) columns (\(m \geq 1\)), then the general form of an LO-model with slack variables can be written as:

\[
\begin{align*}
\text{max} \quad & \begin{bmatrix} c^T \end{bmatrix} \begin{bmatrix} x \end{bmatrix} \\
\text{s.t.} \quad & \begin{bmatrix} A I_m \end{bmatrix} \begin{bmatrix} x \\ x_s \end{bmatrix} = \begin{bmatrix} b \end{bmatrix},
\end{align*}
\]

with \(x \in \mathbb{R}^n, x_s \in \mathbb{R}^m, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n},\) and \(I_m \in \mathbb{R}^{m \times m}.\) Note that the value of \(x_s\) (the vector of slack variables) satisfies \(x_s = b - Ax\) and is therefore completely determined by the value of \(x.\) This means that if the values of the entries of the vector \(x\) are given, then the values of the entries of \(x_s\) are fixed.

1.2.3 Types of optimal solutions and feasible regions

It may happen that an LO-model has more than one optimal solution. For instance, if we replace the objective of Model Dovetail by

\[
\text{max} \quad x_1 + x_2,
\]

then all points on the line segment \(v_2v_3\) (see Figure 1.2) have the same optimal objective value, namely 9, and therefore all points on the line segment \(v_2v_3\) are optimal. In this case, we say that there are multiple optimal solutions; see also Section 3.7 and Section 5.6.1. The feasible region has two optimal vertices, namely \(v_2\) and \(v_3.\)

Three types of feasible regions can be distinguished, namely:
1.2. Definition of an LO-model

Feasible region bounded and nonempty. A feasible region is called bounded if all decision variables are bounded on the feasible region (i.e., no decision variable can take on arbitrarily large values on the feasible region). An example is drawn in Figure 1.6(a). If the feasible region is bounded, then the objective values are also bounded on the feasible region and hence an optimal solution exists. Note that the feasible region of Model Dovetail is bounded; see Figure 1.2.

Feasible region unbounded. A nonempty feasible region is called unbounded if it is not bounded; i.e., at least one of the decision variables can take on arbitrarily large values on the feasible region. Examples of an unbounded feasible region are shown in Figure 1.6(b) and Figure 1.6(c). Whether an optimal solution exists depends on the objective function. For example, in the case of Figure 1.6(b) an optimal solution does not exist. Indeed, the objective function takes on arbitrarily large values on the feasible region. Therefore, the model has no optimal solution. An LO-model with an objective function that takes on arbitrarily large values is called unbounded; it is called bounded otherwise. On the other hand, in Figure 1.6(c), an optimal solution does exist. Hence, this is an example of an LO-model with an unbounded feasible region, but with a (unique) optimal solution.

Feasible region empty. In this case we have that \( F = \emptyset \) and the LO-model is called infeasible. For example, if an LO-model contains the (contradictory) constraints \( x_1 \geq 6 \) and \( x_1 \leq 3 \), then its feasible region is empty. If an LO-model is infeasible, then it has no feasible points and, in particular, no optimal solution. If \( F \neq \emptyset \), then the LO-model is called feasible.

So, an LO-model either has an optimal solution, or it is infeasible, or it is unbounded. Note that an unbounded LO-model necessarily has an unbounded feasible region, but the converse is not true. In fact, Figure 1.6(c) shows an LO-model that is bounded, although it has an unbounded feasible region.
1.3 Alternatives of the standard LO-model

In the previous sections we mainly discussed LO-models of the form

$$\max \{ c^T x \mid Ax \leq b, x \geq 0 \},$$

with $A \in \mathbb{R}^{m \times n}$. We call this form the standard form of an LO-model. The standard form is characterized by a maximizing objective, ‘$\leq$’ technology constraints, and nonnegativity constraints.

In general, many different forms may be encountered, for instance with both ‘$\geq$’ and ‘$\leq$’ technology constraints, and both nonnegativity ($x_i \geq 0$) and nonpositivity constraints ($x_i \leq 0$). All these forms can be reduced to the standard form $\max \{ c^T x \mid Ax \leq b, x \geq 0 \}$.

The following rules can be applied to transform a nonstandard LO-model into a standard model:

- A minimizing model is transformed into a maximizing model by using the fact that minimizing a function is equivalent to maximizing minus that function. So, the objective of the form ‘$\min c^T x$’ is equivalent to the objective ‘$-\max (-c)^T x$’. For example, ‘$\min x_1 + x_2$’ is equivalent to ‘$-\max -x_1 - x_2$’.

- A ‘$\geq$’ constraint is transformed into a ‘$\leq$’ constraint by multiplying both sides of the inequality by $-1$ and reversing the inequality sign. For example, $x_1 - 3x_2 \geq 5$ is equivalent to $-x_1 + 3x_2 \leq -5$.

- A ‘$=$’ constraint of the form ‘$a^T x = b$’ can be written as ‘$a^T x \leq b$ and $a^T x \geq b$’. The second inequality in this expression is then transformed into a ‘$\leq$’ constraint (see the previous item). For example, the constraint ‘$2x_1 + x_2 = 3$’ is equivalent to ‘$2x_1 + x_2 \leq 3$ and $-2x_1 - x_2 \leq -3$’.

- A nonpositivity constraint is transformed into a nonnegativity constraint by replacing the corresponding variable by its negative. For example, the nonpositivity constraint ‘$x_1 \leq 0$’ is transformed into ‘$x'_1 \geq 0$’ by substituting $x_1 = -x'_1$.

- A free variable is replaced by the difference of two new nonnegative variables. For example, the expression ‘$x_1$ free’ is replaced by ‘$x'_1 \geq 0$, $x''_1 \geq 0$’, and substituting $x_1 = x'_1 - x''_1$.

The following two examples illustrate these rules.

**Example 1.3.1.** Consider the nonstandard LO-model:

$$\begin{align*}
\min & \quad -5x_1 + 3x_2 \\
\text{s.t.} & \quad x_1 + 2x_2 \geq 10 \\
& \quad x_1 - x_2 \leq 6 \\
& \quad x_1 + x_2 = 12 \\
& \quad x_1, x_2 \geq 0.
\end{align*}$$ (i.7)
In addition to being a minimizing model, the model has a ‘≥’ constraint and a ‘=’ constraint. By applying the above rules, the following equivalent standard form LO-model is found:

\[
\begin{align*}
- \max & \quad 5x_1 - 3x_2 \\
\text{s.t.} & \quad -x_1 - 2x_2 \leq -10 \\
& \quad x_1 - x_2 \leq 6 \\
& \quad x_1 + x_2 \leq 12 \\
& \quad -x_1 - x_2 \leq -12 \\
& \quad x_1, x_2 \geq 0.
\end{align*}
\]

The (unique) optimal solution of this model reads: \(x_1^* = 9, x_2^* = 3\). This is also the optimal solution of the nonstandard model (1.7).

**Example 1.3.2.** Consider the nonstandard LO-model:

\[
\begin{align*}
\max & \quad 5x_1 + x_2 \\
\text{s.t.} & \quad 4x_1 + 2x_2 \leq 8 \\
& \quad x_1 + x_2 \leq 12 \\
& \quad x_1 - x_2 \leq 5 \\
& \quad \geq 0, x_2 \text{ free}.
\end{align*}
\]

This model has a maximizing objective, and only ‘≤’ constraints. However, the variable \(x_2\) is unrestricted in sign. Applying the above rules, the following equivalent standard form LO-model is found:

\[
\begin{align*}
\max & \quad 5x_1 + x_2' - x_2'' \\
\text{s.t.} & \quad 4x_1 + 2x_2' - 2x_2'' \leq 8 \\
& \quad x_1 + x_2' - x_2'' \leq 12 \\
& \quad x_1 - x_2' + x_2'' \leq 5 \\
& \quad x_1, x_2', x_2'' \geq 0.
\end{align*}
\]

The point \(\hat{x} = [x_1^* (x_2')^* (x_2'')^*]^T = [3 \ 0 \ 2]^T\) is an optimal solution of this model. Hence, the point

\[
\mathbf{x}^* = \left[\begin{array}{c}
x_1^* \\
x_2^*
\end{array}\right] = \left[\begin{array}{c}
x_1^* \\
(x_2')^* - (x_2'')^*
\end{array}\right] = \left[\begin{array}{c}
3 \\
-2
\end{array}\right]
\]

is an optimal solution of model (1.8). Note that \(\hat{\mathbf{x}}' = [3 \ 10 \ 12]^T\) is another optimal solution of (1.9) (why?), corresponding to the same optimal solution \(\mathbf{x}^*\) of model (1.8). In fact, the reader may verify that every point in the set

\[
\left\{ \left[\begin{array}{c}
3 \\
2 + \alpha
\end{array}\right] \in \mathbb{R}^3 \bigg| \alpha \geq 0 \right\}
\]

is an optimal solution of (1.9) that corresponds to the optimal solution \(\mathbf{x}^*\) of model (1.8).
We have listed six possible general nonstandard models below. Any method for solving one of the models (i)–(vi) can be used to solve the others, because they are all equivalent. The matrix $A$ in (iii) and (vi) is assumed to be of full row rank (i.e., $\text{rank}(A) = m$; see Appendix B and Section 3.8). The alternative formulations are:

(i) $\max \{c^T x \mid Ax \leq b, x \geq 0\}$;  
(iv) $\min \{c^T x \mid Ax \geq b, x \geq 0\}$;  

(standard form)  

(standard dual form; see Chapter 4)

(ii) $\max \{c^T x \mid Ax \leq b\}$;  
(v) $\min \{c^T x \mid Ax \geq b\}$;  

(iii) $\max \{c^T x \mid Ax = b, x \geq 0\}$;  
(vi) $\min \{c^T x \mid Ax = b, x \geq 0\}$.

Formulation (i) can be reduced to (ii) by writing:

$$\max \{c^T x \mid Ax \leq b, x \geq 0\} = \max \left\{ c^T x \left| \begin{bmatrix} A \\ -I_n \end{bmatrix} x \leq \begin{bmatrix} b \\ 0 \end{bmatrix} \right. \right\}.$$

The notation $\begin{bmatrix} A \\ -I_n \end{bmatrix}$ and $\begin{bmatrix} b \\ 0 \end{bmatrix}$ is explained in Appendix B. Formulation (ii) can be reduced to (i) as follows. Each vector $x$ can be written as $x = x' - x''$ with $x', x'' \geq 0$. Hence,

$$\max \{c^T x \mid Ax \leq b\} = \max \left\{ c^T (x' - x'') \mid A(x' - x'') \leq b, x', x'' \geq 0 \right\} = \max \left\{ \left[ c^T - c^T \right] \begin{bmatrix} x' \\ x'' \end{bmatrix} \mid \begin{bmatrix} A \\ A \end{bmatrix} \begin{bmatrix} x' \\ x'' \end{bmatrix} \leq b, x' \geq 0, x'' \geq 0 \right\},$$

and this has the form (i). The reduction of (i) to (iii) follows by introducing slack variables in (i). Formulation (iii) can be reduced to (i) by noticing that the constraints $Ax = b$ can be written as the two constraints $Ax \leq b$ and $Ax \geq b$. Multiplying the former by $-1$ on both sides yields $-Ax \leq -b$. Therefore, (iii) is equivalent to:

$$\max \left\{ c^T x \left| \begin{bmatrix} A \\ -A \end{bmatrix} x \leq \begin{bmatrix} b \\ -b \end{bmatrix} \right. \right\}.$$

The disadvantage of this transformation is that the model becomes considerably larger. In Section 3.8, we will see an alternative, more economical, reduction of (iii) to the standard form. Similarly, (iv), (v), and (vi) are equivalent. Finally, (iii) and (vi) are equivalent because

$$\min \{c^T x \mid Ax = b, x \geq 0\} = -\max \{-c^T x \mid Ax = b, x \geq 0\}.$$

### 1.4 Solving LO-models using a computer package

Throughout this book, we will illustrate the ideas in each chapter by means of computer examples. There are numerous computer packages for solving linear optimization models. Roughly two types of linear optimization packages can be distinguished:

- **Linear optimization solvers.** A linear optimization solver takes input values for the cost vector, the technology matrix, and the right hand side vector of the LO-model that
needs to be solved. The purpose of the package is to find an optimal solution to that LO-model. As described in Section 1.2.3, it is not always possible to find an optimal solution because the LO-model may be infeasible or unbounded. Another reason why a solver might fail is because there may be a limit on the amount of time that is used to find a solution, or because of round-off errors due to the fact that computer algorithms generally do calculations with limited precision using so-called floating-point numbers (see also the discussion in Section 3.6.1). Examples of linear optimization solvers are CPLEX and the linprog function in MATLAB.

Linear optimization modeling languages. Although any LO-model can be cast in a form that may serve as an input for a linear optimization solver (see Section 1.3), it is often times rather tedious to write out the full cost vector, the technology matrix, and the right hand side vector. This is, for example, the case when we have variables \( x_1, \ldots, x_{100} \) and we want the objective of the LO-model to maximize \( \sum_{i=1}^{100} x_i \). Instead of having to type a hundred times the entry 1 in a vector, we would prefer to just tell the computer to take the sum of these variables. For this purpose, there are a few programming languages available that allow the user to write LO-models in a more compact way. The purpose of the linear optimization programming package is to construct the cost vector, the technology matrix, and the right hand side vector from an LO-model written in that language. Examples of such linear optimization programming languages are GNU MathProg (also known as GMPL), AMPL, GAMS, and AIMMS.

We will demonstrate the usage of a linear optimization package by using the online solver provided on the website of this book. The online solver is able to solve models that are written in the GNU MathProg language. To solve Model Dovetail using the online solver, the following steps need to be taken:

- Start the online solver from the book website: http://www.lio.yoriz.co.uk/.
- In the editor, type the following code (without the line numbers):

```plaintext
1 var x1 >= 0;
2 var x2 >= 0;
3 maximize z: 3*x1 + 2*x2;
4 subject to c11: x1 + x2 <= 9;
5 subject to c12: 3*x1 + x2 <= 18;
6 subject to c13: x1 <= 7;
7 subject to c14: x2 <= 6;
8 end;
```

Listing 1.1: Model Dovetail as a GNU MathProg model.

This is the representation of Model Dovetail in the MathProg language. Since Model Dovetail is a relatively simple model, its representation is straightforward. For more details on how to use the MathProg language, see Appendix F.
Press the ‘Solve’ button to solve the model. In this step, a few things happen. The program first transforms the model from the previous step into a cost vector, technology matrix, and right hand side vector in some standard form of the LO-model (this standard form depends on the solver and need not correspond to the standard form we use in this book). Then, the program takes this standard form model and solves it. Finally, the solution of the standard-form LO-model is translated back into the language of the original model.

Press the ‘Solution’ button to view the solution. Among many things that might sound unfamiliar for now, the message states that the solver found an optimal solution with objective value 22.5. Also, the optimal values of \( x_1 \) and \( x_2 \) are listed. As should be expected, they coincide with the solution we found earlier in Section 1.1.2 by applying the graphical solution method.

I.5 Linearizing nonlinear functions

The definition of an LO-model states that the objective function and the (left hand sides of the) constraints have to be linear functions of the decision variables. In some cases, however, it is possible to rewrite a model with nonlinearities so that the result is an LO-model. We describe a few of them in this section.

### 1.5.1 Ratio constraints

Suppose that the company Dovetail wants to make at least 75% of its profit from long matches. At the optimal solution in the current formulation (Model Dovetail), the profit from long matches is \( (4\frac{1}{2} \times 3 = 13\frac{1}{2} \times $1,000) \) and the profit from short matches is \( (4\frac{1}{2} \times 2 = 9 \times $1,000) \). Thus, currently, the company gets \( (13\frac{1}{2}/22\frac{1}{2} = 60\%) \) of its profit from long matches. In order to ensure that at least 75% of the profit comes from the production of long matches, we need to add the following constraint:

\[
\frac{3x_1}{3x_1 + 2x_2} \geq \frac{3}{4}. \tag{1.10}
\]

The left hand side of this constraint is clearly nonlinear, and so the constraint – in its current form – cannot be used in an LO-model. However, multiplying both sides by \( 3x_1 + 2x_2 \) (and using the fact that \( x_1 \geq 0 \) and \( x_2 \geq 0 \)), we obtain the formulation:

\[
3x_1 \geq \frac{3}{4}(3x_1 + 2x_2), \quad \text{or, equivalently,} \quad \frac{3}{4}x_1 - \frac{3}{2}x_2 \geq 0,
\]

which is linear. Note that the left hand side of (1.10) is not defined if \( x_1 \) and \( x_2 \) both have value zero; the expression \( \frac{3}{4}x_1 - \frac{3}{2}x_2 \geq 0 \) does not have this problem. Adding the constraint to Model Dovetail yields the optimal solution \( x_1^* = 5\frac{1}{4}, x_2^* = 2\frac{4}{7} \) with corresponding objective value 20\( \frac{4}{7} \). The profit from long matches is \( 15\frac{3}{7} \times $1,000 \), which is exactly 75% of the total profit, as required.
1.5.2 Objective functions with absolute value terms

Consider the following minimization model in which we have an absolute value in the objective function:

\[
\begin{align*}
\min & \quad |x_1| + x_2 \\
\text{s.t.} & \quad 3x_1 + 2x_2 \geq 1 \\
& \quad x_1 \text{ free}, x_2 \geq 0.
\end{align*}
\] (1.11)

The model is not an LO-model, because of the absolute value in the objective function, which makes the objective function nonlinear. Model (1.11) can be written as an LO-model by introducing two new nonnegative variables \(u_1\) and \(u_2\) that will have the following relationship with \(x_1\):

\[
u_1 = \begin{cases} x_1 & \text{if } x_1 \geq 0 \\ 0 & \text{otherwise}, \end{cases} \quad \text{and} \quad u_2 = \begin{cases} -x_1 & \text{if } x_1 \leq 0 \\ 0 & \text{otherwise}. \end{cases}
\] (1.12)

A more compact way of writing (1.12) is:

\[
u_1 = \max\{0, x_1\} \quad \text{and} \quad u_2 = \max\{0, -x_1\}.
\]

Note that we have that \(x_1 = u_1 - u_2\) and \(|x_1| = u_1 + u_2\). Because \(u_1\) and \(u_2\) should never be simultaneously nonzero, they need to satisfy the relationship \(u_1 u_2 = 0\). By adding the constraint \(u_1 u_2 = 0\) and substituting \(u_1 - u_2\) for \(x_1\) and \(u_1 + u_2\) for \(|x_1|\) in (1.11), we obtain the following optimization model:

\[
\begin{align*}
\min & \quad u_1 + u_2 + x_2 \\
\text{s.t.} & \quad 3u_1 - 3u_2 + 2x_2 \geq 1 \\
& \quad u_1 u_2 = 0 \\
& \quad x_2, x_3, u_2 \geq 0.
\end{align*}
\] (1.13)

This model is still nonlinear. However, the constraint \(u_1 u_2 = 0\) can be left out. That is, we claim that it suffices to solve the following optimization model (which is an LO-model):

\[
\begin{align*}
\min & \quad u_1 + u_2 + x_2 \\
\text{s.t.} & \quad 3u_1 - 3u_2 + 2x_2 \geq 1 \\
& \quad u_1, u_2, x_2 \geq 0.
\end{align*}
\] (1.14)

To see that the constraint \(u_1 u_2 = 0\) may be left out, we will show that it is automatically satisfied at any optimal solution of (1.14). Let \(x^* = [u_1^* \ u_2^* \ x_2^*]^T\) be an optimal solution of (1.14). Suppose for a contradiction that \(u_1^* u_2^* \neq 0\). This implies that both \(u_1^* > 0\) and \(u_2^* > 0\). Let \(\varepsilon = \min\{u_1^*, u_2^*\} > 0\), and consider \(\bar{x} = [\hat{u}_1 \ \hat{u}_2 \ \hat{x}_2]^T = [u_1^* - \varepsilon \ u_2^* - \varepsilon \ x_2^*]^T\). It is easy to verify that \(\bar{x}\) a feasible solution of (1.14), and that the corresponding objective value \(z\) satisfies \(z = u_1^* - \varepsilon + u_2^* - \varepsilon + x_2^* < u_1^* + u_2^* + x_2^*\). Thus, we have constructed a feasible solution \(\bar{x}\) of (1.14), the objective value of which is smaller than the objective value of \(x^*\), contradicting the fact that \(x^*\) is an optimal solution of (1.14). Hence, the constraint \(u_1 u_2 = 0\) is automatically satisfied by any optimal solution of (1.14) and, hence, any optimal solution of (1.14) is also an optimal solution of (1.13). We leave it to the reader to show that
Figure 1.7: Piecewise linear function $f(x_1)$.

If $\mathbf{x}^* = [u_1^* \ u_2^* \ x_2^*]^T$ is an optimal solution of (1.14), then $[x_1^* \ x_2^*]^T$ with $x_1^* = u_1^* - u_2^*$ is an optimal solution of (1.14); see Exercise 1.8.9.

Model (1.14) can be solved using the graphical method as follows. Recall that at least one of $u_1$ and $u_2$ has to have value zero in the optimal solution. So we can distinguish two cases: either $u_1 = 0$, or $u_2 = 0$. These two cases give rise to two different LO-models:

\[
\begin{align*}
\min & \quad u_2 + x_2 \\
\text{s.t.} & \quad -3u_2 + 2x_2 \geq 1 \\
              & \quad u_2, x_2 \geq 0,
\end{align*}
\]

and

\[
\begin{align*}
\min & \quad u_1 + x_2 \\
\text{s.t.} & \quad 3u_1 + 2x_2 \geq 1 \\
              & \quad u_1, x_2 \geq 0.
\end{align*}
\]

(1.15) Since both LO-models have two decision variables, they can be solved using the graphical method. The optimal solutions are $[0 \ 1]^T$ (with optimal objective value $\frac{1}{2}$) and $[\frac{1}{3} \ 0]^T$ (with optimal objective value $\frac{1}{3}$), respectively. The optimal solution of (1.14) is found by choosing the solution among these two that has the smallest objective value. This gives $u_1^* = \frac{1}{3}$, $u_2^* = 0$, and $x_2^* = 0$, with optimal objective value $\frac{1}{3}$. The corresponding optimal solution of (1.11) satisfies $x_1^* = u_1^* - u_2^* = \frac{1}{3}$ and $x_2^* = 0$.

It is important to realize that it is not true that every feasible solution of (1.13) corresponds to a feasible solution of (1.11). For example, the vector $[u_1 \ u_2 \ x_2]^T = [2 \ 1 \ 0]^T$ is a feasible solution of (1.13) with objective value 3. However, the corresponding vector $[x_1 \ x_2]^T = [1 \ 0]^T$ in (1.11) has objective value 1. The reason for this mismatch is the fact that $u_1$ and $u_2$ are simultaneously nonzero. Recall that this never happens at an optimal feasible solution of (1.13).

This method also works for a maximizing objective in which the absolute value appears with a negative coefficient. However, it does not work for a maximizing objective in which the absolute value appears in the objective function with a positive coefficient. The reader is asked to verify this in Exercise 1.8.10.
I.5.3 Convex piecewise linear functions

The example from the previous section can be generalized to so-called convex piecewise linear functions. For instance, consider the following optimization model:

\[
\begin{align*}
\text{min} & \quad f(x_1) + 4x_2 \\
\text{s.t.} & \quad x_1 + x_2 \geq 10 \\
& \quad x_1, x_2 \geq 0,
\end{align*}
\]  

where

\[
f(x_1) = \begin{cases} 
  x_1 & \text{if } 0 \leq x_1 \leq 3 \\
  3 + 3(x_1 - 3) & \text{if } 3 < x_1 \leq 8 \\
  18 + 5(x_1 - 8) & \text{if } x_1 > 18.
\end{cases}
\]

Figure 1.7 shows the graph of \(f(x_1)\). The function \(f(x_1)\) is called piecewise linear, because it is linear on each of the intervals \([0, 3]\), \((3, 8]\), and \((8, \infty)\) separately. Like in the previous subsection, model (1.16) can be solved by an alternative formulation. We start by introducing three new nonnegative decision variables \(u_1, u_2,\) and \(u_3\). They will have the following relationship with \(x_1\):

\[
\begin{align*}
u_1 &= \begin{cases} 
  x_1 & \text{if } 0 \leq x_1 \leq 3 \\
  3 & \text{if } x_1 > 3,
\end{cases} \\
u_2 &= \begin{cases} 
  x_1 & \text{if } 0 \leq x_1 \leq 3 \\
  5 & \text{if } x_1 > 8,
\end{cases} \\
u_3 &= \begin{cases} 
  0 & \text{if } x_1 \leq 8 \\
  x_1 - 8 & \text{if } x_1 > 8.
\end{cases}
\end{align*}
\]

Crucially, we have that:

\[
u_1 + u_2 + u_3 = x_1 \quad \text{and} \quad u_1 + 3u_2 + 5u_3 = f(x_1).
\]  

(1.17)

To see this, we need to consider three different cases.

\begin{itemize}
  \item \(0 \leq x_1 \leq 3\). Then \(u_1 = x_1, u_2 = 0,\) and \(u_3 = 0\). Hence, \(u_1 + u_2 + u_3 = x_1\) and \(u_1 + 3u_2 + 5u_3 = x_1 = f(x_1)\).
  \item \(3 < x_1 \leq 8\). Then \(u_1 = 3, u_2 = x_1 - 3,\) and \(u_3 = 0\). Hence, \(u_1 + u_2 + u_3 = 3 + x_1 - 3 + 0 = x_1\) and \(u_1 + 3u_2 + 5u_3 = 3 + 3(x_1 - 3) = f(x_1)\).
  \item \(x_1 > 8\). Then \(u_1 = 3, u_2 = 5, u_3 = x_1 - 8,\) and \(f(x_1) = 18 + 5(x_1 - 8)\). Hence, \(u_1 + u_2 + u_3 = 3 + 5 + x_1 - 8 = x_1\) and \(u_1 + 3u_2 + 5u_3 = 3 + 15 + 5(x_1 - 8) = 18 + 5(x_1 - 8) = f(x_1)\).
\end{itemize}

In each of the cases, it is clear that (1.17) holds. Notice that \(u_1, u_2, u_3\) satisfy the following inequalities:

\[0 \leq u_1 \leq 3, \quad 0 \leq u_2 \leq 5, \quad u_3 \geq 0.\]

Moreover, we have the following equations (this follows from the definition of \(u_1, u_2, u_3\)):

\[u_2(3 - u_1) = 0 \quad \text{and} \quad u_3(5 - u_2) = 0.\]

The first equation states that either \(u_2 = 0,\) or \(u_1 = 3,\) or both. Informally, this says that \(u_2\) has a positive value only if \(u_1\) is at its highest possible value 3. The second equation states
that either \( u_3 = 0 \), or \( u_2 = 5 \), or both. This says that \( u_3 \) has a positive value only if \( u_3 \) is at its highest possible value 5. The two equations together imply that \( u_3 \) has a positive value only if both \( u_1 \) and \( u_2 \) are at their respective highest possible values (namely, \( u_1 = 3 \) and \( u_2 = 5 \)).

Adding these equations to (1.16) and substituting the expressions of (1.17), we obtain the following (nonlinear) optimization model:

\[
\begin{align*}
\text{min } & u_1 + 3u_2 + 5u_3 + 4x_2 \\
\text{s.t. } & u_1 + u_2 + u_3 + x_2 \geq 10 \\
& u_1 \leq 3 \\
& u_2 \leq 5 \\
& (3 - u_1)u_2 = 0 \\
& (5 - u_2)u_3 = 0 \\
& u_1, u_2, u_3, x_2 \geq 0.
\end{align*}
\]

As in Section 1.5.2, it turns out that the nonlinear constraints \((3 - u_1)u_2 = 0\) and \((5 - u_2)u_3 = 0\) may be omitted. To see this, let \( x^* = [u_1^* \ u_2^* \ u_3^* \ x_2^*]^T \) be an optimal solution of (1.18), and let \( z^* \) be the corresponding optimal objective value. Suppose for a contradiction that \( u_2^*(3 - u_1^*) \neq 0 \). Because \( u_2^* \geq 0 \) and \( u_1^* \leq 3 \), this implies that \( u_1^* < 3 \) and \( u_2^* > 0 \). Let \( \varepsilon = \min\{3 - u_1^*, u_2^*\} > 0 \), and define \( \hat{u}_1 = u_1^* + \varepsilon, \hat{u}_2 = u_2^* - \varepsilon, \hat{u}_3 = u_3^* \), and \( \hat{x}_2 = x_2^* \). It is straightforward to check that the vector \( \hat{x} = [\hat{u}_1 \ \hat{u}_2 \ \hat{u}_3 \ \hat{x}_2]^T \) is a feasible solution of (1.18). The objective value corresponding to \( \hat{x} \) satisfies:

\[
\hat{u}_1 + 3\hat{u}_2 + 5\hat{u}_3 + 4\hat{x}_2 = (u_1^* + \varepsilon) + 3(u_2^* - \varepsilon) + 5u_3^* + 4x_2^* = u_1^* + 3u_2^* + 5u_3^* + 4x_2^* - 2\varepsilon = z^* - 2\varepsilon < z^*,
\]

contrary to the fact that \( x^* \) is an optimal solution of (1.18). Therefore, \((3 - u_1^*)u_2^* = 0\) is satisfied for any optimal solution \([u_1^* \ u_2^* \ u_3^* \ x_2^*]^T\) of (1.18), and hence the constraint \((3 - u_1)u_2 = 0\) can be omitted. Similarly, the constraint \((5 - u_2)u_3 = 0\) can be omitted. This means that model (1.16) may be solved by solving the following LO model:

\[
\begin{align*}
\text{min } & u_1 + 3u_2 + 5u_3 + 4x_2 \\
\text{s.t. } & u_1 + u_2 + u_3 + x_2 \geq 10 \\
& u_1 \leq 3 \\
& u_2 \leq 5 \\
& u_1, u_2, u_3, x_2 \geq 0.
\end{align*}
\]

An optimal solution of (1.19) can be found using the graphical method, similarly to the discussion in Section 1.5.2. The cases to consider are: (1) \( u_2 = u_3 = 0 \), (2) \( u_1 = 3 \) and \( u_3 = 0 \), and (3) \( u_1 = 3 \) and \( u_2 = 5 \). An optimal solution turns out to be \( u_1^* = 3, u_2^* = 5, u_3^* = 0, \) and \( x_2^* = 2 \). This means that an optimal solution of (1.16) is \( x_1^* = 8 \) and \( x_2^* = 2 \).

The piecewise linear function \( f(x_1) \) has a special property: it is a convex function (see Appendix D). This is a necessary condition for the method described in this section to work.
In Exercise 1.8.12, the reader is given a model with a nonconvex piecewise linear objective function and is asked to show that the technique does not work in that case.

### 1.6 Examples of linear optimization models

Linear and integer linear optimization are used in a wide range of subjects. Robert E. Bixby (born 1945) has collected the following impressive list of applications.

- **Transportation — airlines**: fleet assignment; crew scheduling; personnel scheduling; yield management; fuel allocation; passenger mix; booking control; maintenance scheduling; load balancing; freight packing; airport traffic planning; gate scheduling; upset recovery and management.
- **Transportation — others**: vehicle routing; freight vehicle scheduling and assignment; depot and warehouse location; freight vehicle packing; public transportation system operation; rental car fleet management.
- **Financial**: portfolio selection and optimization; cash management; synthetic option development; lease analysis; capital budgeting and rationing; bank financial planning; accounting allocations; securities industry surveillance; audit staff planning; assets liabilities management; unit costing; financial valuation; bank shift scheduling; consumer credit delinquency management; check clearing systems; municipal bond bidding; stock exchange operations; debt financing optimization.
- **Process industries (chemical manufacturing, refining)**: plant scheduling and logistics; capacity expansion planning; pipeline transportation planning; gasoline and chemical blending.
- **Manufacturing**: product mix planning; blending; manufacturing scheduling; inventory management; job scheduling; personnel scheduling; maintenance scheduling and planning; steel production scheduling; blast furnace burdening in the steel industry.
- **Coal industry**: coal sourcing and transportation logistics; coal blending; mining operation management.
- **Forestry**: Forest land management; forest valuation models; planting and harvesting models.
- **Agriculture**: production planning; farm land management; agriculture pricing models; crop and product mix decision models; product distribution.
- **Oil and gas exploration and production**: oil and gas production scheduling; natural gas transportation planning.
- **Public utilities and natural resources**: electric power distribution; power generator scheduling; power tariff rate determination; natural gas distribution planning; natural gas pipeline transportation; water resource management; alternative water supply evaluation; water reservoir management; public water transportation models; mining excavation models.
Food processing: food blending; recipe optimization; food transportation logistics; food manufacturing logistics and scheduling.

Communications and computing: circuit board (VLSI) layout; logical circuit design; magnetic field design; complex computer graphics; curve fitting; virtual reality systems; computer system capacity planning; office automation; multiprocessor scheduling; telecommunications scheduling; telephone operator scheduling; telemarketing site selection.

Health care: hospital staff scheduling; hospital layout; health cost reimbursement; ambulance scheduling; radiation exposure models.

Pulp and paper industry: inventory planning; trimloss minimization; waste water recycling; transportation planning.

Textile industry: pattern layout and cutting optimization; production scheduling.

Government and military: post office scheduling and planning; military logistics; target assignment; missile detection; manpower deployment.

Miscellaneous applications: advertising mix/media scheduling; sales region definition; pollution control models; sales force deployment.

The current section contains a number of linear optimization models that illustrate the wide range of applications from real world problems. They also illustrate the variety and the complexity of the modeling process. See also Chapters 10–11 for more real world applications.

I.6.1 The diet problem

A doctor prescribes to a patient exact amounts of daily vitamin A and vitamin C intake. Specifically, the patient should choose her diet so as to consume exactly 3 milligrams of vitamin A and exactly 75 milligrams of vitamin C. The patient considers eating three kinds of food, which contain different amounts of vitamins and have different prices. She wants to determine how much of each food she should buy in order to minimize her total expenses, while making sure to ingest the prescribed amounts of vitamins. Let $x_i$ be the amount of food $i$ that she should buy ($i = 1, 2, 3$). Each unit of food 1 contains 1 milligram of vitamin A and 30 milligrams of vitamin C, each unit of food 2 contains 2 milligrams of vitamin A and 10 milligrams of vitamin C, and each unit of food 3 contains 2 milligrams of vitamin A and 20 milligrams of vitamin C. The unit cost of food 1, 2, and 3 is $40, $100, and $150 per week, respectively. This problem can be formulated as follows:

$$\begin{align*}
\text{min} & \quad 40x_1 + 100x_2 + 150x_3 \\
\text{s.t.} & \quad x_1 + 2x_2 + 2x_3 = 3 \\
& \quad 30x_1 + 10x_2 + 20x_3 = 75 \\
& \quad x_1, x_2, x_3 \geq 0.
\end{align*}$$


Finding an optimal solution of this LO-model can be done by elimination as follows. First, subtracting ten times the first constraint from the second one yields:

\[ 20x_1 - 10x_2 = 45, \quad \text{or, equivalently,} \quad x_2 = 2x_1 - 4.5. \]

Similarly, subtracting five times the first constraint from the second one yields:

\[ 25x_1 + 10x_3 = 60, \quad \text{or, equivalently,} \quad x_3 = 6 - 2.5x_1. \]

Substituting these expressions for \( x_2 \) and \( x_3 \) into the original model, we obtain:

\[
\begin{align*}
\min & \quad 40x_1 + 100(2x_1 - 4.5) + 150(6 - 2.5x_1) \\
\text{s.t.} & \quad 6 - 2.5x_1 \geq 0 \\
& \quad x_1 \geq 0.
\end{align*}
\]

Hence, the model becomes:

\[
\begin{align*}
\min & \quad -135x_1 + 450 \\
\text{s.t.} & \quad x_1 \geq 2.25 \\
& \quad x_1 \leq 2.4 \\
& \quad x_1 \geq 0.
\end{align*}
\]

Since the objective coefficient of \( x_1 \) is negative, the optimal solution is found by choosing \( x_1 \) as large as possible, i.e., \( x_1^* = 2.4 \). Thus, the optimal solution is \( x_1^* = 2.4, x_2^* = 2 \times 2.4 - 4.5 = 0.3, x_3^* = 6 - 2.5 \times 2.4 = 0 \), and \( z^* = -135 \times 2.4 + 450 = 126 \), which means that she should buy 2.4 units of food 1, 0.3 units of food 2, and none of food 3. The total cost of this diet is $126 per week.

An interesting phenomenon appears when one of the right hand side values is changed. Suppose that the doctor’s prescription was to take 85 milligrams of vitamin C instead of 75 milligrams. So, the corresponding LO-model is:

\[
\begin{align*}
\min & \quad 40x_1 + 100x_2 + 150x_3 \\
\text{s.t.} & \quad x_1 + 2x_2 + 2x_3 = 3 \\
& \quad 30x_1 + 10x_2 + 20x_3 = 85 \\
& \quad x_1, x_2, x_3 \geq 0.
\end{align*}
\]

The optimal solution of this new model is \( x_1^* = 2.8, x_2^* = 0.1, x_3^* = 0 \), and \( z^* = 122 \). Observe that the corresponding diet is $4 cheaper than the original diet. Hence the patient gets more vitamins for less money. This is the so-called more-for-less paradox. It is of course only a seeming paradox, because the fact that ‘an exact amount’ of the vitamins is prescribed can in practice be relaxed to ‘at least the amount’. Replacing the equality signs in the original LO-model by ‘\( \geq \)’ signs gives the optimal solution \( x_1^* = 3, x_2^* = 0, x_3^* = 0 \), and \( z^* = 120 \). The corresponding vitamin A and C intake is 3 and 90 milligrams, respectively. This ‘paradox’ therefore only tells us that we have to be careful when formulating LO-models, and with the interpretation of the solution.
We conclude with the model code for the diet problem:

```plaintext
var x1 >= 0;
var x2 >= 0;
var x3 >= 0;

minimize z:
40 * x1 + 100 * x2 + 150 * x3;

subject to vitaminA:
x1 + 2 * x2 + 2 * x3 = 3;

subject to vitaminC:
30 * x1 + 10 * x2 + 20 * x3 = 75;

end;
```

**Listing 1.2:** The diet problem.

### 1.6.2 Estimation by regression

Suppose that we want to estimate a person’s federal income tax based on a certain personal profile. Suppose that this profile consists of values for a number of well-defined attributes. For example, we could use the number of semesters spent in college, the income, the age, or the total value of the real estate owned by the person. In order to carry out the estimation, we collect data of say \((n =)\) 11 persons. Let \(m\) be the number of attributes, labeled 1, \ldots, \(m\). We label the persons in the sample as \(i = 1, \ldots, n\). For \(i = 1, \ldots, n\), define \(a_i = [a_{i1} \ldots a_{im}]^T\) with \(a_{ij}\) the value of attribute \(j\) of person \(i\). Moreover, let \(b_i\) be the amount of the federal income tax to be paid by person \(i\). It is assumed that the values of the eleven vectors \(a_1 = [a_{11} a_{12}]^T, \ldots, a_{11} = [a_{11,1} a_{11,2}]^T\) together with the values of \(b_1, \ldots, b_{11}\) are known. Table 1.1 lists an example data set for eleven persons of whom we collected the values of two attributes, \(a_1\) and \(a_2\). In Figure 1.8, we have plotted these data points.

The question now is: how can we use this data set in order to estimate the federal income tax \(b\) of any given person (who is not in the original data set) based on the values of a given profile vector \(a\)? To that end, we construct a graph ‘through’ the data points \([a_{11} \ldots a_{1m} b]^T, \ldots, [a_{n1} \ldots a_{nm} b]^T\) in such a way that the total distance between these points and this graph is as small as possible. Obviously, how small or large this total distance is depends on the shape of the graph. In Figure 1.8, the shape of the graph is a plane in three-dimensional space. In practice, we may take either a convex or a concave graph (see Appendix D). However, when the data points do not form an apparent ‘shape’, we may choose a hyperplane. This hyperplane is constructed in such a way that the sum of the deviations of the \(n\) data points \([a_{i1} \ldots a_{im} b]^T (i = 1, \ldots, n)\) from this hyperplane is as small as possible. The general form of such a hyperplane \(H(u, v)\) is:

\[
H(u, v) = \{ [a_{1} \ldots a_{m} b]^T \in \mathbb{R}^{m+1} \mid b = u^T a + v \}.
\]
(see also Section 2.1.1) with variables $a (\in \mathbb{R}^m)$ and $b (\in \mathbb{R})$. The values of the parameters $u (\in \mathbb{R}^m)$ and $v (\in \mathbb{R})$ need to be determined such that the total deviation between the $n$ points $[a_{i1} \ldots a_{im} b_i]^T$ (for $i = 1, \ldots, n$) and the hyperplane is as small as possible. As the deviation of the data points from the hyperplane, we use the ‘vertical’ distance. That is, for each $i = 1, \ldots, n$, we take as the distance between the hyperplane $H$ and the point $[a_{i1} \ldots a_{im} b_i]^T$:

$$|u^T a_i + v - b_i|.$$ 

In order to minimize the total deviation, we may solve the following LO-model:

$$\begin{align*}
\min & \quad y_1 + \ldots + y_n \\
\text{s.t.} & \quad -y_i \leq a_i^T u + v - b_i \leq y_i \\
& \quad y_i \geq 0 \\
& \quad \text{for } i = 1, \ldots, n.
\end{align*}$$

In this LO-model, the variables are the entries of $y$, the entries of $u$, and $v$. The values of $a_i$ and $b_i$ ($i = 1, \ldots, n$) are given. The ‘average’ hyperplane ‘through’ the data set reads:

$$H(u^*, v^*) = \left\{ [a_1, \ldots, a_m, b]^T \in \mathbb{R}^{m+1} \mid b = (u^*)^T x + v^* \right\},$$

where $u^*$ and $v^*$ are optimal values for $u$ and $v$, respectively. Given this hyperplane, we may now estimate the income tax to be paid by a person that is not in our data set, based on the person’s profile. In particular, for a person with given profile $\tilde{a}$, the estimated income tax to be paid is $\tilde{b} = (u^*)^T \tilde{a} + v^*$. The optimal solution obviously satisfies $y_i^* = |a_i^T u^* + v^* - b_i|$, with $u^*$ and $v^*$ the optimal values, so that the optimal value of $y_i^*$ measures the deviation of data point $[a_{i1} \ldots a_{im} b_i]$ from the hyperplane.

<table>
<thead>
<tr>
<th>Person $i$</th>
<th>$b_i$</th>
<th>$a_{i1}$</th>
<th>$a_{i2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4,585</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>7,865</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>3,379</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>6,203</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>2,466</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>3,248</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>4,972</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>3,437</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>3,845</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>3,878</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>5,674</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1.1: Federal income tax data for eleven persons. The second column contains the amount of income tax paid, and the third and fourth columns contain the profile of each person.

Figure 1.8: Plot of eleven persons’ federal income tax $b$ as a function of two attributes $a_1$ and $a_2$. The dots are the data points. The lines show the distance from each data point to the hyperplane $H$. 

For the example with the data given in Table 1.1, the optimal solution turns out to be:

\[ u^*_1 = 249.5, \quad u^*_2 = 368.5, \quad v^* = 895.5, \quad y^*_1 = 494.5, \quad y^*_2 = 1039, \quad y^*_3 = 0, \quad y^*_4 = 2348, \]
\[ y^*_5 = 1995.5, \quad y^*_6 = 1236.5, \quad y^*_7 = 0, \quad y^*_8 = 1178, \quad y^*_9 = 1226, \quad y^*_10 = 0, \quad y^*_11 = 214.5. \]

Thus, the best hyperplane \( H \) through the points is given by:

\[ H = \left\{ \begin{bmatrix} a_1 & a_2 & b \end{bmatrix}^T \in \mathbb{R}^3 \mid b = 249.5a_1 + 368.5a_2 + 895.5 \right\}. \]

Moreover, for a person with profile \( \hat{a} = [\hat{a}_1, \hat{a}_2] \), the estimated income tax to be paid is \( \hat{b} = 249.5\hat{a}_1 + 368.5\hat{a}_2 + 895.5 \). This means that the second attribute has more influence on the estimate than the first attribute. Also, from the values of \( y^*_1, \ldots, y^*_11 \), we see that the distances from the data points corresponding to persons 5 and 6 are largest, meaning that the amount of income tax that they had to pay deviates quite a bit from the amount predicted by our model. For example, for person 5, the estimated income tax is \((249.5 \times 1 + 368.5 \times 9 + 895.5 =) \$4,461.50\), whereas this person’s actual income tax was \$2,466.00.

The model described above is a special kind of (linear) regression model. Regression models are used widely in statistics. One of the most common such models is the so-called least squares regression model, which differs from the model above in that the distance between the data points and the hyperplane is not measured by the absolute value of the deviation, but by the square of the deviation. Thus, in a least squares regression model, the objective is to minimize the sum of the squared deviations. In addition to different choices of the distance function, it is also possible to apply ‘better’ graphs, such as convex or concave functions. Such models are usually nonlinear optimization models and hence these topics lie outside of the scope of this book.

### 1.6.3 Team formation in sports

Another application of linear optimization is the formation of sports teams. In this section we describe a simple LO-model for choosing a soccer line-up from a given set of players. In soccer, a line-up consists of a choice of eleven players (from a potentially larger set) that are assigned to eleven positions. The positions depend on the system of play. For example, in a so-called 1-4-3-3 system, the team consists of a goal keeper, four defenders, three midfielders, and three forward players.

Let \( N \) denote the number of players that we can choose from, and let \( M \) denote the number of positions in the field (i.e., \( M = 11 \) in the case of soccer). We assume that \( N \geq M \), i.e., there are enough players available to play a game. The players that are not lined up will not play. For each \( i = 1, \ldots, N \) and \( j = 1, \ldots, M \), define the decision variable \( x_{ij} \), with the following meaning:

\[ x_{ij} = \begin{cases} 1 & \text{if player } i \text{ is assigned to position } j \\ 0 & \text{otherwise.} \end{cases} \]
Since not every assignment of 0’s and 1’s to the \( x_{ij} \)’s represents a legal line-up, we need to impose some restrictions on the values of the \( x_{ij} \)’s. For one, we need to make sure that exactly one player is assigned to each position. This is captured by the following set of \( M \) constraints:

\[
\sum_{i=1}^{N} x_{ij} = 1 \quad \text{for } j = 1, \ldots, M.
\]

These constraints state that, for each position \( j \), the sum of \( x_{1j}, \ldots, x_{Nj} \) equals 1. Since each of the \( x_{ij} \)’s should equal either 0 or 1, this implies that exactly one of \( x_{1j}, \ldots, x_{Nj} \) will have value 1. Moreover, although not every player has to be lined up, we require that no player is lined up on two different positions. This is achieved by the following set of \( N \) constraints:

\[
\sum_{j=1}^{M} x_{ij} \leq 1 \quad \text{for } i = 1, \ldots, N.
\]

These constraints state that for each player \( i \), at most one of the \( x_{i1}, \ldots, x_{iM} \) has value 1.

In order to formulate an optimization model, we also need to add an objective function which measures how good a given line-up is. To do so, we introduce the parameter \( c_{ij} \) (\( i = 1, \ldots, N \) and \( j = 1, \ldots, M \)), which measures how well player \( i \) fits on position \( j \). A question that arises is of course: how to determine the values of the \( c_{ij} \)’s? We will come back to this later. With the values of the \( c_{ij} \)’s at hand, we can now write an objective. Let us say that if player \( i \) is assigned to position \( j \), this player contributes \( c_{ij} \) to the objective function. Thus, the objective is to maximize the sum of the \( c_{ij} \)’s, over all pairs \( i, j \), such that player \( i \) is assigned to position \( j \). The corresponding objective function can be written as a linear function of the \( x_{ij} \)’s. The objective then reads:

\[
\max \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij}.
\]

Since \( x_{ij} = 1 \) if and only if player \( i \) is assigned to position \( j \), the term \( c_{ij} x_{ij} \) equals \( c_{ij} \) if player \( i \) is assigned to position \( j \), and 0 otherwise, as required.

Combining the constraints and the objective function, we have the following optimization model:

\[
\max \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij}
\]

s.t. \( \sum_{i=1}^{N} x_{ij} = 1 \quad \text{for } j = 1, \ldots, M \)

\( \sum_{j=1}^{M} x_{ij} \leq 1 \quad \text{for } i = 1, \ldots, N \)

\( x_{ij} \in \{0, 1\} \quad \text{for } i = 1, \ldots, N \text{ and } j = 1, \ldots, M. \)

This optimization model, in its current form, is not an LO-model, because its variables are restricted to have integer values and this type of constraint is not allowed in an LO-model.
Actually, the model is a so-called integer linear optimization model (abbreviated as ILO-model). We will see in Chapter 7 that, in general, integer linear optimization models are hard to solve. If we want to write the model as an LO-model, we will have to drop the constraint that the decision variables be integer-valued. So let us consider the following optimization model instead:

\[
\begin{align*}
\text{max} & \quad \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_{i=1}^{N} x_{ij} = 1 \quad \text{for all } j = 1, \ldots, M \\
& \quad \sum_{j=1}^{M} x_{ij} \leq 1 \quad \text{for all } i = 1, \ldots, N \\
& \quad 0 \leq x_{ij} \leq 1 \quad \text{for all } i = 1, \ldots, N \text{ and } j = 1, \ldots, M.
\end{align*}
\] (1.20)

This model is an LO-model. But recall from Model Dovetail that, in general, an LO-model may have a solution whose coordinate values are fractional. Therefore, by finding an optimal solution of (1.20), we run the risk of finding an optimal solution that has a fractional value for \( x_{ij} \) for some \( i \) and \( j \). Since it does not make sense to put only part of a player in the field, this is clearly not desirable (although one could interpret half a player as a player who is playing only half of the time). However, for this particular LO-model something surprising happens: as it turns out, (1.20) always has an optimal solution for which all \( x_{ij} \)'s are integer-valued, i.e., they are either 0 or 1. The reason for this is quite subtle and will be described in Chapter 8. But it does mean that (1.20) correctly models the team formation problem.

We promised to come back to the determination of the values of the \( c_{ij} \)'s. One way is to just make educated guesses. For example, we could let the values of \( c_{ij} \) run from 0 to 5, where 0 means that the player is completely unfit for the position, and 5 means that the player is perfect for the position.

A more systematic approach is the following. We can think of forming a team as a matter of economic supply and demand of qualities. Let us define a list of ‘qualities’ that play a role in soccer. The positions demand certain qualities, and the players supply these qualities. The list of qualities could include, for example, endurance, speed, balance, agility, strength, inventiveness, confidence, left leg skills, right leg skills. Let the qualities be labeled \( 1, \ldots, Q \). For each position \( j \), let \( d_{jq} \) be the ‘amount’ of quality \( q \) demanded on position \( j \) and, for each player \( i \), let \( s_{iq} \) be the ‘amount’ of quality \( q \) supplied by player \( i \). We measure these numbers all on the same scale from 0 to 5. Now, for all \( i \) and \( j \), we can define how well player \( i \) fits on position \( j \) by, for example, calculating the average squared deviation of player \( i \)'s supplied qualities compared to the qualities demanded for position \( j \):

\[
c_{ij} = -\sum_{q=1}^{Q} (s_{iq} - d_{jq})^2.
\]
The negative sign is present because we are maximizing the sum of the $c_{ij}$’s, so a player $i$ that has exactly the same qualities as demanded by a position $j$ has $c_{ij} = 0$, whereas any deviation from the demanded qualities will give a negative number.

Observe that $c_{ij}$ is a nonlinear function of the $s_{iq}$’s and $d_{jq}$’s. This, however, does not contradict the definition of an LO-model, because the $c_{ij}$’s are not decision variables of the model; they show up as parameters of the model, in particular as the objective coefficients.

The current definition of $c_{ij}$ assigns the same value to a positive deviation and a negative deviation. Even worse: suppose that there are two players, $i$ and $i’$, say, who supply exactly the same qualities, i.e., $s_{iq} = s_{i’q}$, except for some quality $q$, for which we have that $s_{iq} = d_{jq} - 1$ and $s_{i’q} = d_{jq} + 2$ for position $j$. Then, although player $i’$ is clearly better for position $j$ than player $i$ is, we have that $c_{ij} > c_{i’j}$, which does not make sense. So, a better definition for $c_{ij}$ should not have that property. For example, the function:

$$c_{ij} = -\sum_{q=1}^{Q} \left( \min(0, s_{iq} - d_{jq}) \right)^{2}$$

does not have this property. It is left to the reader to check this assertion.

### 1.6.4 Data envelopment analysis

In this section, we describe data envelopment analysis (DEA), an increasingly popular management tool for comparing and ranking the relative efficiencies of so-called decision making units (DMUs), such as banks, branches of banks, hospitals, hospital departments, universities, or individuals. After its introduction in 1978, data envelopment analysis has become a major tool for performance evaluation and benchmarking, especially for cases where there are complex relationships between multiple inputs and multiple outputs. The interested reader is referred to Cooper et al. (2011). An important part of DEA concerns the concept of efficiency. The efficiency of a DMU is roughly defined as the extent to which the inputs for that DMU are used to produce its outputs, i.e.,

$$\text{efficiency} = \frac{\text{output}}{\text{input}}.$$ 

The efficiency is defined in such a way that an efficiency of 1 means that the DMU makes optimal use of its inputs. So, the efficiency can never exceed 1. This measure of efficiency, however, may not be adequate because there are usually multiple inputs and outputs related to, for example, different resources, activities, and environmental factors. It is not immediately apparent how to compare these (possibly very different) inputs and outputs with each other. DEA takes a data driven approach. We will discuss here the basic ideas behind DEA, and use an example to show how relative efficiencies can be determined and how targets for relatively inefficient DMUs can be set.

As an example, consider the data presented in Table 1.2. We have ten universities that are the DMUs. All ten universities have the same inputs, namely real estate and wages. They also
produce the same outputs, namely economics, business, and mathematics graduates. The universities differ, however, in the amounts of the inputs they use, and the amounts of the outputs they produce. The first two columns of Table 1.2 contain the input amounts for each university, and the last three columns contain the output amounts.

Given these data, comparing universities 1 and 2 is straightforward. University 1 has strictly larger outputs than university 2, while using less inputs than university 2. Hence, university 1 is more efficient than university 2. But how should universities 3 and 4 be compared? Among these two, university 3 has the larger number of graduated economics students, and university 4 has the larger number of graduated business and mathematics students. So, the outputs of universities 3 and 4 are hard to compare. Similarly, the inputs of universities 3 and 5 are hard to compare. In addition, it is hard to compare universities 9 and 10 to any of the other eight universities, because they are roughly twice as large as the other universities.

In DEA, the different inputs and outputs are compared with each other by assigning weights to each of them. However, choosing weights for the several inputs and outputs is generally hard and rather arbitrary. For example, it may happen that different DMUs have organized their operations differently, so that the output weights should be chosen differently. The key idea of DEA is that each DMU is allowed to choose its own set of weights. Of course, each DMU will then choose a set of weights that is most favorable for their efficiency assessment. However, DMUs are not allowed to ‘cheat;’ a DMU should choose the weights in such a way that the efficiency of the other DMUs is restricted to at most 1. It is assumed that all DMUs convert (more or less) the same set of inputs into the same set of outputs: only the weights of the inputs and outputs may differ among the DMUs. DEA can be formulated as a linear optimization model as follows.

<table>
<thead>
<tr>
<th>DMU</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Wages</td>
</tr>
<tr>
<td>1</td>
<td>72</td>
<td>81</td>
</tr>
<tr>
<td>2</td>
<td>73</td>
<td>82</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>87</td>
<td>83</td>
</tr>
<tr>
<td>5</td>
<td>53</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>71</td>
<td>85</td>
</tr>
<tr>
<td>7</td>
<td>65</td>
<td>68</td>
</tr>
<tr>
<td>8</td>
<td>59</td>
<td>62</td>
</tr>
<tr>
<td>9</td>
<td>134</td>
<td>186</td>
</tr>
<tr>
<td>10</td>
<td>134</td>
<td>140</td>
</tr>
</tbody>
</table>

Table 1.2: Inputs and outputs for ten universities.
Let the DMUs be labeled \((k = 1, \ldots, N)\). For each \(k = 1, \ldots, N\), the relative efficiency (or efficiency rate) \(RE(k)\) of DMU \(k\) is defined as:

\[
RE(k) = \frac{\text{weighted sum of output values of DMU } k}{\text{weighted sum of input values of DMU } k},
\]

where \(0 \leq RE(k) \leq 1\). DMU \(k\) is called relatively efficient if \(RE(k) = 1\), and relatively inefficient otherwise. Note that if DMU \(k\) is relatively efficient, then its total input value equals its total output value. In other words, a relative efficiency rate \(RE(k)\) of DMU \(k\) means that DMU \(k\) is able to produce its outputs with a \(100\%\) use of its inputs.

In order to make this definition more precise, we introduce the following notation. Let \(m (\geq 1)\) be the number of inputs, and \(n (\geq 1)\) the number of outputs. For each \(i = 1, \ldots, m\) and \(j = 1, \ldots, n\), define:

\[
\begin{align*}
x_i &= \text{the weight of input } i; \\
y_j &= \text{the weight of output } j; \\
u_{ik} &= \text{the (positive) amount of input } i \text{ to DMU } k; \\
v_{jk} &= \text{the (positive) amount of output } j \text{ to DMU } k.
\end{align*}
\]

These definitions suggest that the same set of weights, the \(x_i\)'s and the \(y_j\)'s, are used for all DMUs. As described above, this is not the case. In DEA, each DMU is allowed to adopt its own set of weights, namely in such a way that its own relative efficiency is maximized. Hence, for each DMU, the objective should be to determine a set of input and output weights that yields the highest efficiency for that DMU in comparison to the other DMUs. The optimization model for DMU \(k \ (= 1, \ldots, N)\) can then be formulated as follows:

\[
RE^*(k) = \max \frac{v_{1k}y_1 + \ldots + v_{nk}y_n}{u_{1k}x_1 + \ldots + u_{mk}x_m}
\]

s.t. \[
\frac{v_{1r}y_1 + \ldots + v_{nr}y_n}{u_{1r}x_1 + \ldots + u_{mr}x_m} \leq 1 \quad \text{for } r = 1, \ldots, N
\]

\[
x_1, \ldots, x_m, y_1, \ldots, y_n \geq \varepsilon.
\]

The decision variables in this model are constrained to be at least equal to some small positive number \(\varepsilon\), so as to avoid any input or output becoming completely ignored in determining the efficiencies. We choose \(\varepsilon = 0.00001\). Recall that in the above model, for each \(k\), the weight values are chosen so as to maximize the efficiency of DMU \(k\). Also note that we have to solve \(N\) such models, namely one for each DMU.

Before we further elaborate on model \((M_1(k))\), we first show how it can be converted into a linear model. Model \((M_1(k))\) is a so-called fractional linear model, i.e., the numerator and the denominator of the fractions are all linear in the decision variables. Since all denominators are positive, this fractional model can easily be converted into a common LO-model. In order to do so, first note that when maximizing a fraction, it is only the relative value of the numerator and the denominator that are of interest and not the individual values. Therefore, we can set the value of the denominator equal to a constant value (say, 1), and then maximize
the numerator. For each \( k = 1, \ldots, N \), we obtain the following LO-model:

\[
RE^*(k) = \max \ v_k y_1 + \ldots + v_n y_n \\
\text{s.t.} \quad u_k x_1 + \ldots + u_m x_m = 1 \\
\v_j y_1 + \ldots + v_r y_j - u_j x_1 - \ldots - u_m x_m \leq 0 \\
\text{for } r = 1, \ldots, N \\
1 \leq y_j \leq N, \quad j = 1, \ldots, n
\]

Obviously, if \( RE^*(k) = 1 \), then DMU \( k \) is relatively efficient, and if \( RE^*(k) < 1 \), then there is a DMU that is more efficient than DMU \( k \). Actually, there is always one that is relatively efficient, because at least one of the ‘\( \leq \)'-constraints of \( (M_2(k)) \) is satisfied with equality in the optimal solution of \( (M_2(k)) \); see Exercise 1.8.17. If \( RE^*(k) < 1 \), then the set of all relatively efficient DMUs is called the peer group \( PG(k) \) of the inefficient DMU \( k \). To be precise, for any DMU \( k \) with \( RE^*(k) < 1 \), we have that:

\[
PG(k) = \{ p \in \{1, \ldots, N\} | RE^*(p) = 1 \}.
\]

The peer group contains DMUs that can be set as a target for the improvement of the relative efficiency of DMU \( k \).

Consider again the data of Table 1.2. Since there are ten universities, there are ten models to be solved. For example, for \( k = 1 \), the LO-model \( M_2(1) \) reads:

\[
\begin{align*}
\max & \quad 72y_1 + 81y_2 \\
\text{s.t.} & \quad 77x_1 + 73x_2 + 78x_3 = 1 \\
& \quad 72y_1 + 81y_2 - 77x_1 - 73x_2 - 78x_3 \leq 0 \\
& \quad 73y_1 + 82y_2 - 73x_1 - 70x_2 - 69x_3 \leq 0 \\
& \quad 70y_1 + 59y_2 - 72x_1 - 67x_2 - 80x_3 \leq 0 \\
& \quad 87y_1 + 83y_2 - 69x_1 - 74x_2 - 84x_3 \leq 0 \\
& \quad 53y_1 + 64y_2 - 57x_1 - 65x_2 - 65x_3 \leq 0 \\
& \quad 71y_1 + 85y_2 - 78x_1 - 72x_2 - 73x_3 \leq 0 \\
& \quad 65y_1 + 68y_2 - 81x_1 - 71x_2 - 69x_3 \leq 0 \\
& \quad 59y_1 + 62y_2 - 64x_1 - 66x_2 - 56x_3 \leq 0 \\
& \quad 67y_1 + 93y_2 - 75x_1 - 84x_2 - 86x_3 \leq 0 \\
& \quad 67y_1 + 70y_2 - 67x_1 - 65x_2 - 65x_3 \leq 0 \\
& \quad x_1, x_2, x_3, y_1, y_2 \geq 0.00001.
\end{align*}
\]

Listing 1.3 contains the optimization model written as a GMPL model.

```plaintext
1  set INPUT;
2  set OUTPUT;
3  param N >= 1;
4  param u(1..N, INPUT);    # input values
5  param v(1..N, OUTPUT);  # output values
6
```
The optimal solution reads: $y_1^* = 0.01092$, $y_2^* = 0.00264$, $x_1^* = 0.00505$, $x_2^* = 0.00001$, $x_3^* = 0.00694$, with optimal objective value 0.932. Hence, the relative efficiency of uni-
Table 1.3: DEA results for the data in Table 1.2. The weights have been multiplied by 1,000.

<table>
<thead>
<tr>
<th>DMU</th>
<th>Input weights</th>
<th>Output weights</th>
<th>Efficiency</th>
<th>Peer set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real estate</td>
<td>Economies graduates</td>
<td>Business graduates</td>
<td>Math graduates</td>
</tr>
<tr>
<td>1</td>
<td>1.092</td>
<td>0.05</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>1.013</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>4</td>
<td>0.88</td>
<td>2.95</td>
<td>0.01</td>
<td>1.30</td>
</tr>
<tr>
<td>5</td>
<td>0.54</td>
<td>11.11</td>
<td>0.01</td>
<td>14.89</td>
</tr>
<tr>
<td>6</td>
<td>1.407</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>7</td>
<td>1.58</td>
<td>13.20</td>
<td>12.33</td>
<td>0.01</td>
</tr>
<tr>
<td>8</td>
<td>0.54</td>
<td>10.85</td>
<td>1.86</td>
<td>13.34</td>
</tr>
<tr>
<td>9</td>
<td>0.1179</td>
<td>0.26</td>
<td>4.24</td>
<td>8.11</td>
</tr>
<tr>
<td>10</td>
<td>0.80</td>
<td>8.93</td>
<td>1.27</td>
<td>11.70</td>
</tr>
</tbody>
</table>

University 1 is:

\[
RE^*(1) = \frac{72y_1^* + 81y_2^*}{73x_2^* + 78x_3^*} = z^* = 0.932.
\]

We have listed the optimal solutions, the relative efficiencies, and the peer sets in Table 1.3. It turns out that universities 3, 5, 7, 8, and 9 are relatively efficient, and that the other universities are relatively inefficient. Note that universities 9 and 10 are about twice as large as the other universities. Recall that the DEA approach ignores the scale of a DMU and only considers the relative sizes of the inputs and the outputs.

1.6.5 Portfolio selection; profit versus risk

An investor considers investing $10,000 in stocks during the next month. There are \( n \) (\( n \geq 1 \)) different stocks that can be bought. The investor wants to buy a portfolio of stocks at the beginning of the month, and sell them at the end of the month, without making any changes to the portfolio during that time. Since the investor wants to make as much profit as possible, a portfolio should be selected that has the highest possible total selling price at the end of the month.

Let \( i = 1, \ldots, n \). Define \( R_i \) to be the rate of return of stock \( i \), i.e.:

\[
R_i = \frac{V_i^1}{V_i^0},
\]

where \( V_i^0 \) is the current value of stock \( i \), and \( V_i^1 \) is the value of stock \( i \) in one month. This means that if the investor decides to invest $1 in stock \( i \), then this investment will be worth \( $R_i \) at the end of the month. The main difficulty with portfolio selection, however, is that the rate of return is not known in advance, i.e., it is uncertain. This means that we cannot know in advance how much any given portfolio will be worth at the end of the month.
### 1.6. Examples of Linear Optimization Models

<table>
<thead>
<tr>
<th>Scenario $s$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>−4.23</td>
<td>−1.58</td>
<td>0.20</td>
<td>5.50</td>
<td>2.14</td>
</tr>
<tr>
<td>2</td>
<td>8.30</td>
<td>0.78</td>
<td>−0.34</td>
<td>5.10</td>
<td>2.48</td>
</tr>
<tr>
<td>3</td>
<td>6.43</td>
<td>1.62</td>
<td>1.19</td>
<td>−2.90</td>
<td>4.62</td>
</tr>
<tr>
<td>4</td>
<td>0.35</td>
<td>3.98</td>
<td>2.14</td>
<td>−0.19</td>
<td>−2.72</td>
</tr>
<tr>
<td>5</td>
<td>1.85</td>
<td>0.61</td>
<td>1.60</td>
<td>−3.30</td>
<td>−0.58</td>
</tr>
<tr>
<td>6</td>
<td>−6.10</td>
<td>1.79</td>
<td>0.61</td>
<td>2.39</td>
<td>−0.24</td>
</tr>
</tbody>
</table>

| $\mu_i$      | 1.10 | 1.20 | 0.90 | 1.10 | 0.95 |
| $\rho_i$     | 4.43 | 1.27 | 0.74 | 3.23 | 2.13 |

**Table 1.4**: The values of $R_{is}$ of each stock $i$ in each scenario $s$, along with the expected rate of return $\mu_i$, and the mean absolute deviation $\rho_i$. All numbers are in percentages.

One way to deal with this uncertainty is to assume that, although we do not know the exact value of $R_i$ in advance, we know a number of possible scenarios that may happen. For example, we might define three scenarios, describing a bad outcome, an average outcome, and a good outcome. Let $S$ be the number of scenarios. We assume that all scenarios are equally likely to happen. For each $i$, let $R_{is}$ be the rate of return of stock $i$ in scenario $s$. Table 1.4 lists an example of values of $R_{is}$ for ($n = 5$) stocks and ($S = 6$) scenarios. For example, in scenario 1, the value of stock 1 decreases by 4.23% at the end of the month. So, in this scenario, an investment of $1 in stock 1 will be worth $(1 - 0.0423) = 0.9567$ at the end of the month. On the other hand, in scenario 2, this investment will be worth $(1 + 0.083) = 1.083$. Since we do not know in advance which scenario will actually happen, we need to base our decision upon all possible scenarios. One way to do this is to consider the so-called *expected rate of return* of stock $i$, which is denoted and defined by:

$$
\mu_i = \frac{1}{S} \sum_{s=1}^{S} R_{is}.
$$

The expected rate of return is the average rate of return, where the average is taken over all possible scenarios. It gives an indication of the rate of return at the end of the month. Since the value of $R_{is}$ is assumed to be known, the value of $\mu_i$ is known as well. Hence, it seems reasonable to select a portfolio of stocks that maximizes the total expected rate of return. However, there is usually a trade-off between the expected rate of return of a stock and the associated *risk*. A low-risk stock is a stock that has a rate of return that is close to its expected value in each scenario. Putting money on a bank account (or in a term-deposit) is an example of such a low-risk investment: the interest rate $r$ (expressed as a fraction, e.g., 2% corresponds to 0.02) is set in advance, and the investor knows that if $x$ is invested ($x \geq 0$), then after one month, this amount will have grown to $$(1 + r)x$$. In this case, the rate of return is the same for each scenario. On the other hand, the rate of return of a high-risk stock varies considerably among the different scenarios. Stock 1 in Table 1.4 is an example of a high-risk stock.
Risk may be measured in many ways. One of them is the mean absolute deviation. The mean absolute deviation of stock $i$ is denoted and defined as:

$$\rho_i = \frac{1}{S} \sum_{s=1}^{S} |R_{si} - \mu_i|.$$ 

So, $\rho_i$ measures the average deviation of the rate of return of stock $i$ compared to the expected rate of return of stock $i$.

For each $i = 1, \ldots, n$, define the following decision variable:

$$x_i = \text{the fraction of the$10,000 to be invested in stock } i.$$ 

Since the investor wants to invest the full $10,000, the $x_i$'s should satisfy the constraint:

$$x_1 + \ldots + x_n = 1.$$ 

If the investor simply wants the highest expected rate of return, then one may choose the stock $i$ that has the largest value of $\mu_i$ and set $x_i = 1$. However, this strategy is risky: it is better to diversify. So, the problem facing the investor is a so-called multiobjective optimization problem: on the one hand, the expected rate of return should be as large as possible; on the other hand, the risk should be as small as possible. Since stocks with a high expected rate of return usually also have a high risk, there is a trade-off between these two objectives; see also Chapter 14. If $x_i$ is invested in stock $i$, then it is straightforward to check that the expected rate of return $\mu$ of the portfolio as a whole satisfies:

$$\mu = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{n} R_{si} x_i = \sum_{i=1}^{n} \mu_i x_i. \quad (1.21)$$

The mean absolute deviation $\rho$ of a portfolio in which $x_i$ is invested in stock $i$ satisfies:

$$\rho = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{n} |R_{si} x_i - \mu| = \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{n} (R_{si} - \mu) x_i.$$ 

To solve the multiobjective optimization problem, we introduce a positive weight parameter $\lambda$ which measures how much importance we attach to maximizing the expected rate of return of the portfolio, relative to minimizing the risk of the portfolio. We choose the following objective:

$$\max \lambda (\text{expected rate of return}) - (\text{mean absolute deviation}).$$
Thus, the objective is $\max(\lambda \mu - \rho)$. The resulting optimization problem is:

$$
\begin{align*}
\max & \quad \lambda \sum_{i=1}^{n} \mu_i x_i - \frac{1}{S} \sum_{s=1}^{S} \left| \sum_{i=1}^{n} (R_i^s - \mu_i) x_i \right| \\
\text{s.t.} & \quad \sum_{i=1}^{n} x_i = 1 \\
& \quad x_1, \ldots, x_n \geq 0.
\end{align*}
$$

(1.22)

Although this is not an LO-model because of the absolute value operations, it can be turned into one using the technique of Section 1.5.2. To do so, we introduce, for each $s = 1, \ldots, S$, the decision variable $u_s$, and define $u_s$ to be equal to the expression $\sum_{i=1}^{n} (R_i^s - \mu_i) x_i$ inside the absolute value bars in (1.22). Note that $u_s$ measures the rate of return of the portfolio in scenario $s$. Next, we write $u_s = u_s^+ - u_s^-$ and $|u_s| = u_s^+ + u_s^-$. In Section 1.5.2, this procedure is explained in full detail. This results in the following LO-model:

$$
\begin{align*}
\max & \quad \lambda \sum_{i=1}^{n} \mu_i x_i - \frac{1}{S} \sum_{s=1}^{S} (u_s^+ + u_s^-) \\
\text{s.t.} & \quad \sum_{i=1}^{n} x_i = 1 \\
& \quad u_s^+ - u_s^- = \sum_{i=1}^{n} (R_i^s - \mu_i) x_i \quad \text{for } s = 1, \ldots, S \\
& \quad x_1, \ldots, x_n, u_s^+, u_s^- \geq 0 \quad \text{for } s = 1, \ldots, S.
\end{align*}
$$

(1.23)

Consider again the data in Table 1.4. As a validation step, we first choose the value of $\lambda$ very large, say $\lambda = 1000$. The purpose of this validation step is to check the correctness of the model by comparing the result of the model to what we expect to see. When the value of $\lambda$ is very large, the model tries to maximize the expected rate of return, and does not care much about minimizing the risk. As described above, in this case we should choose $x_j^* = 1$ where $j$ is the stock with the largest expected rate of return, and $x_i^* = 0$ for $i \neq j$. Thus, when solving the model with a large value of $\lambda$, we expect to see exactly this solution. After solving the model with $\lambda = 1000$ using a computer package, it turns out that this is indeed an optimal solution. On the other hand, when choosing $\lambda = 0$, the model tries to minimize the total risk. The optimal solution then becomes:

$$
\begin{align*}
x_1^* &= 0.036, \quad x_2^* = 0, \quad x_3^* = 0.761, \quad x_4^* = 0.151, \quad x_5^* = 0.052, \\
u_4^+ &= 0.0053, \quad u_1^+ = u_2^+ = u_3^+ = u_5^+ = u_6^+ = 0, \\
u_1^- &= 0.0018, \quad u_2^- = 0.0035, \quad u_3^- = u_6^- = 0, \\
u_4^- &= 0.0018.
\end{align*}
$$

The corresponding expected rate of return is 0.94%, and the corresponding average absolute deviation of the portfolio is 0.177%. This means that if the investor invests, at the beginning of the month, $360 in stock 1, $7,610 in stock 3, $1,510 in stock 4, and $520 in stock 5, then the investor may expect a 0.94% profit at the end of the month, i.e., the investor may expect that the total value of the stocks will be worth $10,094. Note, however, that the
actual value of this portfolio at the end of the month is not known in advance. The actual value can only be observed at the end of the month and may deviate significantly from the expected value. For example, if scenario 6 happens to occur, then the rate of return of the portfolio will be \((0.94 - 0.35 =) 0.59\%\); hence, the portfolio will be worth only \$10,059. On the other hand, if scenario 4 occurs, the portfolio will be worth \$10,147, significantly more than the expected value.

By setting the value of \(\lambda\) to a strictly positive number, the expected return and the risk are balanced. As an example, we choose \(\lambda = 5\). In that case, the following is an optimal solution:

\[
\begin{align*}
x_1^* &= 0, \ x_2^* = 0.511, \ x_3^* = 0, \ x_4^* = 0.262, \ x_5^* = 0.227, \\
u_2^+ &= 0.0118, \ u_4^+ = 0.0025, \ u_6^+ = 0.0037, \ u_1^- = u_3^- = u_5^- = 0, \\
u_5^- &= 0.018, \ u_1^- = u_2^- = u_3^- = u_4^- = u_6^- = 0,
\end{align*}
\]

with an expected return of 1.12\% (about 30\% higher than the minimum-risk portfolio) and an average absolute deviation of 0.6\% (more than three times as much as the minimum-risk portfolio).

In general, the choice of the value of \(\lambda\) depends on the investor. If the investor is very risk averse, a small value of \(\lambda\) should be chosen; if the investor is risk-seeking, a large value should be chosen. Since it is not clear beforehand what exact value should be chosen, we can solve the model for various values of \(\lambda\). Figure 1.9 shows the expected return and the risk for different optimal solutions with varying values of \(\lambda\). This figure illustrates the fact that as \(\lambda\) increases (i.e., as the investor becomes increasingly risk-seeking), both the expected return and the risk increase. The investor can now choose the portfolio that provides the combination between expected return and risk that meets the investors’ preferences best. The process in which we analyze how optimal values depend on the choice of a parameter (in this case, \(\lambda\)) is called sensitivity analysis, which is the main topic of Chapter 5.
1.7 Building and implementing mathematical models

In practical situations, designing mathematical models is only one part of a larger decision-making process. In such settings, designing a mathematical model is therefore not a goal by itself. The mathematical models form a tool to help make a suitable and informed decision. In this section, we discuss mathematical models in the broader setting of the decision-making process.

This decision-making process usually involves a team of people from various disciplines, e.g., business managers and subject experts. Among the subject experts, there may be one or more operations research experts. The objective of the operations research expert is to design useful mathematical models that are appropriate for the real-life problem at hand. Since the expert needs to work closely with the other team members, the job requires not only the appropriate technical and mathematical knowledge, but also social skills.

Complicated practical problems can be investigated by using models that reflect the relevant features of the problem, in such a way that properties of the model can be related to the real (actual) situation. One of the main decisions to be made by the operations research expert is which features of the problem to take into account, and which ones to leave out. Clearly, the more features are taken into account, the more realistic the model is. On the other hand, taking more features into account usually complicates the interpretation of the model results, in addition to making the model harder to solve. Choosing which features to take into account usually means finding a good balance between making the model sufficiently realistic on the one hand, and keeping the model as simple as possible on the other hand. There are no general rules to decide which features are appropriate, and which ones are not. Therefore, mathematical modeling can be considered as an art.

Throughout the decision process, the following questions need to be asked. What is the problem? Is this really the problem? Can we solve this problem? Is mathematical optimization an appropriate tool to solve the problem, or should we perhaps resort to a different mathematical (or nonmathematical) technique? Once it is clear that mathematical optimization is appropriate, other questions need to be answered. What data do we need to solve the problem? Once we have found a solution, the question then arises whether or not this is a useful solution. How will the solution be implemented in practice? By ‘solving a problem’ we mean the successful application of a solution to a practical problem, where the solution is a result from a mathematical model.

The following steps provide some guidelines for the decision-making process; usually these steps do not have to be made explicit during the actual model building and implementing process. Usually, there is a certain budget available for the completion of the decision process. During the whole process, the budget has to be carefully monitored.

Algorithm 1.7.1. (Nine steps of the decision process)

- Step 1: Problem recognition;
We will discuss each of the nine steps in more detail.

- **Step 1. Problem recognition.**
  The recognition of a problem precedes the actual procedure. It may start when a manager in a business does not feel quite happy about the current situation, but does not exactly know what is wrong. More often than not, the issue simmers for a while, before the manager decides to take action. For instance, a planner discovers that orders are delivered late, monitors the situation for a couple of weeks, and, when the situation does not change, informs the logistics department.

- **Step 2. Problem definition.**
  The next step in the decision-making process concerns the definition of the problem. For example, the definition of the problem is: ‘The warehouse is too small’, or ‘The orders are not manufactured before their due dates’. This step is usually carried out by a small team of experts. This team discusses and reconsiders the problem definition, and – if necessary – improves it. An important aspect is setting the goals of the decision-making process. A provisional goal could be to improve the insight and understanding of the complex situation of the problem.

- **Step 3. Observation and problem analysis.**
  Since a problem usually occurs in large (sub)systems, it is necessary to observe and analyze the problem in the context of the company as a whole. It is important to communicate with other experts within and outside the organization. Relevant data need to be collected and analyzed to understand all aspects of the problem. It may happen that it is necessary to return to Step 2 in order to revise the problem definition. At the end of Step 3 the structure of the problem should become clear.

- **Step 4. Designing a conceptual model; validation.**
  During Step 3, the relevant information has become available. It is now clear which aspects and parameters are of importance and should be taken into account for further considerations. However, this information may not be enough to solve the problem. The conceptual model reflects the relationships between the various aspects or parameters of the problem, and the objective that has to be reached. Usually, not all aspects can be included. If vital details are omitted, then the model will not be realistic. On the other
hand, if too many details are included, then attention will be distracted from the crucial factors and the results may become more difficult to interpret. The trade-off between the size of the model (or, more precisely, the time to solve the model) and its practical value must also be taken into account. Conceptual models have to be validated, which means investigating whether or not the model is an accurate representation of the practical situation. The conceptual model (which usually does not contain mathematical notions) can be discussed with experts inside and outside the company. In case the model does not describe the original practical situation accurately enough, the conceptual model may need to be changed, or one has to return to Step 3. At the end of Step 4, the project team may even decide to change the problem definition, and to return to Step 2.

▶ **Step 5. Designing a mathematical model; verification.**
In this step the conceptual relationships between the various parameters are translated into mathematical relationships. This is not always a straightforward procedure. It may be difficult to find the right type of mathematical model, and to formulate the appropriate specification without an overload of variables and constraints. If significant problems occur, it may be useful to return to Step 3, and to change the conceptual model in such a way that the mathematical translation is less difficult. If the project team decides to use a linear optimization model, then the relationships between the relevant parameters should be linear. This is in many situations a reasonable restriction. Instead of solving the whole problem immediately, it is usually better to start with smaller subproblems; this may lead to new insights and to a reformulation of the conceptual model. The mathematical model has to be verified, which means determining whether or not the model performs as intended, and whether or not the computer running time of the solution technique that is used is acceptable (see also Chapter 9). This usually includes checking the (in)equalities and the objective of the model. Another useful technique is to compute solutions of the model for a number of extreme worst-case instances, or for data for which realizations are already known. Sensitivity analysis on a number of relevant parameters may also provide insight into the accuracy and robustness of the model. These activities may lead to a return to Step 3 or Step 4.

▶ **Step 6. Solving the mathematical model.**
Sometimes a solution of the mathematical model can be found by hand, without using a computer. In any case, computer solution techniques should only be used after the mathematical model has been carefully analyzed. This analysis should include looking for simplifications which may speed up the processing procedures, analyze the models outputs thoroughly, and carry out some sensitivity analysis. Sensitivity analysis may, for instance, reveal a strong dependence between the optimal solution and a certain parameter, in which case the conceptual model needs to be consulted in order to clarify this situation. Moreover, sensitivity analysis may provide alternative (sub)optimal solutions which can be used to satisfy constraints that could not be included in the mathematical model.

▶ **Step 7. Taking a decision.**
As soon as Step 6 has been finished, a decision can be proposed. Usually, the project team
formulates a number of alternative proposals together with the corresponding costs and other business implications. In cooperation with the management of the business, it is decided which solution will be implemented or, if implementation is not possible, what should be done otherwise. Usually, this last situation means returning to either Step 4, or Step 3, or even Step 2.

▶ Step 8. Implementing the decision.
Implementing the decision requires careful attention. The expectations that are set during the decision process have to be realized. Often a pilot-project can be started to detect any potential implementation problems at an early stage. The employees that have to work with the new situation, need to be convinced of the effectiveness and the practicability of the proposed changes. Sometimes, courses need to be organized to teach employees how to deal with the new situation. One possibility to prevent problems with the new implementation is to communicate during the decision process with those involved in the actual decision and execution situation.

During the evaluation period, the final checks are made. Questions to be answered are: Does everything work as intended? Is the problem solved? Was the decision process organized well? Finally, the objectives of the company are compared to the results of the implemented project.

1.8 Exercises

Exercise 1.8.1. In Section 1.2.1, we defined the standard form of an LO-model. Write each of the following LO-models in the standard form \( \max \{ c^T x \mid Ax \leq b, x \geq 0 \} \).

(a) \[ \begin{align*}
\min & \quad 25x_1 + 17x_2 \\
\text{s.t.} & \quad 0.21x_1 + 0.55x_2 \geq 3 \\
& \quad 0.50x_1 + 0.30x_2 \geq 7 \\
& \quad 0.55x_1 + 0.10x_2 \geq 5 \\
& \quad x_1, x_2 \text{ free}
\end{align*} \]

(b) \[ \begin{align*}
\min & \quad 12x + |5y| \\
\text{s.t.} & \quad x + 2y \geq 4 \\
& \quad 5x + 6y \leq 7 \\
& \quad 8x + 9y = 5 \\
& \quad x, y \text{ free}
\end{align*} \]

(c) \[ \begin{align*}
\min & \quad 3x_1 + 4x_2 \\
\text{s.t.} & \quad x_1 + x_2 + x_3 + x_4 = 10 \\
& \quad x_1 - 2x_2 + 3x_3 = 6 \\
& \quad 0 \leq x_1 \leq 4, 0 \leq x_2 \leq 4 \\
& \quad 0 \leq x_3 \leq 4, 0 \leq x_4 \leq 12
\end{align*} \]

(d) \[ \begin{align*}
\min & \quad |x_1| + |x_2| + |x_3| \\
\text{s.t.} & \quad x_1 - 2x_2 = 3 \\
& \quad -x_2 + x_3 \leq 1 \\
& \quad x_1, x_2, x_3 \text{ free}
\end{align*} \]

Exercise 1.8.2. Show that an LO-model of the form

\[ \max \{ c_1^T x_1 + c_2^T x_2 \mid A_1 x_1 = b_1, A_2 x_2 = b_2, x_1 \geq 0, x_2 \geq 0 \} \]
with \( c_i \in \mathbb{R}^{n_i}, x_i \in \mathbb{R}^{n_i}, A_i \in \mathbb{R}^{m_i \times n_i}, b_i \in \mathbb{R}^{m_i} \) \((i = 1, 2)\), can be solved by solving the two models

\[
\max \{c_1^T x_1 \mid A_1 x_1 = b_1, x_1 \geq 0\} \quad \text{and} \quad \max \{c_2^T x_2 \mid A_2 x_2 = b_2, x_2 \geq 0\}.
\]

Given an optimal solution of these two, what is the optimal solution for the original model, and what is the corresponding optimal objective value?

**Exercise 1.8.3.** Solve the following LO-models by using the graphical solution method.

(a) \[
\begin{align*}
\max \quad & x_1 + 2x_2 \\
\text{s.t.} \quad & -x_1 + 2x_2 \leq 4 \\
& 3x_1 + x_2 \leq 9 \\
& x_1 + 4x_2 \geq 4 \\
& x_1, x_2 \geq 0
\end{align*}
\]

(b) \[
\begin{align*}
\min \quad & 11x_1 + 2x_2 \\
\text{s.t.} \quad & x_1 - x_2 \leq 4 \\
& 15x_1 - 2x_2 \geq 0 \\
& 5x_1 + x_2 \geq 5 \\
& 2x_1 + x_2 \geq 3 \\
& x_1, x_2 \geq 0
\end{align*}
\]

(c) \[
\begin{align*}
\max \quad & 2x_1 + 3x_2 - 2x_3 + 3x_4 \\
\text{s.t.} \quad & x_1 + x_2 \leq 6 \\
& 2x_3 + 3x_4 \leq 12 \\
& -x_3 + x_4 \geq -2 \\
& 2x_1 - x_2 \leq 4 \\
& 2x_3 - x_4 \geq -1
\end{align*}
\]

(d) \[
\begin{align*}
\max \quad & 3x_1 + 7x_2 + 4x_3 - 3x_4 \\
\text{s.t.} \quad & x_1 + 2x_2 \leq 6 \\
& x_3 + x_4 \leq 6 \\
& 4x_1 + 5x_2 \leq 20 \\
& -2x_1 + x_2 \leq 1 \\
& 2x_3 - x_4 \leq 4 \\
& x_3 \geq 2
\end{align*}
\]

\[
0 \leq x_2 \leq 2, \quad x_1, x_3, x_4 \geq 0
\]

Exercise 1.8.4. The LO-models mentioned in this chapter all have ‘\(\leq\)’, ‘\(\geq\)’, and ‘\(=\)’ constraints, but no ‘\(<\)’ or ‘\(>\)’ constraints. The reason for this is the fact that models with such constraints may not have an optimal solution, even if the feasible region is bounded.

Show this by constructing a bounded model with ‘\(<\)’ and/or ‘\(>\)’ constraints, and argue that the constructed model does not have an optimal solution.

**Exercise 1.8.5.** Consider the constraints of Model Dovetail in Section 1.1. Determine the optimal vertices in the case of the following objectives:

(a) \[
\max 2x_1 + x_2
\]

(b) \[
\max x_1 + 2x_2
\]

(c) \[
\max \frac{3}{2} x_1 + \frac{1}{2} x_2
\]

**Exercise 1.8.6.** In Section 1.1.2 the following optimal solution of Model Dovetail is found:

\[
x_1^* = 4 \frac{1}{2}, \quad x_2^* = 4 \frac{1}{2}, \quad x_3^* = 0, \quad x_4^* = 0, \quad x_5^* = 2 \frac{1}{2}, \quad x_6^* = 1 \frac{1}{2}.
\]
What is the relationship between the optimal values of the slack variables $x_3, x_4, x_5, x_6$ and the constraints (1.1), (1.2), (1.3), (1.4)?

Exercise 1.8.7. Let $\alpha, \beta \in \mathbb{R}$. Consider the LO-model:

$$\begin{align*}
\min & \quad x_1 + x_2 \\
\text{s.t.} & \quad \alpha x_1 + \beta x_2 \geq 1 \\
& \quad x_1 \geq 0, \ x_2 \text{ free}.
\end{align*}$$

Determine necessary and sufficient conditions for $\alpha$ and $\beta$ such that the model

(a) is infeasible,
(b) has an optimal solution,
(c) is feasible, but unbounded,
(d) has multiple optimal solutions.

Exercise 1.8.8. Solve the following LO-model with the graphical solution method.

$$\begin{align*}
\max & \quad 2x_1 + 3x_2 - 4x_3 \\
\text{s.t.} & \quad -x_1 - 5x_2 + x_3 = 4 \\
& \quad -x_1 - 3x_2 + x_4 = 2 \\
& \quad x_1, x_2, x_3, x_4 \geq 0.
\end{align*}$$

Exercise 1.8.9. Show that if $x^* = [u_1^* \ u_2^* \ x_2^*]^T$ is an optimal solution of (1.14), then $[x_1^* \ x_2^*]^T$ with $x_1^* = u_1^* - u_2^*$ is an optimal solution of (1.11).

Exercise 1.8.10. The method of Section 1.5.2 to write a model with an absolute value as an LO-model does not work when the objective is to maximize a function in which an absolute value appears with a positive coefficient. Show this by considering the optimization model $\max \{ |x| \mid -5 \leq x \leq 5 \}.$

Exercise 1.8.11. Show that any convex piecewise linear function is continuous.

Exercise 1.8.12. The method of Section 1.5.3 to write a model with a piecewise linear function as an LO-model fails if the piecewise linear function is not convex. Consider again model (1.16), but with the following function $f$:

$$f(x_1) = \begin{cases} 
  x_1 & \text{if } 0 \leq x_1 \leq 3 \\
  3 + 3(x_1 - 3) & \text{if } 3 < x_1 \leq 8 \\
  18 + (x_1 - 8) & \text{if } x_1 > 18.
\end{cases}$$
This function is depicted in Figure 1.10. What goes wrong when the model is solved using the solution technique of Section 1.5.3?

Exercise 1.8.13. Consider the team formation model in Section 1.6.3. Take any \( p \in \{1, \ldots, N\} \). For each of the following requirements, modify the model to take into account the additional requirement:

(a) Due to an injury, player \( p \) cannot be part of the line-up.
(b) Player \( p \) has to play on position 1.
(c) Due to contractual obligations, player \( p \) has to be part of the line-up, but the model should determine at which position.
(d) Let \( A \subseteq \{1, \ldots, N\} \) and \( B \subseteq \{1, \ldots, M\} \). The players in set \( A \) can only play on positions in set \( B \).

Exercise 1.8.14. Consider the data envelopment analysis example described in Section 1.6.4. Give an example that explains that if a DMU is relatively efficient according to the DEA solution, then this does not necessarily mean that it is 'inherently efficient'. (Formulate your own definition of 'inherently efficient'.) Also explain why DMUs that are relatively inefficient in the DEA approach, are always 'inherently inefficient'.

Exercise 1.8.15. One of the drawbacks of the data envelopment analysis approach is that all DMUs may turn out to be relatively efficient. Construct an example with three different DMUs, each with three inputs and three outputs, and such that the DEA approach leads to three relatively efficient DMUs.

Exercise 1.8.16. Construct an example with at least three DMUs, three inputs and three outputs, and such that all but one input and one output have weights equal to \( \varepsilon \) in the optimal solution of model (\( M_2(k) \)).
Exercise 1.8.17. Consider model \((M_2(k))\). Show that there is always at least one DMU that is relatively efficient. (Hint: show that if all ‘\(\leq\)’-constraints of \((M_2(k))\) are satisfied with strict inequality in the optimal solution of \((M_2(k))\), then a ‘more optimal’ solution of \((M_2(k))\) exist, and hence the optimal solution was not optimal after all.)

Exercise 1.8.18. Consider the following data for three baseball players.

<table>
<thead>
<tr>
<th>Players</th>
<th>Bats</th>
<th>Hits</th>
<th>Home runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>123</td>
<td>39</td>
<td>7</td>
</tr>
<tr>
<td>John</td>
<td>79</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>James</td>
<td>194</td>
<td>35</td>
<td>5</td>
</tr>
</tbody>
</table>

We want to analyze the efficiency of these baseball players using data envelopment analysis.

(a) What are the decision making units, and what are the inputs and outputs?
(b) Use DEA to give an efficiency ranking of the three players.

Exercise 1.8.19. Consider again the portfolio selection problem of Section 1.6.5. Using the current definition of risk, negative deviations from the expected rate of return have the same weight as positive deviations. Usually, however, the owner of the portfolio is more worried about negative deviations than positive ones.

(a) Suppose that we give the negative deviations a weight \(\alpha\) and the positive deviations a weight \(\beta\), i.e., the risk of the portfolio is:

\[
\rho = \frac{1}{S} \sum_{s=1}^{S} |f(R_i^s - \mu_i)|,
\]

where \(f(x) = -\alpha x\) if \(x < 0\) and \(f(x) = \beta x\) if \(x \geq 0\). Formulate an LO-model that solves the portfolio optimization problem for this definition of risk.

(b) What conditions should be imposed on the values of \(\alpha\) and \(\beta\)?

Exercise 1.8.20. When plotting data from a repeated experiment in order to test the validity of a supposed linear relationship between two variables \(x_1\) and \(x_2\), the data points are usually not located exactly on a straight line. There are several ways to find the ‘best fitting line’ through the plotted points. One of these is the method of least absolute deviation regression.

In least absolute deviation regression it is assumed that the exact underlying relationship is a straight line, say \(x_2 = \alpha x_1 + \beta\). Let \(\{[a_1, b_1], \ldots, [a_n, b_n]\}\) be the data set. For each \(i = 1, \ldots, n\), the absolute error, \(r_i\), is defined by \(r_i = a_i - b_i\alpha - \beta\). The problem is to determine values of \(\alpha\) and \(\beta\) such that the sum of the absolute errors, \(\sum_{i=1}^{n} |r_i|\), is as small as possible.

(a) Formulate the method of least absolute deviation regression as an LO-model.
(b) Determine the least absolute deviation regression line $x_2 = \alpha x_1 + \beta$ for the following data set: $\{[1, 1], [2, 2], [3, 4]\}$.

(c) Table 1.5 contains the ages and salaries of twenty employees of a company. Figure 1.11 shows a scatter plot of the same data set. Determine the least absolute deviation regression line $x_2 = \alpha x_1 + \beta$ for the data set, where $x_1$ is the age, and $x_2$ is the salary of the employee.

(d) Modify the model so that instead of minimizing the sum of the absolute deviations, the maximum of the absolute deviations is minimized. Solve the model for the data set in Table 1.5.

Exercise 1.8.21. The new budget airline CheapNSafe in Europe wants to promote its brand name. For this promotion, the company has a budget of €95,000. CheapNSafe hired a marketing company to make a television advertisement. It is now considering buying 30-second advertising slots on two different television channels: ABC and XYZ. The number of people that are reached by the advertisements clearly depends on how many slots are bought. Fortunately, the marketing departments of ABC and XYZ have estimates of how many people are reached depending on the number of slots bought. ABC reports that the first twenty slots on their television channel reach 20,000 people per slot; the next twenty slots reach another 10,000 each; any additional slots do not reach any additional people. So,
for example, thirty slots reach \((20 \times 20,000 + 10 \times 10,000 =) 500,000\) people. One slot on ABC costs \(€2,000\). On the other hand, XYZ reports that the first fifteen slots on XYZ reach 30,000 people each; the next thirty slots reach another 15,000 each; any additional slots do not reach any additional people. One slot on XYZ costs \(€3,000\).

Suppose that CheapNSafe’s current objective is to maximize the number of people reached by television advertising.

(a) Draw the two piecewise linear functions that correspond to this model. Are the functions convex or concave?

(b) Assume that people either watch ABC, or XYZ, but not both. Design an LO-model to solve CheapNSafe’s advertising problem.

(c) Solve the LO-model using a computer package. What is the optimal advertising mix, and how many people does this mix reach?

(d) XYZ has decided to give a quantity discount for its slots. If a customer buys more than fifteen slots, the slots in excess of fifteen cost only \(€2,500\) each. How can the model be changed in order to incorporate this discount? Does the optimal solution change?