Chapter 1

Linear Programming Formulations

1.1 ▪ Definition of Linear Programming Problems

In this section, we will present the concept of linear programming problems, which falls under the broader field of "optimization" or "mathematical programming". These terms refer to the mathematical discipline that focuses on solving problems involving the minimization or maximization of a function under given constraints. Let us start by defining a general optimization problem.

1.1.1 ▪ General Optimization Problems

A general optimization problem can be formulated as

$$
\text{(P)} \qquad \begin{array}{ll} \min & f(x_1, x_2, \ldots, x_n) \\ \text{s.t.} & g_1(x_1, x_2, \ldots, x_n) \leq b_1, \\ & \qquad \vdots \\ & g_m(x_1, x_2, \ldots, x_n) \leq b_m, \\ & h_1(x_1, x_2, \ldots, x_n) = d_1, \\ & \qquad \vdots \\ & h_p(x_1, x_2, \ldots, x_n) = d_p, \end{array}
$$

where "s.t." is short for "subject to", and in addition

- $x_1, x_2, \ldots, x_n$ are the **decision variables**,
- f is the **objective function**,
- $g_i, i = 1, \ldots, m$, and $h_j, j = 1, \ldots, p$, are the **constraint functions**,
- the first m constraints are **inequality constraints**,
- the last p constraints are **equality constraints**.

For example, consider the problem

$$
\begin{array}{ll} \min & x_1^2 - x_2 + 1 \\ \text{s.t.} & x_1 + x_2 \leq 0, \\ & x_2^2 = 0. \end{array} \tag{1.1}
$$

In this problem, x_1 and x_2 are the decision variables, $f(x_1, x_2) = x_1^2 - x_2 + 1$ is the objective function, $g_1(x_1, x_2) = x_1 + x_2$ is a constraint function corresponding to the inequality constraint, and $h_1(x_1, x_2) = x_2^2$ is a constraint function corresponding to the equality constraint.

We seek to find the "best values" of x_1 and x_2 in the sense that they correspond to the smallest objective function value and satisfy the two constraints. Of course, solving problem (1.1) is a simple task since the second constraint enforces $x_2 = 0$, reducing the problem to

$$\begin{array}{ll} \min & x_1^2 + 1 \\ \text{s.t.} & x_1 \leq 0. \end{array}$$

The optimal solution to the above problem is obviously $x_1 = 0$. To summarize, the optimal solution of problem (1.1) is $(x_1, x_2) = (0, 0)$, and the minimum objective function value is 1.

We will now formally define the basic concepts of optimality and feasibility for the general minimization problem (P).

Definition 1.1.

1. *A* **feasible solution** *of problem (P) is a vector $(x_1, x_2, \ldots, x_n)$ that satisfies all the constraints of (P).*
2. *The* **feasible set** *of problem (P) is the set of all feasible solutions of (P).*
3. *An* **optimal solution** *of problem (P) is a feasible solution $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ satisfying that*

   $$f(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \leq f(x_1, x_2, \ldots, x_n)$$

   for any feasible solution $(x_1, x_2, \ldots, x_n)$ of problem (P).
4. *The* **optimal set** *of problem (P) is the set of all optimal solutions of (P).*
5. *The* **optimal value** *of problem (P) is the value of the objective function at optimal solutions of the problem, meaning $f(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, where $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ is an optimal solution of the problem.*

For example, in problem (1.1), the following hold:

- $(1, 1)$ is not a feasible solution since it does not satisfy any of the constraints.
- $(-1, 1)$ is not a feasible solution since it does not satisfy the equality constraint.
- $(-2, 0)$ is a feasible solution since it satisfies both constraints. It is not an optimal solution since its objective function value is 5, while $(0, 0)$, which is also a feasible solution, has an objective function value of 1. Actually, we just showed that $(0, 0)$ is an optimal solution, and therefore, the optimal value of problem (1.1) is 1.

Remark 1.2.

1. *Any maximization problem can be turned into a minimization problem using the following formula:*

   $$\max f(\mathbf{x}) = -\min(-f(\mathbf{x})).$$

   This formula is somewhat schematic and its meaning is as follows: instead of seeking the feasible solution that maximizes f, we will find a feasible solution that minimizes $-f$ (under the same constraints). The problems are equivalent in the sense that the optimal set *of the minimization problem is the same as the optimal set of the maximization problem; however, the optimal value is not the same, but opposite in sign. In more detail, if* $\mathbf{x}^*$ *is an optimal solution of the minimization problem, then it is also an optimal solution of the maximization problem and vice versa—any optimal solution of the maximization problem is also an optimal solution of the minimization problem. The optimal value of the minimization problem is* $v = -f(\mathbf{x}^*)$*, while* $-v = f(\mathbf{x}^*)$ *is the optimal value of the maximization problem.*

2. *In the description of problem (P), all the inequality constraints were "less-than-or-equal-to" constraints, meaning constraints of the form* $g(\mathbf{x}) \leq b$*. In fact, formulation (P) also implicitly encompasses greater-than-or-equal-to constraints since the constraint* $g(\mathbf{x}) \geq b$ *is equivalent to the constraint* $-g(\mathbf{x}) \leq -b$ *(after multiplication by* -1*). Note that all the inequalities we consider are weak inequalities (meaning greater-than-or-equal-to or smaller-than-or-equal-to), and we will never consider strict inequalities (greater-than or smaller-than).*

There are special constraints, which usually appear at the end of the formulation:

(a) **Nonnegativity constraints**—constraints of the form $x_j \geq 0$ for some (or all) the variables.

(b) **Box constraints**—constraints of the form $\ell_j \leq x_j \leq u_j$, where $l_j \leq u_j$ are two real numbers.

(c) **Integer constraints**—constraints of the form "x_j is integer".

A priori, all variables in optimization problems are assumed to be able to attain any value in the real line, and as such, they are referred to as **continuous variables**. Obviously, the constraints of the problem might limit the range of individual variables. A variable that can a priori attain only a finite or countably infinite number of values is called a **discrete variable**. For example, the following constraints on x_j imply that it is a discrete variable:

- $x_j \in \mathbb{Z}$ – x_j is integer.
- $x_j \in \mathbb{N}$ – x_j is a nonnegative integer.
- $x_j \in \{0, 1\}$ – x_j is binary.
- $x_j \in \{0, 1, \ldots, K\}$ $(K \in \mathbb{N})$ – x_j attains one of the integer values between 0 and K (inclusive).

In this book we will deal exclusively with problems with a finite number of constraints. We will not address problems with an infinite number of constraints (although they do exist!) or *unconstrained problems*, which are problems without any constraints.

1.1.2 ▪ Linear Programming Problems

Definition 1.3 (linear function). *A function* $f : \mathbb{R}^n \to \mathbb{R}$ *is called* **linear** *if there exist* $a_1, a_2, \ldots, a_n \in \mathbb{R}$ *for which*

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \tag{1.2}$$

for any $x_1, x_2, \ldots, x_n \in \mathbb{R}$.

If we define the vectors $\mathbf{a}, \mathbf{x}$ as

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix},$$

then the linear function given in (1.2) can be written as $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ or $f(\mathbf{x}) = \mathbf{x}^T\mathbf{a}$.

Definition 1.4 (linear constraint). *A constraint is called* **linear** *if its associated constraint function is linear.*

Thus, for example, the constraint

$$x_1 + x_2 - x_3 \leq 2$$

is a linear inequality constraint, and the constraint

$$2x_1 - x_2 + 3x_3 = 5$$

is a linear equality constraint. We are now ready to define the notion of linear programming problems.

Definition 1.5 (linear programming problem). *A* **linear programming (LP) problem** *is an optimization problem with a linear objective function and a finite number of linear constraints.*

Example 1.6. The problem

$$\begin{array}{ll} \max & x_1 - x_2 + x_3 \\ \text{s.t.} & x_1 + 2x_2 \leq 3, \\ & x_1 + x_2 + x_3 = 4, \\ & x_1, x_2, x_3 \geq 0, \end{array}$$

is a linear programming problem since its objective function is linear and it has a finite number of linear constraints. In particular, the problem has five constraints; among them are a linear inequality constraint, a linear equality constraint, and three additional linear inequality constraints which are nonnegativity constraints on the decision variables. ∎

Example 1.7. The optimization problem

$$\begin{array}{ll} \max & x_1^2 - x_2 + x_3 \\ \text{s.t.} & x_1 + 2x_2 \leq 3, \\ & x_1 + x_2 + x_3 = 4, \\ & x_1, x_2, x_3 \geq 0, \end{array}$$

is not a linear programming problem since its objective function is not a linear function. The problem

$$\begin{array}{ll} \max & x_1 - x_2 + x_3 \\ \text{s.t.} & x_1^2 + 2x_2 \leq 3, \\ & x_1 + x_2 + x_3 = 4, \\ & x_1, x_2, x_3 \geq 0, \end{array}$$

is also not a linear programming problem since its first constraint is not a linear function. ∎

Remark 1.8. *In a linear programming problem all the variables are continuous, and in particular, there are no integer variables since there is no way to represent integrality constraints as a set of linear constraints.*

1.2 ▪ Formulations as Linear Programming Problems

In this section, we will introduce various problems that can be formulated as LP problems. The modeling phase, which is the initial step in addressing optimization problems, will be discussed. Subsequently, in this chapter and in the upcoming chapters, we will delve into the second phase: solving the formulated mathematical models.

1.2.1 ▪ The Diet Problem

In this problem there are several food products and several nutrients. The data includes the following information:

- Number of units of each nutrient that one kilogram of each food product contains ("kilogram" can also be replaced by any other unit).
- The minimum amount of each nutritional component that must be consumed.
- The price of a kilogram of each of the food products.

The goal is to build the cheapest menu that will meet the minimum health needs, that is, provide the required amount of each of the nutritional components. We will start with an example.

The Diet Problem (Special Case). Consider two food products—corn and wheat, and three nutrients—minerals, protein, and vitamins. The following table contains the necessary data. For example, 1 kilogram of wheat contains 1 unit of minerals, 6 units of protein, and 1 unit of vitamin, and its price is 0.4 dollars.

	food products		minimum required quantity
	wheat	corn	
minerals	1	1	3
protein	6	2	6
vitamins	1	5	5
cost of 1 kg.	0.4	0.6	

The goal is to build a menu as cheap as possible that meets the health requirements, meaning it provides at least 3 units of minerals, 6 units of protein, and 5 units of vitamins.

Decision variables. We begin by defining the decision variables:
x_1 – number of kilograms of wheat in the menu.
x_2 – number of kilograms of corn in the menu.

Objective function. Since each kilogram of wheat costs 0.4 dollars and the number of kilograms of wheat in the menu is x_1, it follows that the cost of wheat in the menu is $0.4x_1$. Similarly, the cost of corn in the menu is $0.6x_2$, so the total cost of the menu is $0.4x_1 + 0.6x_2$, which is the objective function of the minimization problem.

Constraints. Each kilogram of wheat provides one unit of minerals. The number of kilograms of wheat in the menu is x_1, providing x_1 units of minerals. Likewise, each kilogram of corn provides one unit of minerals. Therefore, from x_2 kilograms of corn we can extract x_2 units of minerals. Consequently, the total amount of minerals in the menu is $x_1 + x_2$. As per the specified requirements, since this quantity should be at least 3, the corresponding constraint for the minerals requirement can be expressed as $x_1 + x_2 \geq 3$. Similarly, the minimum requirement for protein leads to the constraint $6x_1 + 2x_2 \geq 6$, and the minimum requirement for vitamins results in the constraint $x_1 + 5x_2 \geq 5$.

Furthermore, since the variables represent quantities of kilograms to be purchased, they must be nonnegative. Therefore, the nonnegativity constraints $x_1 \geq 0$ and $x_2 \geq 0$ are included.

Overall, the formulation of the problem as a linear programming problem is as follows:

$$\begin{array}{ll} \min & 0.4x_1 + 0.6x_2 \\ \text{s.t.} & x_1 + x_2 \geq 3, \\ & 6x_1 + 2x_2 \geq 6, \\ & x_1 + 5x_2 \geq 5, \\ & x_1, x_2 \geq 0. \end{array} \tag{1.3}$$

Later in the chapter, in Section 1.3.4, we will see how this particular problem can be solved and that its optimal solution is $x_1 = 2.5, x_2 = 0.5$ with an optimal value of 1.3.

In general, there could be more than two food products and more than three nutrients involved in the problem. Hence, we will now proceed to describe and formulate the diet problem in its general setting.

The Diet Problem (General Case). Consider m food products and n nutrients with the following given data:

- c_j – cost of one kilogram of food product j $(j = 1, 2, \dots, n)$.
- b_i – minimum amount of nutrient i required in the menu $(i = 1, 2, \dots, m)$.
- a_{ij} – amount of nutrient i in one kilogram of food product j $(i = 1, 2, \dots, m, j = 1, 2, \dots, n)$.

The goal is to build a menu as cheap as possible that meets the health requirements, that is, providing all the required amounts of each nutrient.

Decision variables.

x_j – number of kilograms of food product j in the menu ($j = 1, 2, \dots, n$).

Problem formulation.

$$\begin{array}{ll} \min & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \; i = 1, 2, \dots, m, \\ & x_j \geq 0, \; j = 1, 2, \dots, n. \end{array}$$

Denote the $m \times n$ matrix $(a_{ij})_{i,j}$ by $\mathbf{A}$, and the vectors $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{x} \in \mathbb{R}^n$ as

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Then the problem can be rewritten in matrix form as follows:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{1.4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the n-length column vector comprising the n decision variables $x_1, x_2, \ldots, x_n$. Note that each of the nutrients defines a constraint (a total of m constraints), and each of the food products corresponds to a decision variable (a total of n variables) where each variable also corresponds to a nonnegativity constraint (additional n constraints). Obviously, the diet problem in its general form (1.4) is an LP problem.

The diet problem was one of the first optimization problems studied in the 1930s and 1940s. It came from the United States army's requirement to meet the soldiers' nutritional requirements at a minimal cost. One of the first researchers to study this problem was George Stigler. He arrived at a solution with the help of a heuristic[a] method with a value of $39.93 (cost per soldier per year at 1939 prices). In 1947 Jack Laderman solved the problem using the simplex method that will be discussed in Chapter 3. This was the first time a "large-scale" calculation had been carried out. The problem consisted of 77 variables and 9 constraints. Nine clerks, each with a mechanical computer, worked on its solution, and after 120 person-days, they found an optimal solution of $39.69 (an annual savings of 24 cents). The fact that Stigler found a solution whose value was far from the optimal value by less than a percent is undoubtedly impressive, but it is quite probable that if the problem was truly large-scale (e.g., thousands of variables and constraints), his success would have been more limited. Today, linear programming problems with hundreds of thousands of constraints and variables can be easily solved on standard personal computers in fractions of a second.

[a] A solution method is called "heuristic" if it is not a method that guarantees finding an optimal solution. It is usually a method based on intuition and logical rules of thumb. As soon as it is proven that a certain heuristic method yields an optimal solution, it effectively ceases to be a heuristic method.

1.2.2 ▪ The Production Problem

The problem involves a factory that manufactures multiple products, each constructed from a combination of various raw materials. The problem data is as follows:

- The quantities of raw materials required to produce one unit of each product.
- Amounts of raw materials available to the factory.
- Cost of raw materials.
- The selling price of each of the products.

The goal is to decide how much to produce from each of the products to maximize the factory's profit, taking into account the limited stock of each raw material. Let us start with a small example.

The Production Problem (Special Case). A restaurant should determine the quantities of chicken soup and vegetable soup it produces. The restaurant has the following information:

- Chicken soup composition: one liter of chicken soup requires 3 kilograms of chicken and 1 kilogram of vegetables.
- Vegetable soup composition: one liter of vegetable soup requires 2 kilograms of chicken and 6 kilograms of vegetables.
- Costs: A kilogram of chicken costs \$5, and a kilogram of vegetables costs \$1.
- Selling price: A liter of chicken soup is sold for \$23, and a liter of vegetable soup is sold for \$27.
- Resource constraints: The restaurant can purchase a maximum of 12 kilograms of chicken and 12 kilograms of vegetables.

How many kilograms of soup of each type should the restaurant produce to maximize its profit?
Disclaimer: Please do not regard this data as a suggestion for real recipes ⌣̈.

Decision variables.
x_1 – number of liters of chicken soup the restaurant produces.
x_2 – number of liters of vegetable soup the restaurant produces.

Problem formulation. We begin by calculating the profit from the sale of each of the two soups. For a liter of chicken soup we will receive \$23 and pay (for the raw materials) $\$(3 \cdot 5)$ for the chicken and $\$(1 \cdot 1)$ for the vegetables, so the total profit is $\$(23 - (3 \cdot 5 + 1 \cdot 1) = 7)$ per liter. Similarly, the profit for a kilogram of vegetable soup is $\$(27 - (2 \cdot 5 + 6 \cdot 1) = 11)$. Since the restaurant plans to sell x_1 liters of chicken soup and x_2 liters of vegetable soup, its profit will be $7x_1 + 11x_2$, and this is the objective function in the maximization problem.

When the restaurant produces x_1 liters of chicken soup, it uses $3x_1$ kilograms of chicken; when the restaurant produces x_2 liters of vegetable soup, it requires $2x_2$ kilograms of chicken, so in total, it needs $3x_1 + 2x_2$ kilograms of chicken. Since it can buy at most 12 kilograms of chicken, the constraint we obtain is $3x_1 + 2x_2 \leq 12$.

Similarly, the constraint that the restaurant can buy at most 12 kilograms of vegetables can be formulated as $x_1 + 6x_2 \leq 12$. The quantities produced from each soup are of course nonnegative, and therefore, $x_1, x_2 \geq 0$. The problem formulation is, therefore,

$$\begin{array}{ll} \max & 7x_1 + 11x_2 \\ \text{s.t.} & 3x_1 + 2x_2 \leq 12, \\ & x_1 + 6x_2 \leq 12, \\ & x_1, x_2 \geq 0. \end{array} \tag{1.5}$$

Later, in Section 1.3.1, we will see how to solve this particular problem.

In the general production problem, we are given m raw materials from which n products can be produced. The data of the problem describes the production technology (that is, how many raw materials are required to produce one unit of each product), the prices of the raw materials, the selling prices of the products, and the quantities of the given raw materials. The factory's problem is: how much to produce of each of the products so that the total profit from their sale will be maximal?

The Production Problem (General Case). A certain factory uses m raw materials to produce n products. The following parameters are given:

- a_{ij} – amount of raw material i needed to produce one unit of product j ($i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$).
- p_j – selling price of product j ($j = 1, 2, \ldots, n$).
- q_i – purchase price of a unit of raw material i ($i = 1, 2, \ldots, m$).
- b_i – maximum amount of raw material i ($i = 1, 2, \ldots, m$).

How many units should be produced from each product, under the constraints of the limited amount of the raw materials, so that the profit obtained from the sale of the products will be maximal?

First, denote the matrix $(a_{ij})_{i,j}$ by $\mathbf{A} \in \mathbb{R}^{m \times n}$. In addition, denote

$$\mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix}, \mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

Decision variables.
x_j – number of units of product j the factory produces ($j = 1, 2, \ldots, n$).

The decision variables (column) vector is denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. Before we continue formulating the problem, we will perform the following auxiliary calculation: the net profit from selling one unit of product j is $c_j = p_j - \sum_{i=1}^m q_i a_{i,j}$, and in vector notation, $\mathbf{c} = \mathbf{p} - \mathbf{A}^T \mathbf{q}$.

Problem formulation.

$$\begin{array}{lll} \max & \sum_{j=1}^n c_j x_j & \\ \text{s.t.} & \sum_{j=1}^n a_{ij} x_j \leq b_i, & i = 1, 2, \ldots, m, \\ & x_j \geq 0, & j = 1, 2, \ldots, n. \end{array}$$

In matrix notation,

$$\begin{array}{ll} \max & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$

Remark 1.9. *Production problems are sometimes called "resource allocation problems".*

Remark 1.10. *In some production problems, the products in question are not divisible, and in these cases, it is natural to add integrality constraints on the variables. Problems with integrality constraints are discussed in Section 1.4.*

1.2.3 ▪ The Transportation Problem

We begin with a special case of the transportation problem, which will be followed by the formulation of the general problem.

The Transportation Problem (Special Case). Resources located in 2 warehouses must be distributed to 3 stores. Each warehouse has a certain supply of the resources (inventory), and each store has a certain demand for the resources. The transportation costs from each warehouse to each store are also given.
The goal is to distribute the resources from the warehouses to the stores with minimal transportation costs, while meeting the demand and supply requirements.
The following parameters are known:

- c_{ij} – unit transportation cost from warehouse i to store j ($i = 1, 2,\ j = 1, 2, 3$).
- s_i – supply of warehouse i ($i = 1, 2$).
- d_j – demand of store j ($j = 1, 2, 3$).

We will assume that the total supply is equal to the total demand, meaning that $s_1 + s_2 = d_1 + d_2 + d_3$ ("balanced transportation problem").

Decision variables.
x_{ij} – number of units of resources transported from warehouse i to store j ($i = 1, 2,\ j = 1, 2, 3$).
Problem formulation.

$$\begin{array}{ll} \min & c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23} \\ \text{s.t.} & x_{11} + x_{12} + x_{13} = s_1, \\ & x_{21} + x_{22} + x_{23} = s_2, \\ & x_{11} + x_{21} = d_1, \\ & x_{12} + x_{22} = d_2, \\ & x_{13} + x_{23} = d_3, \\ & x_{ij} \geq 0, i = 1, 2, j = 1, 2, 3. \end{array} \tag{1.6}$$

The first two constraints are the *supply constraints* that express the requirement that all the resources in the warehouses are being transported. The next three constraints are the *demand constraints* that express the requirement that the stores receive the demanded resources.

To write the problem in matrix notation, define the vectors

$$\mathbf{c} = \begin{pmatrix} c_{11} \\ c_{12} \\ c_{13} \\ c_{21} \\ c_{22} \\ c_{23} \end{pmatrix} \in \mathbb{R}^6, \mathbf{b} = \begin{pmatrix} s_1 \\ s_2 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} \in \mathbb{R}^5, \mathbf{x} = \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{pmatrix} \in \mathbb{R}^6$$

and the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{5 \times 6}. \tag{1.7}$$

With the above notation, the LP problem (1.6) can be rewritten as

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}. \end{array}$$

Remark 1.11. *The supply and demand constraints can also be written as inequalities (smaller-than-or-equal-to in the supply constraints and larger-than-or-equal-to in the demand constraints), and in any case they will be satisfied as equalities since the total supply here is equal to the total demand.*

Remark 1.12 (unbalanced transportation problems). *The assumption that total supply equals total demand is not a limiting assumption in the following sense. First, it is clear that we must assume that the total demand is less-than-or-equal-to the total supply, that is, that* $d_1+d_2+d_3 \leq s_1+s_2$*, since otherwise it will not be possible to meet the demands of the stores, meaning that the problem is not feasible. Suppose then that* $d_1+d_2+d_3 < s_1+s_2$*. In this case, an additional store can be defined with the demand equal to the difference between the total supply and the total demand,* $s_1+s_2-d_1-d_2-d_3$*, and with transportation costs 0 from each of the warehouses. Any resource that the optimal solution of the new problem supplies to this store actually remains in the warehouse. We thus obtain a reduction to a balanced transportation problem for which the assumption "total demand equals total supply" holds true.*

Remark 1.13 (integrality constraints). *We have initially defined the problem without any integrality constraints on the variables. However, if the resources are indivisible, we must introduce integrality constraints on the variables. When the decision variables in a linear programming problem have integrality constraints, then the problem is referred to as an* **integer programming problem**. *Although it is simple to add integrality constraints on the decision variables, the critical question is whether these constraints make the problem more difficult to solve. In general, integer programming problems are more challenging than (continuous) linear programming problems, as we will explore later. Nonetheless, the transportation problem is a unique type of an integer programming problem that can be solved as easily as linear programming problems. Further details on this can be found in Chapter 8.*

The matrix $\mathbf{A}$ corresponding to the transportation problem given in (1.7) can be written in abbreviated notation as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{1}_3 \\ \mathbf{I}_3 & \mathbf{I}_3 \end{pmatrix},$$

where $\mathbf{1}_3$ and $\mathbf{0}_3$ are the three-dimensional row vectors of all ones and all zeros, respectively, and $\mathbf{I}_3$ is the identity matrix of order 3.

Clearly, this matrix has a very special structure. A generalization of this structure naturally exists in the general transportation problem presented below.

The Transportation Problem (General Case). Resources located in m warehouses must be distributed to n stores. Each warehouse has a certain supply of the resources (inventory), and each store has a certain demand for the resources. The transportation costs from each warehouse to each store are also given.
The goal is to distribute the resources from the warehouses to the stores with minimum transportation costs, while meeting the demand and supply requirements.
The following parameters are known:

- c_{ij} – unit transportation cost from warehouse i to store j ($i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$).
- s_i – supply of warehouse i ($i = 1, 2, \ldots, m$).
- d_j – demand of store j ($j = 1, 2, \ldots, n$).

We will assume that the total supply is equal to the total demand, meaning that $\sum_{i=1}^m s_i = \sum_{i=1}^n d_i$.

Decision variables.
x_{ij} – number of units of resources transported from warehouse i to store j ($i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$).

Problem formulation. Denote by $\mathbf{C} \in \mathbb{R}^{m\times n}$ the matrix of transportation costs, meaning that $\mathbf{C} = (c_{ij})_{ij}$. Similarly, let $\mathbf{X} \in \mathbb{R}^{m\times n}$ be the matrix of decision variables $\mathbf{X} = (x_{ij})_{ij}$.

The mathematical formulation of the transportation problem is

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{array} \tag{1.8}$$

where $\mathbf{c} \in \mathbb{R}^{mn}, \mathbf{b} \in \mathbb{R}^{m+n}, \mathbf{A} \in \mathbb{R}^{(m+n)\times mn}$, and the decision variables vector $\mathbf{x} \in \mathbb{R}^{mn}$ are defined as

$$\mathbf{x} = \text{vec}(\mathbf{X}), \mathbf{c} = \text{vec}(\mathbf{C}), \mathbf{b} = \begin{pmatrix} s_1 \\ \vdots \\ s_m \\ d_1 \\ \vdots \\ d_n \end{pmatrix}, \mathbf{A} = \begin{pmatrix} \mathbf{1}_n & \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{1}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{0}_n & \mathbf{1}_n & \cdots & \mathbf{0}_n & \mathbf{0}_n \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \mathbf{0}_n & \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{1}_n & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{0}_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & \mathbf{1}_n \\ \mathbf{I}_n & \mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{I}_n & \mathbf{I}_n \end{pmatrix},$$

where for any matrix $\mathbf{M}$, the vector $\text{vec}(\mathbf{M})$ represents the column vector formed by concatenating the columns of $\mathbf{M}$ consecutively.

Remark 1.14. *Later, in Section 8.1 (see Remark 8.14), we will show that the matrix* $\mathbf{A}$ *defined above has a special property called "total unimodularity". As we will see in Chapter 8, this property of the coefficient matrix entails the following property of the LP problem (1.8): if the problem has an optimal solution, and the right-hand side vector* $\mathbf{b}$ *is integer, then it necessarily has an integer optimal solution. Therefore, in particular, if the transportation problem has a unique optimal solution, and the supplies and demands are all integers, then its unique optimal solution is necessarily integer-valued.*

1.2.4 ▪ Line fitting

Given n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in the plane, the objective is to find a line of the form $y = ax + b$ that "best fits" these points. For example, Figure 1.1 describes 5 points marked as $(x_i, y_i), i = 1, 2, 3, 4, 5$, and a line that approximates these points "well".

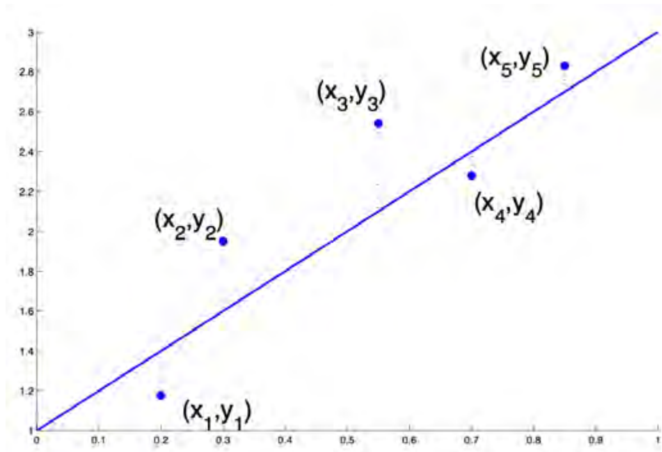

Figure 1.1: Five points $(x_i, y_i), i = 1, 2, \ldots, 5$, and a line that best fits the points.

The problem, as presented verbally above, is not fully defined. We have yet to precisely define the criterion of proximity between the points and the line. We will show that different criteria for closeness yield different optimization problems.

All the objective functions we will look at refer to the *vertical distances* between the line and the points, meaning the quantities $|ax_i + b - y_i|, i = 1, 2, 3, 4, 5$, which are exactly the lengths of the dashed lines in Figure 1.1. The ith vertical line is actually the length of the line segment between (x_i, y_i) and $(x_i, ax_i + b)$. Note that in this problem the decision variables are a and b, while the constants are $x_i, y_i, i = 1, 2, 3, 4, 5$.

First objective function: sum of squares of the vertical distances between the points and the line.

In this case, the optimization problem is

$$\min_{a,b} \sum_{i=1}^{n} (ax_i + b - y_i)^2.$$

This is an unconstrained problem since a priori a and b can take any value. However, the objective function is nonlinear,[1] and therefore this is not a linear programming problem. Hence, we will not address this particular problem.

[1]The fact that the resulting problem is nonlinear does not mean that it is difficult. In this case, for example, it is relatively simple to find an explicit expression for the optimal (a, b). The line that is derived from solving this problem is commonly referred to as the *least squares line*.

Second objective function: sum of the vertical distances between the points and the line.

In this case, the optimization problem is

$$\min_{a,b} \sum_{i=1}^{n} |ax_i + b - y_i|. \tag{1.9}$$

The above problem does not fall under the category of linear programming since it involves absolute value expressions, which are not allowed in such problems. However, it is possible to convert this problem into a linear programming one. To do so, we can substitute $d_i = |ax_i + b - y_i|$, where $d_1, d_2, \ldots, d_n$ are new decision variables. This substitution transforms (1.9) into the problem

$$(P_1) \quad \begin{array}{ll} \min & \sum_{i=1}^{n} d_i \\ \text{s.t.} & |ax_i + b - y_i| = d_i, \; i = 1, 2, \ldots, n, \end{array}$$

in the decision variables $a, b, d_1, d_2, \ldots, d_n$. Now, exchanging the equality constraints by inequality constraints leads to the problem

$$(P_2) \quad \begin{array}{ll} \min & \sum_{i=1}^{n} d_i \\ \text{s.t.} & |ax_i + b - y_i| \le d_i, \; i = 1, 2, \ldots, n. \end{array}$$

Problems (P_1) and (P_2) are equivalent in the sense that a vector $(a, b, d_1, d_2, \ldots, d_n)$ is an optimal solution of (P_1) if and only if it is an optimal solution of problem (P_2). The reason for this is that any optimal solution of (P_2) necessarily satisfies the constraints as equalities. To see this, assume in contradiction that there is an optimal solution of (P_2), $(a^*, b^*, d_1^*, d_2^*, \ldots, d_n^*)$, that does not satisfy the inequalities as equalities. That is, there exists a constraint that is strictly satisfied. Assume without loss of generality that it is the first constraint, meaning that $|a^*x_1 + b^* - y_1| < d_1^*$. In this case, the vector $(a, b, d_1, d_2, \ldots, d_n) = (a^*, b^*, |a^*x_1 + b^* - y_1|, d_2^*, \ldots, d_n^*)$ constructed by reducing d_1^* to $|a^*x_1 + b^* - y_1|$ is a feasible solution. For this feasible solution of problem (P_2), the first constraint holds as an equality and the objective function is smaller, which is a contradiction to the assumed optimality of $(a^*, b^*, d_1^*, d_2^*, \ldots, d_n^*)$.

To rewrite problem (P_2) as an LP problem, we will use the simple fact that the inequality $|x| \le \alpha$ is the same as the relation $-\alpha \le x \le \alpha$; that is, the relation $|x| \le \alpha$ is equivalent to the two inequalities $x \le \alpha, -x \le \alpha$. We can thus conclude that problem (P_2), and hence also the original problem (1.9), is equivalent to the following LP problem in the decision variables $a, b, d_1, d_2, \ldots, d_n$:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} d_i \\ \text{s.t.} & ax_i + b - y_i \le d_i, \; i = 1, 2, \ldots, n, \\ & -(ax_i + b - y_i) \le d_i, \; i = 1, 2, \ldots, n. \end{array}$$

Remark 1.15. *There is no need to impose nonnegativity constraints on the d-variables since any feasible solution must satisfy them anyway.*

Third objective function: maximum of the vertical distances between the points and the line.

The optimization problem in this case is

$$\min_{a,b} \max_{i=1,\ldots,n} |ax_i + b - y_i|. \tag{1.10}$$

As with the previous objective function, this problem, as it is formulated, is not a linear programming problem, as linear programming problems cannot contain max expressions in their objective function (or constraints). To transform the problem into an LP problem, we introduce a new decision variable z, which will represent the objective function, and rewrite problem (1.10) as

$$\begin{array}{ll} \min & z \\ \text{s.t.} & \max\limits_{i=1,\dots,n} |ax_i + b - y_i| = z. \end{array}$$

From similar considerations to those applied for the development of the LP formulation of the previous objective function, we can conclude that it is possible to replace the equality constraints with inequality constraints and obtain the equivalent problem

$$\begin{array}{ll} \min & z \\ \text{s.t.} & \max\limits_{i=1,\dots,n} |ax_i + b - y_i| \le z. \end{array} \tag{1.11}$$

For any $n+1$ numbers $\alpha_1, \alpha_2, \dots, \alpha_n, \beta$ it holds that $\max_{i=1,\dots,n} \alpha_i \le \beta$ if and only if $\alpha_i \le \beta$ for any $i = 1, 2, \dots, n$. Using this property leads us to conclude that problem (1.11) is the same as

$$\begin{array}{ll} \min & z \\ \text{s.t.} & |ax_i + b - y_i| \le z, \ i = 1, 2, \dots, n. \end{array}$$

The inequality $|ax_i + b - y_i| \le z$ can be rewritten as $-z \le ax_i + b - y_i \le z$; that is, the relation $|ax_i + b - y_i| \le z$ is equivalent to the following two inequalities:

$$\begin{aligned} ax_i + b - y_i &\le z, \\ -ax_i - b + y_i &\le z. \end{aligned}$$

In conclusion, the optimization problem (1.10) is equivalent to the following LP problem in the decision variables a, b, z:

$$\begin{array}{ll} \min & z \\ \text{s.t.} & ax_i + b - y_i \le z, \ i = 1, 2, \dots, n, \\ & -ax_i - b + y_i \le z, \ i = 1, 2, \dots, n. \end{array}$$

1.3 ▪ Graphical Solutions of Two-Dimensional LP Problems

This section introduces a solution technique aimed at solving two-dimensional LP problems through their graphical representation. We begin with an example.

1.3.1 ▪ Example: Solving the Production Problem

We will see how the production problem (1.5) formulated in Section 1.2.2 can be solved. Recall that the formulation of the problem is

$$\begin{array}{ll} \max & 7x_1 + 11x_2 \\ \text{s.t.} & 3x_1 + 2x_2 \le 12, \\ & x_1 + 6x_2 \le 12, \\ & x_1, x_2 \ge 0. \end{array} \tag{1.12}$$

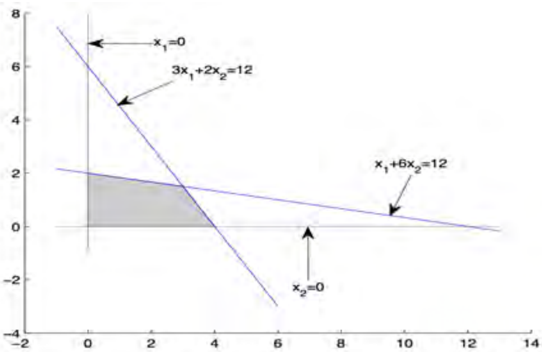

Figure 1.2: Feasible set of problem (1.12).

The constraints of the problem and the feasible set are plotted in Figure 1.2. Each constraint is represented by a line. This line which we refer to as the "constraint line" is the line formed by writing the constraint as an equality. The feasible region is the area that satisfies all of the constraints and is a polygon with vertices at the points where the constraint lines intersect.
The nonnegativity constraints limit us to the first quadrant. The set of feasible solutions is the shaded polygon whose vertices are

$$(0, 2), (3, 1.5), (4, 0), (0, 0).$$

Is it possible to get a profit of \$14? Of course yes, because the point (2,0) that belongs to the feasible set (that is, satisfies all the constraints) gives this profit. Furthermore, the line $7x_1 + 11x_2 = 14$ has an infinite number of intersection points with the feasible set, as can be seen in Figure 1.3.

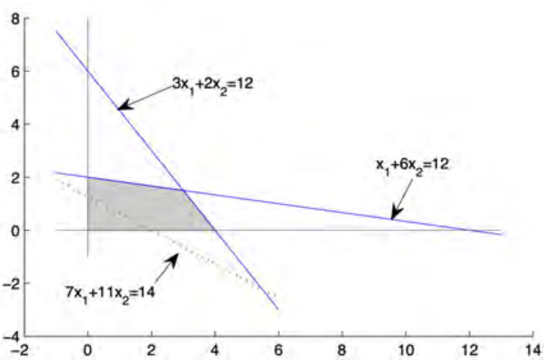

Figure 1.3: The intersection of the line $7x_1 + 11x_2 = 14$ with the feasible set of problem (1.12) is a line segment with an infinite number of points.

The question we want to answer is, actually, What is the largest c for which the equation $7x_1 + 11x_2 = c$ still has an intersection with the feasible set, or in other words, how far can the line be moved, in a parallel way, in the appropriate direction which increases c, so we will still intersect the feasible set? In Figure 1.4, it can seen that the last intersection point (when moving the line) is at the vertex $(3, 1.5)$, and therefore, the point $x_1 = 3, x_2 = 1.5$ that gives an objective function value of 37.5 is the optimal solution (in this case, the unique optimal solution).

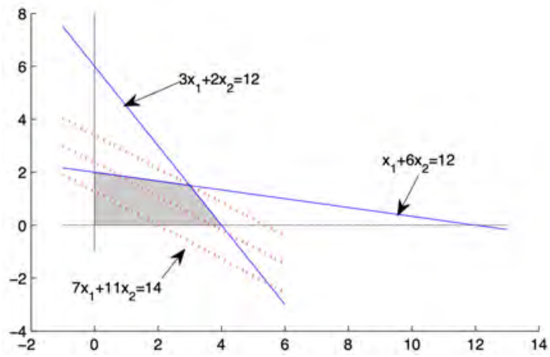

Figure 1.4: The optimal solution is the vertex $(3, 1.5)$, which is the last intersection of the objective function line with the feasible set when moving the line in the direction that increases its value.

In general, any line always “leaves” the feasible set through at least one of its vertices, indicating that the optimal solution should be sought among these vertices. Later, in Section 2.5, this observation will be formally stated and proved.

If we were to change the objective function to $6x_1 + 4x_2$, how would the optimal solution change? Note that in this scenario, the slope of the line representing the objective function is identical to that of the first constraint, meaning that the lines are parallel. Consequently, the optimal set encompasses all the points on the line segment stretching between $(3, 1.5)$ and $(4, 0)$ and the optimal value is $6 \cdot 3 + 4 \cdot 1.5 = 6 \cdot 4 + 4 \cdot 0 = 24$. The reason behind this is that if we move the line representing the objective function in a parallel way in the direction that increases its value, the line will leave the set of feasible solutions through the constraint line $3x_1 + 2x_2 = 12$; see the illustration in Figure 1.5.

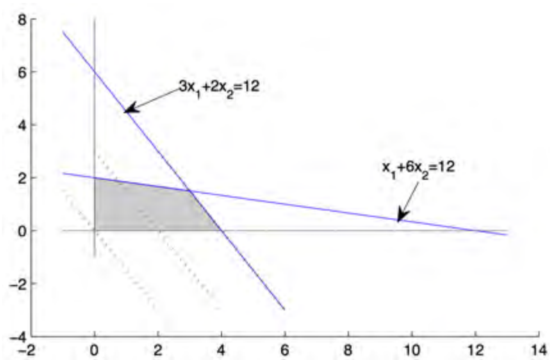

Figure 1.5: The optimal solution is the line segment between $(3, 1.5)$ and $(4, 0)$.

To summarize, the vertices $(3, 1.5)$ and $(4, 0)$ and the line segment between them comprise the optimal set of the problem, and they all correspond to an optimal value of 24. A formal definition of a line segment follows.

Definition 1.16 (line segment). *Suppose that* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. *The line segment between* $\mathbf{x}$ *and* $\mathbf{y}$ *is the set*

$$[\mathbf{x}, \mathbf{y}] = \{\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} : 0 \leq \lambda \leq 1\}.$$

For instance, in the last example, the set of optimal solutions, which is the line segment between $(3, 1.5)^T$ and $(4, 0)^T$, can be analytically expressed as

$$S = \left\{ \lambda \begin{pmatrix} 3 \\ 1.5 \end{pmatrix} + (1-\lambda) \begin{pmatrix} 4 \\ 0 \end{pmatrix} : 0 \le \lambda \le 1 \right\} = \left\{ \begin{pmatrix} 4-\lambda \\ 1.5\lambda \end{pmatrix} : 0 \le \lambda \le 1 \right\}.$$

We will now perform a sanity check test and see that all the points in S satisfy the constraints of the problem and that their objective function value is 24. First,

$$\begin{aligned} 3x_1 + 2x_2 &= 3(4-\lambda) + 2(1.5\lambda) = 12 - 3\lambda + 3\lambda = 12, \\ x_1 + 6x_2 &= 1(4-\lambda) + 6(1.5\lambda) = 4 - \lambda + 9\lambda = 4 + 8\lambda \le 12, \end{aligned}$$

where the last inequality follows by the fact that $\lambda \le 1$. Therefore, we have determined that the two linear inequalities are satisfied. In addition, since $0 \le \lambda \le 1$, we obviously have that $4 - \lambda \ge 0$ and $1.5\lambda \ge 0$, and consequently the nonnegativity constraints are also met. The objective function value at the points in S is

$$6x_1 + 4x_2 = 6(4-\lambda) + 4(1.5\lambda) = 24,$$

as expected.

In general, based on graphical intuition, we can draw the following conclusions regarding LP problems with two variables:

- The feasible set of an LP problem, if not empty, is a polygon with a finite number of vertices.
- If an LP problem has an optimal solution, then it has an optimal solution that is a vertex. It thus follows that to find the optimal solution of the problem, it is enough to compute the objective function value at the vertices of the polygon and pick the one with the best value (under the assumption that an optimal solution does exist).
- If two vertices are optimal solutions of an LP problem, then all the points on the line segment between these two points are optimal solutions of the problem.

The above observations can also be formally proved.

1.3.2 ▪ Infeasible Problems

Consider the problem

$$\begin{array}{ll} \min & 3x_1 + x_2 \\ \text{s.t.} & x_1 + x_2 \le 2, \\ & 2x_1 + 4x_2 \ge 12, \\ & x_1, x_2 \ge 0. \end{array}$$

In Figure 1.6 the constraint lines are drawn.

Clearly, since any feasible point needs to be below the line $x_1 + x_2 = 2$ and above the line $2x_1 + 4x_2 = 12$ as well as in the first quadrant, no such point exists, and thus the feasible set of the problem is the empty set, and in particular the problem has no optimal solutions. Optimization problems with empty feasible sets are called *infeasible problems*.

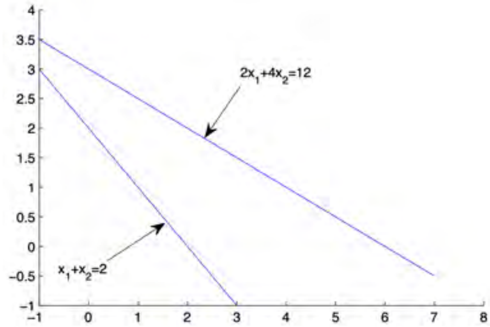

Figure 1.6: There are no solutions to the system $x_1 + x_2 \leq 2, 2x_1 + 4x_2 \geq 12, x_1, x_2 \geq 0$.

1.3.3 ▪ LP Problems with Unbounded Feasible Sets

Consider the problem

$$\begin{array}{lrl} \max & x_1 + & x_2 \\ \text{s.t.} & -x_1 + & x_2 \leq 2, \\ & 0.5x_1 - & 0.5x_2 \leq 1, \\ & & x_1, x_2 \geq 0. \end{array} \tag{1.13}$$

The constraint lines are drawn in Figure 1.7. Note that in this example the two constraint lines are parallel and the feasible set is unbounded. When drawing lines of the form $x_1 + x_2 = c$, we get that we can increase c as we wish and still stay within the feasible set; see Figure 1.8. We conclude that the feasible set is unbounded and the maximal value is ∞.

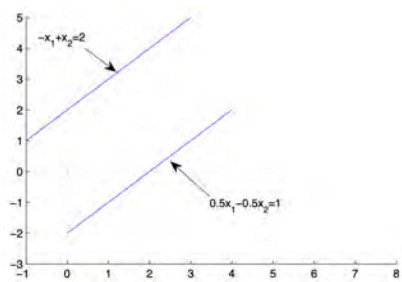

Figure 1.7: The two constraint lines of problem (1.13) are parallel.

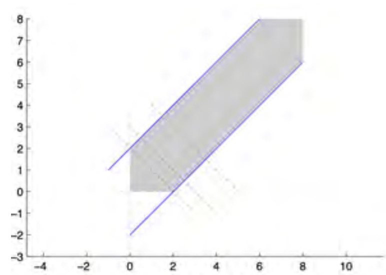

Figure 1.8: The line $x_1 + x_2 = c$ has a nonempty intersection with the feasible set of problem (1.13) for any $c \geq 0$.

Definition 1.17 (unbounded and bounded problems). *An optimization problem is called* **unbounded** *if its feasible set is unbounded and its optimal value is ∞ in case of a maximization problem and $-\infty$ in case of a minimization problem. A feasible optimization problem with a finite optimal value is called* **bounded**.

Now let us consider the case where the objective function changes to $-2x_1 + x_2$. Figure 1.9 describes the lines $-2x_1 + x_2 = c$ for different values of c along with the feasible set of the problem. In this case, the problem has a finite optimal value since the line $-2x_1 + x_2 = c$ intersects the feasible set as long as $c \leq 2$, meaning in particular that the maximal value is 2. This is an example of a bounded problem with an unbounded feasible set.

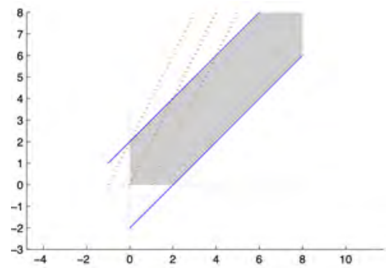

Figure 1.9: The line $-2x_1 + x_2 = c$ has a nonempty intersection with the feasible set for any $c \leq 2$.

In general, an LP problem may have an unbounded feasible set, and then there are two cases:

(a) The optimal value of the problem is infinite, and then the problem is *unbounded*.

(b) The optimal value of the problem is finite, and then the problem is *bounded*.

1.3.4 ▪ Example: Solution of the Diet Problem

We now demonstrate the graphical procedure by solving the diet problem (1.3) described in Section 1.2.1:

$$\begin{array}{ll} \min & 0.4x_1 + 0.6x_2 \\ \text{s.t.} & x_1 + x_2 \geq 3, \\ & 6x_1 + 2x_2 \geq 6, \\ & x_1 + 5x_2 \geq 5, \\ & x_1, x_2 \geq 0. \end{array}$$

The constraint lines and the feasible set are described in Figure 1.10. Note that the constraint $6x_1 + 2x_2 \geq 6$ is dominated by the constraint $x_1 + x_2 \geq 3$ in the first quadrant, meaning that any point in the first quadrant satisfying $x_1 + x_2 \geq 3$ also satisfies the inequality $6x_1 + 2x_2 \geq 6$. We can therefore ignore the constraint $6x_1 + 2x_2 \geq 6$. Drawing the objective function, we can see that it leaves the feasible set at the point $(2.5, 0.5)$; see Figure 1.11. Hence, the (unique) optimal solution of the problem is $(2.5, 0.5)$ with a corresponding optimal objective function value of 1.3.

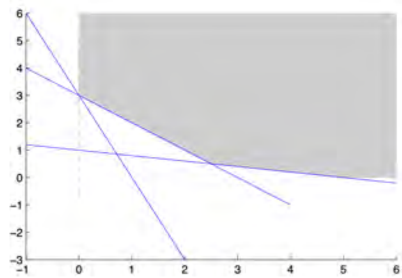

Figure 1.10: The feasible set of the diet problem.

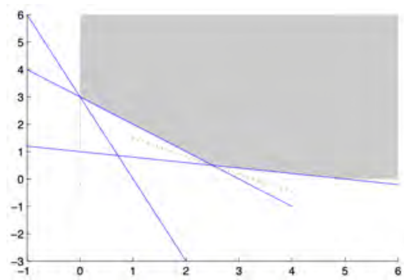

Figure 1.11: The objective function line leaves the feasible set at the optimal point $(2.5, 0.5)$.

1.4 ▪ Integer Programming Problems

An integer programming (IP) problem is a linear programming problem in which all variables are constrained to be integers. For certain problems with integer variables, specialized algorithms can handle these constraints, while for others, the constraints can be ignored, and an integer solution can still be obtained by solving the problem as a continuous LP problem. Unfortunately, most integer programming problems are considered as hard problems that cannot be solved in a reasonable running time. This section will describe formulations of some important integer programming problems. In upcoming chapters, we will present techniques for solving integer programming problems. It is worth mentioning that if only a subset of variables are required to be integers, the problem is referred to as a *mixed integer programming* problem, which is equally as challenging to solve as IP problems.

An important class of IP problems are *binary programming* problems in which the variables can only attain the values 0 or 1, meaning that these problems incorporate the constraints $x_i \in \{0, 1\}$ for all the variables x_i in the formulation. Since the binary constraint is equivalent to

$$\begin{aligned} &x_i \ \text{ integer}, \\ &0 \le x_i \le 1, \end{aligned}$$

it follows that binary programming problems are a subclass of the class of IP problems.

1.4.1 ▪ The Knapsack Problem

The knapsack problem presented below is essentially the simplest non-trivial example of an integer programming problem.

The Knapsack Problem. We are given a set of n items, labeled as $1, 2, \ldots, n$, where each item j has a corresponding value v_j and weight w_j. The objective is to choose a collection of items and pack them into a knapsack in such a way as to maximize the total value of the chosen collection while ensuring that the total weight does not exceed a predefined capacity C.

Decision variables.

x_j – a decision variable that equals 1 if item j is packed in the knapsack and 0 if it is not packed ($j = 1, 2, \ldots, n$).

Problem formulation.

$$\begin{array}{ll} \max & \sum_{j=1}^{n} v_j x_j \\ \text{s.t.} & \sum_{j=1}^{n} w_j x_j \le C, \\ & x_j \in \{0,1\}, j = 1, 2, \ldots, n. \end{array}$$

1.4.2 ▪ The Assignment Problem

The Assignment Problem. We are given n workers and n tasks, each labeled as $1, 2, \ldots, n$. The profit from assigning worker i to task j is c_{ij}. Each task should be assigned to exactly one worker, and each worker should be assigned to exactly one task. The objective is to determine the optimal assignment of workers to tasks such that the total profit is maximized.

Decision variables.

x_{ij} – a decision variable that equals 1 if worker i is assigned to task j and 0 otherwise ($i, j = 1, 2, \ldots, n$).

Problem formulation.

$$\begin{array}{lll} \max & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} & \text{[maximization of the total profit]} \\ \text{s.t.} & \sum_{j=1}^{n} x_{ij} = 1, \ i = 1, 2, \ldots, n, & \text{[each worker is assigned to one task]} \\ & \sum_{i=1}^{n} x_{ij} = 1, \ j = 1, 2, \ldots, n, & \text{[each task is assigned one worker]} \\ & x_{ij} \in \{0,1\}, \ i, j = 1, 2, \ldots, n. & \text{[binary constraints]} \end{array}$$

Remark 1.18.

1. *Assignment problems can be cast as either maximization or minimization problems.*

2. *The binary constraints can be replaced by nonnegativity and integrality constraints. Specifically, the constraints*

$$x_{ij} \in \{0,1\}, \ i, j = 1, 2, \ldots, n,$$

are the same as

$$\begin{array}{l} x_{ij} \ge 0, \ i, j = 1, 2, \ldots, n, \\ x_{ij} \ \textit{integer}, \ i, j = 1, 2, \ldots, n. \end{array} \tag{1.14}$$

The reason for the above equivalency is that the equality constraints are of the form "sum of variables equals 1". Since each of the variables appears in at least one equality constraint, it follows that the nonnegativity and integrality constraints imply the binary constraints.

3. *The assignment problem is a special case of the transportation problem described in Section 1.2.3, where the "supplies" and "demands" are 1. Consequently, the assignment problem also has the special feature that was mentioned in the context of the transportation problem (see Remark 1.14): if we remove the integrality constraints from (1.14), then we get a linear (continuous) programming problem that has at least one optimal solution which is integer, i.e., binary.*

1.4.3 ▪ Vertex Cover

The vertex cover problem is a well known problem from graph theory. We begin with the definition of a graph. A graph $G = (V, E)$ is defined by two sets: V, a vertex set, and E, an edge set. The set of edges describes which vertices are connected by an edge. For example, Figure 1.12 describes a graph $G = (V, E)$ with

- vertex set $V = \{1, 2, 3, 4\}$,
- edge set $E = \{\{1, 2\}, \{2, 3\}, \{1, 3\}, \{3, 4\}\}$.

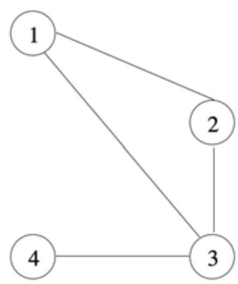

Figure 1.12: An example of a graph with four vertices.

We use curly brackets to denote the edges to emphasize the fact that the edges have no direction. Such a graph is called an *undirected graph*. If the edges have a direction, the graph is a *directed graph*, and the edges are sometimes referred to as *arcs*. For example, the graph described in Figure 1.13 is a directed graph. The vertex and edge sets are given by

- $V = \{1, 2, 3, 4\}$,
- $E = \{(1, 2), (2, 3), (3, 1), (4, 3)\}$.

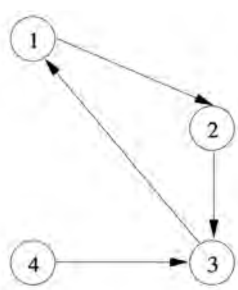

Figure 1.13: An example of a directed graph with four vertices.

Here we used the ordered pairs notation to describe the arcs since the order of the vertices matters. We are now ready to define the vertex cover problem.

The Vertex Cover Problem. Given an undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, find a smallest possible subset of vertices such that every edge in the graph is incident to at least one vertex in the subset.

An edge is said to be "covered" if at least one of its vertices is chosen.

Decision variables.

x_i – a decision variable that equals 1 if vertex i is chosen and 0 otherwise ($i = 1, 2, \ldots, n$).

Problem formulation.

min	$\sum_{i=1}^n x_i$	[minimum number of vertices in the chosen subset]
s.t.	$x_i + x_j \geq 1, \{i, j\} \in E$	[each edge is covered by at least one vertex]
	$x_i \in \{0, 1\},\ i = 1, 2, \ldots, n.$	[binary constraints]

Note that the vertex cover problem is considered a difficult problem. There is no known algorithm for its solution whose running time is polynomial in the size of the input.

Remark 1.19. *In the above formulation, the binary constraints*

$$x_i \in \{0, 1\},\ i = 1, 2, \ldots, n, \tag{1.15}$$

can be replaced by the nonnegativity and integrality constraints

$$x_i \geq 0 \text{ integer},\ i = 1, 2, \ldots, n. \tag{1.16}$$

The explanation for this is that, when solving the problem with the integrality constraints (1.16) instead of the binary constraints (1.15), an optimal solution will always satisfy the condition that the decision variables are less-than-or-equal-to 1. To see why, suppose that this condition does not hold for an optimal solution $\mathbf{x}$*, and consider a variable, say* x_1*, that has a value greater than 1 in the optimal solution. Then, we can construct a feasible solution with a smaller objective function value by reducing* x_1 *to 1 and leaving all other variables unchanged. This contradicts the optimality of* $\mathbf{x}$*.*

1.4.4 ▪ Graph Coloring

The graph coloring problem, which traces its roots back to the task of coloring maps, has intrigued mathematicians for many generations. Suppose that a map is given, which is actually a division of part of the plane into separate areas. When coloring a map, the crucial rule is to avoid coloring neighboring countries with the same color. A coloring that adheres to this rule is referred to as a "legal coloring". The objective of the coloring problem for maps is to determine the minimum number of colors needed to achieve a legal coloring. For example, in the left drawing of Figure 1.14, we can see a map of (imaginary) five countries. It is easy to see that the minimum number of colors that must be used is 3. An example of a legal coloring in three colors is given in the right drawing of Figure 1.14 (here, different shades of gray represent different colors).

The problem of coloring maps can be described as a problem in undirected graphs. The graph corresponding to the map is the graph in which each vertex corresponds to a country and an (undirected) edge exists between two vertices if the corresponding countries are neighbors. For example, the graph in Figure 1.15 is the graph corresponding to the colored graph of Figure 1.14.

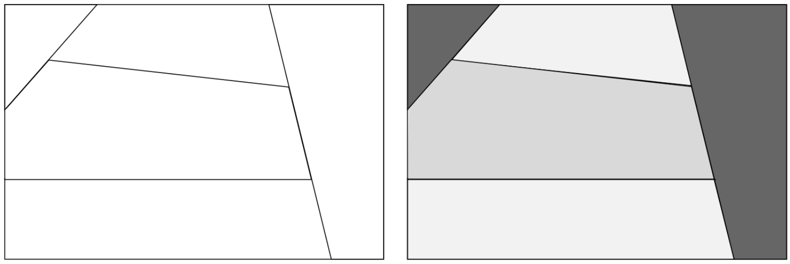

Figure 1.14: A map (left image) and its coloring (right image).

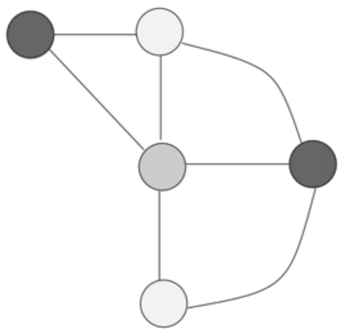

Figure 1.15: A graph representation of the map in Figure 1.14.

Note that we have colored each vertex according to the color of the corresponding area on the map. Consequently, a vertex coloring in an undirected graph is considered "valid" when no two adjacent vertices have the same color. Therefore, the problem of vertex coloring in undirected graphs can be defined as follows.

The Graph Coloring Problem. Given an undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, find the minimum number of colors needed to color the vertices in such a way that no two adjacent vertices share the same color.

We will assume that the colors that are being (potentially) used are $1, 2, \ldots, n$ (obviously no more than n colors are needed).

Decision variables.

y_j – a decision variable that equals 1 if color j is used and 0 otherwise ($j = 1, 2, \ldots, n$).
x_{ij} – a decision variable that equals 1 if vertex i is colored in color j ($i, j = 1, 2, \ldots, n$).

Problem formulation.

$$\begin{array}{lll}
\min & \sum_{j=1}^{n} y_j & \text{[minimum number of colors used]} \\
\text{s.t.} & x_{ij} \le y_j, \ i, j = 1, \ldots, n, & \text{[if vertex } i \text{ is assigned to color } j, \\
& & \ \text{then color } j \text{ is used.]} \\
& \sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n, & \text{[each vertex is assigned to one color]} \\
& x_{kj} + x_{lj} \le 1, \{k, l\} \in E, j = 1, \ldots, n, & \text{[the vertices of each edge} \\
& & \ \text{are not assigned to the same color]} \\
& y_j, x_{ij} \in \{0, 1\}, \ i, j = 1, \ldots, n. & \text{[binary constraints]}
\end{array}$$

Note that the first set of constraints that links the x and y variables is crucial. Without these constraints, a trivial (and obviously invalid) solution will assign zeros to all the y variables and n different colors to each of the vertices.

A very famous problem related to coloring is the four-color problem. Basically, this is a hypothesis that says that every map can be colored in four colors. The question was asked in the context of map coloring as early as 1850 by Francis Guthrie, who was a student of Augustus De Morgan. Guthrie noticed that he was able to color the counties in England using only four colors:[2]

> A student of mine [Guthrie] asked me today to give him a reason for a fact which I did not know was a fact—and do not yet. He says that if a figure be anyhow divided and the compartments differently colored so that figures with any portion of common boundary line are differently colored—four colors may be wanted, but not more—the following is the case in which four colors are wanted. Query cannot a necessity for five or more be invented...

This conjecture for graphs says that any planar graph can be colored in four colors. A planar graph is a graph that can be drawn in the plane without any intersections between its edges. The reason we concentrate on planar graphs is that every graph that represents a map is a planar graph. Over the years proving the hypothesis became an obsession, leading to the invention of numerous incorrect "proofs". The five-color theorem, which states that any planar graph can be colored using five colors, is relatively straightforward to prove (proven in 1890 and covered in most introductory courses on graph theory). However, it was not until 1976 that the four-color theorem was finally proven, utilizing a computer to test hundreds of cases.

1.4.5 ▪ The Traveling Salesman Problem

The Traveling Salesman Problem. Given a list of cities labeled $1, 2, \dots, n$ and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

A few remarks:

- The set of cities is denoted by $V = \{1, 2, \dots, n\}$.
- We denote the distance between two different cities i and j by $d_{ij} > 0$. We do not assume symmetry of the distances, meaning that it is possible that $d_{ij} \neq d_{ji}$.
- The number of different circular routes is $(n-1)!$, which is of course a very large number for large values of n.
- We note that the traveling salesman problem is considered a difficult problem. There is no known algorithm for its solution whose running time is polynomial in the size of the input.

Decision variables.
x_{ij} – a decision variable that equals 1 when the route passes directly from city i to city j and 0 otherwise $(i, j = 1, 2, \dots, n, i \neq j)$.

[2]The quote can be found in Robin Wilson, *Four Colors Suffice*, Penguin Books, London, 2002.

Problem formulation.

$$
\begin{array}{lll}
\min & \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} d_{ij} x_{ij} & \text{[minimum route distance]} \\
\text{s.t.} & \sum_{i=1, i\neq j}^{n} x_{ij} = 1, \ j = 1, 2, \ldots, n, & \text{[the route enters city } j \text{ once]} \\
& \sum_{j=1, j\neq i}^{n} x_{ij} = 1, \ i = 1, 2, \ldots, n, & \text{[the route leaves city } i \text{ once]} \\
& \sum_{i\in S} \sum_{j\notin S} x_{ij} \geq 1, \ S \subseteq V, S \neq V, \emptyset, & \text{[connectivity constraints]} \\
& x_{ij} \in \{0, 1\}, \ i, j = 1, 2, \ldots, n, i \neq j. & \text{[binary constraints]}
\end{array}
$$

We will refer to circular routes also as "cycles". To better understand the connectivity constraints, assume for a moment that these constraints do not exist. In this case, a solution that consists of a union of disjoint cycles will also be a feasible solution of the problem. An example of such a union of cycles appears in Figure 1.16.

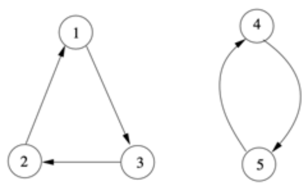

Figure 1.16: Union of disjoint cycles.

In this case, the values of the decision variables are

$$
\begin{aligned}
x_{13} = x_{32} = x_{21} = 1, \\
x_{45} = x_{54} = 1,
\end{aligned}
$$

and all other variables are zeros. These values of the decision variables obviously satisfy the first two sets of constraints in the LP formulation ("the route enters and leaves each city exactly once") which in this case are

$$
\begin{aligned}
x_{21} + x_{31} + x_{41} + x_{51} &= 1, \\
x_{12} + x_{32} + x_{42} + x_{52} &= 1, \\
x_{13} + x_{23} + x_{43} + x_{53} &= 1, \\
x_{14} + x_{24} + x_{34} + x_{54} &= 1, \\
x_{15} + x_{25} + x_{35} + x_{45} &= 1, \\
\\
x_{12} + x_{13} + x_{14} + x_{15} &= 1, \\
x_{21} + x_{23} + x_{24} + x_{25} &= 1, \\
x_{31} + x_{32} + x_{34} + x_{35} &= 1, \\
x_{41} + x_{42} + x_{43} + x_{45} &= 1, \\
x_{51} + x_{52} + x_{53} + x_{54} &= 1.
\end{aligned}
$$

The third set of constraints prevents the situation of disjoint cycles in the following way. For every partition of cities into two sets, S (a set of cities) and its complement set $\bar{S}$, a constraint is introduced. This constraint requires that there must be at least one edge in the route that directly connects a city in S with a city in $\bar{S}$. The solution described in Figure 1.16 fails to satisfy this constraint for the partition defined by the set $S = \{1, 2, 3\}$.

Notice that the number of constraints is of the order of 2^n (number of subsets of V), showing that the formulation itself is not polynomial in the input. Although it is possible to create a formulation that involves a polynomial number of variables and constraints, the inherent difficulty of the problem persists due to the binary constraints.

1.4.6 ▪ The Cutting Stock Problem

The Cutting Stock Problem. Suppose that rolls are produced in a uniform width of 100cm. You received orders for rolls of widths 45cm, 30cm, 20cm, 15cm as follows:

order width	quantity ordered
45	100
30	95
20	275
15	120

A single 100cm roll can be cut into one or more of the order widths. For example, one roll could be cut into three rolls of 30cm and a 10cm roll of scrap. Alternatively, a roll could be cut into five rolls of 20cm with no scrap. Each such possible combination is called a *configuration*. Find how many rolls of each configuration to cut to satisfy the customer orders while minimizing the number of rolls used.

First, a list of all possible configurations must be made, that is, the ways in which the rolls are cut. There is a large number of configurations. We will give here only a partial list:

	widths			
configuration number	15	20	30	45
1	0	0	0	2
2	0	0	3	0
3	0	5	0	0
4	6	0	0	0
5	1	2	0	1
6	2	2	1	0
7	0	2	2	0

The formulation of the problem needs to take into account the long list of all possible configurations. As an illustration, we formulate the problem under the assumption that only the configurations given in the table above can be used.

Decision variables.
x_i – number of rolls cut according to configuration i $(i = 1, 2, \ldots, 7)$.

Problem formulation.

$$
\begin{array}{lllllllll}
\min & x_1+ & x_2+ & x_3+ & x_4+ & x_5+ & x_6+ & x_7 & \\
\text{s.t.} & 2x_1 & & & & +\ x_5 & & & \geq 100, \\
& & 3x_2 & & & & +\ x_6 & +2x_7 & \geq 95, \\
& & & 5x_3 & & +2x_5 & +2x_6 & +2x_7 & \geq 275, \\
& & & & 6x_4+ & x_5 & +2x_6 & & \geq 120, \\
& \multicolumn{8}{l}{x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \text{ integer.}}
\end{array}
$$

One optimal solution of the above problem (obtained by using software for solving IP problems) is the following. A total of 148 rolls are required: 10 from configuration 1, 80 from configuration 5, 21 from configuration 6, and 37 from configuration 7.

It can be shown that this is actually also an optimal solution of the problem even when all the possible configurations are considered.

Exercises

1.1. (•) A café makes pancakes and toasts. Each pancake requires 5 minutes of work by the assistant chef and 3 minutes of work by the chef. Each toast requires 10 minutes of work by the assistant chef and 1 minute of work by the chef. Pancakes sell for \$7 and toasts sells for \$3.

  (a) Write an LP problem to help the café decide how many pancakes to make and how many toasts to make in order to maximize its income, given that it can use 100 minutes of the assistant chef's time and 20 minutes of the chef's time. For simplicity, assume the café has the capability to sell fractional servings of both pancakes and toasts.

  (b) Solve the problem formulated in (a) using the graphical method.

1.2. (•) A bank offers two loan options. The first option (expensive) allows borrowers to borrow at an annual interest rate of 15 percent. In this option, it is possible to borrow up to \$90,000. The second option (cheap) allows borrowers to borrow at an annual interest rate of 10 percent. In this option, it is possible to borrow up to \$50,000. The bank limits the amount of the loan in the second option to no more than three times the amount of the loan in the first option.

  (a) Jonathan needs to borrow \$100,000. Formulate the LP problem that will help Jonathan decide how much money to borrow in each option so that the total annual interest he pays is minimized.

  (b) Solve the problem formulated in (a) using the graphical method.

1.3. (•) Consider the LP problem

$$
\begin{array}{ll}
\min & x_1 + x_2 \\
\text{s.t.} & 3x_1 - 5x_2 \leq -1, \\
& 3x_1 - 2x_2 \geq -4, \\
& 0 \leq x_2 \leq 2.
\end{array}
\tag{1.17}
$$

Solve the problem using the graphical method.

1.4. (∘) Consider the LP problem

$$\begin{array}{ll} \min & 15x_1 + 20x_2 + 1000 \\ \text{s.t.} & 100x_1 + 200x_2 \geq 1000, \\ & 2x_1 - 3x_2 \leq 6, \\ & x_1 + x_2 \geq 6, \\ & x_1, x_2 \geq 0. \end{array}$$

Solve the problem using the graphical method.

1.5. (∘) Consider the LP problem

$$\begin{array}{ll} \max & 2x_1 + x_2 \\ \text{s.t.} & x_2 \leq 10, \\ & 2x_1 + 5x_2 \leq 60, \\ & x_1 + x_2 \leq 18, \\ & 3x_1 + x_2 \leq 44, \\ & x_1, x_2 \geq 0. \end{array}$$

Solve the problem using the graphical method.

1.6. (∘) Consider the LP problem

$$\begin{array}{ll} \max & 2x_1 - x_2 \\ \text{s.t.} & x_1 - x_2 \leq 1, \\ & 2x_1 + x_2 \geq 6, \\ & x_1, x_2 \geq 0. \end{array}$$

(a) Solve the problem using the graphical method.

(b) The constraint $x_1 + x_2 \leq 10$ is added to the problem. What is the optimal solution of the new problem?

1.7. (∘) Grandma is preparing meatballs. In the mixture of the meatballs, she uses chicken and beef.
1 kilogram of chicken costs \$3 and contains 5 calories, 1 gram of protein, 10 mg of sodium, and 3 mg of vitamin B. 1 kilogram of beef costs \$5 and contains 1 calorie, 3 grams of protein, 1 mg of sodium, and 1 mg of vitamin B.
Grandma wants to make meatballs that will provide at least 10 calories, 5 grams of protein, 20 mg of sodium, and 7 mg of vitamin B.
Formulate an LP problem that will help grandma decide how many kilograms of chicken and how many kilograms of beef to buy in order to meet the nutritional requirements at minimum cost. Assume that grandma can prepare a non-integer number of meatballs. Solve the problem using the graphical method.

1.8. (∘) A furniture manufacturer produces two types of writing desks: a regular desk and a manager's desk. Any quantity produced can be marketed and sold. The manufacturing process of a desk includes four stages: cutting the wood, connecting and gluing, sub-finishing, and finishing.

The following table includes the production times (in minutes) for each desk per day, in each of the departments, as well as the total available time per day, in each of the departments:

table type	cutting	connecting	sub-finishing	finishing
regular desk	48	120	40	320
manager's desk	72	180	120	240
total time available to the department	960	1800	960	3840

The profit from a regular desk is \$40 and from a manager's desk is \$50. Formulate this problem as an LP and solve it using the graphical method.
Note: There are no integer constraints on the variables, since parts of desks can be produced for a day of work, and the production can be completed the next day.

1.9. (•) Consider the following optimization problem in the decision variables x, y, z:

$$\begin{array}{ll} \min & |x+y+z| + \max\{x+y, y-z\} \\ \text{s.t.} & |2x+3y| \leq 5, \\ & \max\{x^2, z^2\} \leq 1, \\ & \min\{x, y, z\} \geq -0.5. \end{array}$$

Formulate the problem as an LP problem. Add new decision variables if necessary.

1.10. (∘) Consider the following optimization problem in the decision variables x_i, $i = 1, 2, \ldots, n$:

$$\min \quad \sum_{i=1}^{n} |ix_i + i^2|.$$

Formulate the problem as an LP problem. Add new decision variables if necessary.

1.11. (∘) Let a_i, c_i, $i = 1, 2, \ldots, n$, be nonnegative constants and consider the following optimization problem in the decision variables x_i $i = 1, 2, \ldots, n$:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & x_i x_{i+1} = 0, \ i = 1, 2, \ldots, n-1, \\ & (\sum_{i=1}^{n} a_i x_i^2)^2 - \sum_{i=1}^{n} a_i x_i^2 \geq 2, \\ & x_i = 1 \Rightarrow x_{i+2} = 0, \ i = 1, 2, \ldots, n-2, \\ & x_i \in \{0, 1\}, \ i = 1, \ldots, n. \end{array}$$

Present the problem as a (linear) binary programming problem.

1.12. Consider the following optimization problem in the decision variables x_i, $i = 1, 2, \ldots, n$:

$$\begin{array}{ll} \min & \sqrt{\sum_{i=1}^{n} a_i x_i} \\ \text{s.t.} & x_i x_{i+1} = 0, \ i = 1, 2, \ldots, n-1, \\ & \sum_{i=1}^{n-2} \max\{c_i x_i + x_{i+2}, d_i x_{i+1}\} \leq M, \\ & x_i \in \{0, 1\}, \ i = 1, \ldots, n, \end{array}$$

where a_i $i = 1, 2, \ldots, n$, c_i, d_i $i = 1, 2, \ldots, n-2$, and M are positive constants. Formulate the problem as a mixed integer programming problem.

1.13. (•) A furniture factory wants to determine for each of the next four years the number of chairs to be produced. The demands in the next four years are given in the following table:

year	1	2	3	4
demand	1000	2000	6000	3000

The demands in each year must be met at the end of the year. In each year, no more than 7,000 chairs can be produced. If a chair is produced in a certain year but not supplied at the end of the year, it can be stored at no cost and sold in later years. At the end of four years, no chairs should be left in storage.
The production costs, in dollars, of a single chair in each year are different from each other and are given in the following table:

year	1	2	3	4
costs	0.5	0.8	0.2	0.5

(a) Write the chair production problem with minimal cost, while meeting the demand constraints, as an IP problem (it is not possible to produce part of a chair).

(b) Suppose now that the storage is not free of cost. Specifically, a chair produced in a certain year and not supplied at the end of that year is placed in storage at a cost of \$0.5 per year. A chair placed in storage will remain there for at least one year and can be released at the end of any year thereafter to be supplied. For example, a chair which is produced in the first year and not supplied at the end of this year will be stored for at least a year, and may be supplied at the end of the second year. Update the formulation from (a) to take into account the storage constraints and costs.

1.14. (•) In the warehouse of a packaging factory, there are 8,000 lollipops and 10,000 chewing gums. The factory can produce 3 types of packages as follows:

- Package A: four lollipops and one chewing gum. Market price: \$10.
- Package B: two lollipops and two chewing gums. Market price: \$8.
- Package C: one lollipop and three chewing gums. Market price: \$5.

Assume that there are no production costs and that the profit from each package is its market price.
According to the factory's rules, at least 1,000 sweets must remain in the warehouse (lollipops or chewing gums). In addition, the factory donates 100 lollipops from the warehouse to charity, and if there are fewer than 100 lollipops in the warehouse, then it donates all the lollipops that remain. Note that the 1,000 sweets that must remain in the warehouse are counted before the donation.[3]
For the sweets remaining in the warehouse (after production and after the donation), the factory pays storage costs of \$0.2 per lollipop and \$0.1 per chewing gum.
Write an integer programming problem to find a production plan that maximizes the factory's profits.

1.15. (∘) A French farmer raises a horse (small farm...). He wants to determine the quantities of food he will purchase for the horse at the beginning of each week and the quantities of food he will give to the horse to meet nutritional requirements at a minimum price. The number of units of each nutritional component in each kilogram of food is given in the

[3]For example, if after production, the warehouse contains 90 lollipops and 950 chewing gums, then all 90 lollipops are donated, and the remaining sweets in the warehouse are 950 chewing gums.

following table, along with the weekly nutrition requirements and food prices in the first and second weeks. In the first week, the farmer can purchase food, store some of it in inventory, and use it in the second week. The cost of inventory holding is \$2 per week per kilogram (only paid for the inventory remaining at the end of the first week).

	food products			minimum weekly
	corn	potato	wheat	required quantity
carbohydrates	90	40	20	1400
protein	30	60	80	1260
vitamins	10	50	20	1050
first week cost	42	30	36	
second week cost	50	45	32	

Formulate an LP problem to help the farmer to devise a minimum cost food purchasing plan for a two-week period. Note that the number of kilograms of food the farmer purchases is not necessarily integer.

1.16. (•) Four children play with six toys. Let $c_{i,j}$ be the level of happiness of child i from toy j for $i = 1, \ldots, 4$ and $j = 1, \ldots, 6$. Find the assignment of toys to children that maximizes the total happiness of the children while meeting the following requirements:

- Exactly one child plays with each toy.
- Children 1 and 2 play with one toy each; children 3 and 4 play with 2 toys each.
- Child 3 does not play with toy 5.
- If child 4 plays with toy 2, then child 2 plays with toy 3.

Formulate the problem as a binary programming problem.

1.17. (•) Emily loves watching TV, and tomorrow 8 shows are scheduled, each lasting exactly one hour with no breaks in between. To maximize her enjoyment while keeping her mom happy, Emily cannot watch two consecutive shows. The following table contains the values of level of "enjoyment" for each of the 8 shows.

program	1	2	3	4	5	6	7	8
amount of enjoyment	5	7	3	7	4	3	8	2

(a) Write a binary programming problem to help Emily decide which programs to watch so as to maximize her total enjoyment while meeting her mother's requirement that she does not watch two programs in a row.

(b) Emily's father imposes even stricter rules, mandating that there must be a minimum of a 2-hour break between any two shows she watches. Update your binary programming problem accordingly.

1.18. **Two-knapsacks problem.** We are given two knapsacks and a set of n items, labeled as $1, 2, \ldots, n$. Item i may be packed in at most one of the knapsacks and has a corresponding value v_i if it is packed in the first knapsack and value u_i if it is packed in the second knapsack. In addition, item i has weight w_i. The objective is to choose a collection of items and pack them into the two knapsacks in such a way as to maximize the total value of the chosen collection while ensuring that the total weight does not exceed a predefined capacity C_1 of the first knapsack and C_2 of the second knapsack. Formulate the problem as a binary programming problem.

1.19. In a small toy store there are exactly 15 items, which are listed as follows:

item number	kind	color	size
1	doll	blue	small
2	doll	blue	large
3	doll	yellow	medium
4	doll	yellow	large
5	doll	red	medium
6	doll	red	medium
7	car	blue	small
8	car	blue	small
9	car	blue	medium
10	car	blue	large
11	car	yellow	medium
12	car	yellow	large
13	car	yellow	small
14	car	red	medium
15	car	red	large

The price of item i is c_i, $i = 1, 2 \ldots, 15$. Ann's mother comes to the store and needs to buy a collection of toys so that all of the following conditions are met:

- the collection will have at least one doll and at least one car;
- the collection will have at least two blue items and at least three red items;
- the collection will have at least three large items and at least two small items;
- if the collection contains the red large car (item number 15), then it will not contain any blue cars.

Ann's mother aims to buy a collection of toys that fulfills the above requirements while minimizing the total cost. Formulate the problem as a binary programming problem.

1.20. (•) Harry is a participant in a television game show. In the show, he is presented with 20 tasks; the prize for completing task i is c_i for $i = 1, 2 \ldots, 20$. To help him with the tasks, he is offered a collection of ten assistants whom he can hire; the price of assistant j is a_j for $j = 1, 2 \ldots, 10$. In any case, he can hire at most six assistants. In order to complete task i, Harry needs to hire all the assistants in group $A_i \subseteq \{1, 2, \ldots, 10\}$, which is part of the group of the ten assistants ($A_1, A_2, \ldots, A_{20}$ are given). Formulate a binary programming problem to help Harry decide which assistants to hire in order to maximize his profit—the total amount of money he earns from completing the tasks minus the money he pays for the help of the assistants.

1.21. (•) A company is developing four new products. The company's management must decide which products to produce and how many units to produce. The cost (in dollars) of starting production for each product is given in the first row of the following table. The profit (in dollars) for each unit of product is given in the second row of the table.

products	1	2	3	4
starting production cost	50000	40000	70000	60000
profit for each unit	70	60	90	80

The management imposes the following policy restrictions:

(i) At most two products may be produced.

(ii) Product 3 will not be the only product that is being produced.

(iii) The maximum total number of units that can be produced is 1,500.

(iv) The maximum number of units that can be produced from each product is 1,000.

Formulate the problem as an IP problem.

1.22. (•) **Either-or constraints.** Write the following system of "either-or" inequalities:

$$\begin{array}{ll} \text{either} & 3x_1 + 2x_2 \leq 18 \\ \text{or} & x_1 + 4x_2 \leq 16 \end{array}$$

as an equivalent linear system with mixed integer variables. By "either-or" we refer to the mathematical notion of "or", meaning that the requirement is that at least one of the two inequalities is satisfied. Note that the systems of linear inequalities and equalities that we considered in this book thus far consist of sets of constraints that are satisfied *simultaneously*.

1.23. (○) A furniture store sells 100 sets of furniture, each set containing an armchair and a sofa. The sets are numbered from 1 to 100. The price of the armchair in set i is a_i. The price of the sofa in set i is b_i. The price of the entire set i (armchair and sofa) is c_i with $c_i < a_i + b_i$. Joe decided to buy ten pieces of furniture from the store (any combination of sofas and armchairs) with the following constraint: If Joe buys the three sets numbered 1, 2, and 3, then he will not buy the sofa of set 4.
Joe's goal is to buy ten pieces of furniture at a minimum cost, subject to the above constraint. Formulate this problem as a binary programming problem.

1.24. A factory produces k different types of toys. The factory has m new production machines (k and m are positive integers). The following parameters are given:

- $a_{i,j} = 1$ if machine i can produce toys of type j and $a_{i,j} = 0$ otherwise for $i = 1, 2, \ldots, m$ and $j = 1, 2 \ldots, k$.
- $c_{i,j}$ is the cost of producing one toy of type j on machine i (assuming the machine is able to produce the toy) for $i = 1, 2, \ldots, m$ and $j = 1, 2 \ldots, k$.
- Since the machines are new, they need to be initialized if they are to be used. The initialization cost of machine i is s_i for $i = 1, 2, \ldots, m$,
- Machine i can produce at most u_i toys for $i = 1, 2, \ldots, m$.
- The factory has a demand for at least d_j toys of type j for $j = 1, 2, \ldots, k$.

Write an IP problem which finds a production plan, under the given constraints, that minimizes the production costs.

1.25. (•) **Edge cover problem.** Given an undirected graph $G = (V, E)$, find a smallest possible subset of edges such that every vertex in the graph is incident to at least one edge in the subset. Formulate the problem as a binary programming problem.

1.26. (•) **Bin packing problem.** Given n bins (containers), each of size 1, and n items, where the size of item i is c_i $(0 < c_i < 1)$. The objective is to pack the items in a minimum number of bins. Formulate the problem as a binary programming problem.