Chapter 1
Introduction to Modelling in Mathematical Programming

1.1 Model

In general, a model is a representation of reality (Colin 1973). In that order and in a more extensive way, Pidd (2010) proposes the model definition as “an explicit and external representation of part of reality as seen by people who want to use the model to understand, change, manage and control this part of reality”. On the other hand, Aracil (1983) defines that “a model constitutes an abstract representation of a certain aspect of reality and has a structure that is formed by the elements that characterize the aspect of modelled reality and the relationships between these elements”.

Based on these definitions, we can assume that the objective when creating a model must be to create a representation as complete and as close as possible to reality. And the representation of reality is the representation of the elements that participate in it and their relationships. Based on this, this book aims to be a tool to build models, mathematical models and, more specifically, mathematical programming models, also called optimization models.

García Sabater (2015) defines mathematical models as “formal models that use the language of mathematics to describe a system, expressing parameters, variables and relationships.”

Lowry (1965) distinguishes three types of mathematical models:

Descriptive models: they are used to describe an existing situation. On a set of variables subject to some mathematical equation, results are obtained that inform us about a certain situation.

Prediction or forecasting models: they are used for the simulation of future events. They describe a system over time, which is why some authors include them within the descriptive models.

J. M. García Sánchez, *Modelling in Mathematical Programming*, International Series in Operations Research & Management Science 298,
https://doi.org/10.1007/978-3-030-57250-1_1

Planning or normative models: they are built based on goals and restrictions. They pursue the creation of a plan that better reaches a fixed objective and that is going to be subject to a series of restrictions.

The models of mathematical programming or optimization are planning models in which mathematical relationships are to be expressed through functions. Modelling in the field of mathematical programming corresponds to expressing through mathematical relationships, called constraints, the specifications of a system where a series of activities are performed and represented as variables and in which an optimization criterion or function is followed for its realization. The systems are formed by a set of related elements that favor both the activities of the system and its specifications.

Modelling is the tool that allows us to capture the reality of a system within a mathematical framework, perfectly usable by operational research to find solutions to the problem posed.

The term optimization is fundamental in the area of mathematical programming. Optimization consists in the maximization or minimization of a function, known as the objective function. Mathematical programming looks for valid solutions that represent the activity of a system and that can be evaluated with respect to that objective function.

Generally, modelling has been considered a little regulated technique, based on knowledge of the types of optimization problems and the experience of the modeller. This book aims to provide a methodology for the construction of a mathematical model in an integral way, as well as techniques that help us if we want to follow our own building criteria. The objective is to provide a simple work dynamic that facilitates the modelling process.

An important aspect in modelling is the description of the system to be modelled. It is necessary to perfectly identify the elements that are part of the problem and all the characteristics that are relevant. We should not describe parts of the system that are not part of the problem. In modelling, it is fundamental to start with an exhaustive and precise description and, from this, to lead the modeller to a correct definition of the components of the model.

The book does not deal with the search processes for solutions of the different resolution methodologies. For this there are countless bibliographical references that the reader can use.

1.2 Classical Components of a Mathematical Programming Model

A basic model of mathematical programming consists of four components (Castillo et al. 2002):

- *Data*: the deterministic values that the model handles and that are represented in a simplified way with the use of parameters.
- *Variables*: these define the decisions that occur in the problem. The variables always represent what there is to find out their value. Regarding the value, the variables of a model can be continuous or integer, and within the integer variables, a special and important type of variable is distinguished, the binary variables, which only take the values 1/0 (true/false).
- *Constraints*: equalities and mathematical inequalities that define the specifications and rules of the problem. The constraints are mathematical relationships between the variables and the problem data. The type of relationship determines the type of model as we will see in Sect. 1.3 of this chapter.

On the other hand, sign constraints associated with the variables of the problem are also imposed.

- *Objective Function*: this defines the optimization criterion, which will maximize or minimize a function.

As we will see next, within the types of mathematical programming models, we can work with variants regarding the type of data, the linear character of the mathematical functions, and the number of constraints and objective functions.

1.3 Classification of Mathematical Programming Models

The mathematical programming models or optimization models depend fundamentally on their resolution of the type of variables that are part of it and the linear or non-linear character of the mathematical expressions that compose it. However, there are other factors that also define the classification of the models:

- According to the model data:
  - Deterministic models: all model data are known and accurate.
  - Stochastic models: the data have a random or probabilistic component.

This methodology focuses exclusively on mathematical deterministic models, but independently of this, some of its techniques for the elaboration of probabilistic models could be used.

- According to the number of objectives of the problem and the number of restrictions:

We can work with mathematical models that only have constraints, optimization problems with an objective function (most common case), or optimization problems with several objective functions. Similarly, there are optimization problems that do not have constraints and only have an objective function.

- According to the type of functions that make up the model and its variables:
  - Linear Programming Models: all mathematical expressions are linear.
  - Non-linear Programming Models: there are some non-linear expressions, either in the constraints or in the objective function.

Within the linear models, we distinguish between:

- (Continuous) Linear Programming Models: all the variables are continuous.
- Integer Linear Programming Models: there are integer variables. Within these, the following are distinguished:
  - Pure Integer Linear Programming Models: all are integer variables.
  - Mixed Integer Linear Programming Models: there are integer and continuous variables.
  - Binary Linear Programming Models: all variables are binary.

1.4 First Example

In order to quickly introduce the concept of modelling in mathematical programming, let us first take a look at an example of what it takes to obtain a mathematical model from the description of a system. For our first example, we will consider a very simplified production system.

Butter Production
Imagine a butter production factory that wants to optimize its daily production of butter. Two types of butter are made (Sweet and Raw). A kilo of sweet butter gives the manufacturer a profit of $10 and a kilo of raw a profit of $15. For the production of butter, two machines are used: a pasteurization machine and a whipping machine. The daily use time of the pasteurization machine is 3.5 hours and 6 hours for the whipping machine.

The time (in minutes) consumed by each machine to obtain a kilo of butter is shown in Table 1.1:

To gain simplicity, we ignore the input components for the production of butter (cream, water, salt, preservatives, etc.) and any cost generated in the process.

Let's move on to the identification of the basic components of the mathematical model (variables, constraints, and objective function). The Data component is displayed, while the constraints and objective function are identified.

1. Variables

As mentioned, the variables of a model represent the actions or activities that occur in the system and on which it is necessary to decide a value. In our case, the activity in this system is the production of butter, specifically, the activity of producing sweet butter and raw butter. What we need to find out is how much

Table 1.1 Butter processing times

	Sweet butter	Raw butter
Pasteurization	3	3
Whipping	3	6

sweet butter to produce and how much raw butter to produce. To represent these two actions, we define two variables:

- $x1$: Amount of sweet butter to be produced
- $x2$: Amount of raw butter to be produced

Since butter production is measured in kilograms, these two variables will take continuous values.

We assume that a negative number of kilos of butter cannot be produced, so the sign of the variables is established as:

$$x_1 \geq 0$$
$$x_2 \geq 0$$

2. Constraints

The activities of the system require two processes to be carried out: pasteurization and whipping. Each of them is carried out in a different machine. There are a pasteurization machine with a working capacity of 3.5 h/day and a whipping machine with a capacity of 6 h/day.

We need the use of these two machines to carry out our activities. Therefore, we could consider machines as resources of the production system.

Resources:

– Pasteurization machine
– Whipping machine

The operating specification that these resources impose on the system is its capacity. Therefore, the total consumption of these resources must not exceed their capacity. In other words:

Consumption in the pasteurization machine $\leq$ Capacity of the pasteurization machine
Consumption in the whipping machine $\leq$ Capacity of the whipping machine

The consumption of each resource is generated by each of the system's activities. Sweet butter production consumes time on each machine, as does the production of raw butter, according to the values shown in Table 1.1.

Focusing on the pasteurization machine, it is pointed out that:

1 Kg of sweet butter consumes 3 min of pasteurization; therefore:
2 Kg of sweet butter consumes 6 (3×2) min of pasteurization.
3 Kg of sweet butter consumes 6 (3×3) min of pasteurization.
. . .

so:
x_1 Kg of sweet butter consumes 3 × x_1 min of pasteurization.
Identical analysis for raw butter generates a consumption of $3x_2$.
The total consumption in minutes would be expressed therefore as:
$3x_1 + 3x_2$ min

The capacity of the pasteurization machine was 3.5 h/day = 210 min/day. We express capacity and consumption in the same unit, minutes, and the mathematical expression that defines the constraint of the use of that resource would be as:

$$\mathbf{3x_1 + 3x_2 \leq 210}$$

In a similar way, the constraint associated with the resource of the whipping machine would be developed, obtaining as a final set of system restrictions:

– Pasteurization machine ⇒ $\mathbf{3x_1 + 8x_2 \leq 210}$
– Whipping machine ⇒ $\mathbf{3x_1 + 6x_2 \leq 360}$

3. Objective Function

The optimization criterion of the system is the maximization of the benefit of production. The system has the profit produced by the two activities that constitute the production process, that is, it is known that a profit of $10 is obtained for the production of each kilo of sweet butter and $15 for the production of each kilo of raw butter. Mathematically defining profit expression:

Total Profit = Profit of sweet butter production + Profit of raw butter production

Profit of sweet butter production:

1 Kg of sweet butter ⇒ Profit = $10
2 Kg of sweet butter ⇒ Profit = 10 × 2 = $20
3 Kg of sweet butter ⇒ Profit = 10 × 3 = $30
...
x_1 Kg of sweet butter ⇒ Profit = $$10x_1$
Profit of raw butter production = $15x_1$
Total Profit = $10x_1 + 15x_2$

Therefore, the maximization of the benefit is defined as **Maximize $\mathbf{10x_1 + 15x_2}$.**

4. Complete Model

Complete model is defined as:

Maximize $10x_1 + 15x_2$
subject to
$3x_1 + 3x_2 \leq 210$
$3x_1 + 6x_2 \leq 360$
$x_1 \geq 0$
$x_2 \geq 0$
x_1, x_2 continuous

With this we have just expressed a first model with two continuous variables, two linear constraints and a linear objective function. A solution of the model is any combination of values of the problem variables (x_1, x_2) that satisfy the imposed specifications. Example ($x_1 = 20$, $x_2 = 30$) is a solution that follows the specifications:

$$3 \cdot 20 + 3 \cdot 30 \leq 210 \quad \Rightarrow \quad 150 \leq 210$$
$$3 \cdot 20 + 6 \cdot 30 \leq 360 \quad \Rightarrow \quad 240 \leq 360$$
$$20 \geq 0$$
$$30 \geq 0$$
$$\text{Solution profit}: 10 \times 20 + 15 \cdot 30 = \$650$$

The mission of the resolution methods is to find the optimal solution or one close to the optimum. As already mentioned, the study of resolution methods is not the subject of this book. For this problem, the optimal solution is reached for the production values ($x_1 = 20$, x_2=50) with a profit of \$950 (10 × 20 + 15 × 50).

This first example has served to establish the correspondence between System and Model, about what it means to transform the written description of simple production processes into a set of mathematical expressions. From the next chapter, we will define our methodology from a broader perspective.

1.5 Considerations on the Format of a Mathematical Model

The general format of a model can be represented as follows:

$$\begin{aligned} &\text{Min} \quad f(X) \\ &\text{subject to} \\ &G_i(X) \approx b_i i = 1 \ldots m \end{aligned}$$

where:
X: Vector of variables
$f(X)$: Objective function
$\approx$: Sign of the constraints [$\leq$; $=$; $\geq$]
$b = (b_1 \ldots b_i \ldots b_m)$: Vector of independent terms
$G_i(X) \approx b_i$: Set of m constraints, i = 1 ... m

An extended representation of this format could be seen as follows:

$$
\begin{aligned}
&\text{Min} \;\; c_1x_1 + c_2x_2 + \ldots + c_jx_j + \ldots + c_nx_n \\
&s.t \\
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n \approx b_1 \\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n \approx b_2 \\
&\ldots \\
&a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n \approx b_i \\
&\ldots \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mj}x_j + \ldots + a_{mn}x_n \approx b_m \\
&x_1 \geq 0 \quad x_2 \geq 0 \quad \ldots \quad x_j \geq 0 \quad \ldots \quad x_n \geq 0
\end{aligned}
$$

The objective of the problem is considered in a generic way as to minimize the objective function. The expression of maximizing a function can be transformed into minimizing by a simple transformation:

$$\text{Max}\, f(x) \Rightarrow -\text{Min} - f(x)$$

The first sign does not affect the resolution of the model, only the value resulting from the objective function, in which case the sign should be changed, that is, the value of the objective function of any solution obtained through the minimize problem will have to change sign to convert it to the maximize problem.

The second sign involves changing the sign of the original objective function. Let us see the transformation of our first example:

$$
\begin{aligned}
\text{Maximize } 10x_1 + 15x_2 &\Rightarrow -\text{Minimize} - (10x_1 + 15x_2) \\
&\Rightarrow -\text{Minimize} - 10x_1 - 15x_2
\end{aligned}
$$

The second important aspect in the definition of any model is the sign of the constraints. Any constraint is limited to the use of three signs: $\leq$, $=$, and $\geq$.

In literature it is possible to find the generic definition of a model as:

$$
\begin{aligned}
&\text{Min} \quad f(X) \\
&\text{subject to} \\
&G_i(X) \leq b_i \swarrow\!\!\!\nearrow i = 1 \ldots m
\end{aligned}
$$

This representation is even more simplified, but equivalent:

In the case of having a constraint with sign $G_i\,(X) \geq b_i$, it is sufficient to multiply the restriction by -1 to convert the sign: $-G_i(X) \leq -b_i$.

In the case of having a constraint with sign =, which is equivalent to two similar constraints, one with sign ≤ and one with sign ≥: $G_i(X) = b_i \Rightarrow G_i(X) \leq b_i$; $G_i(X) \geq b_i$.
With the signed restriction ≥, proceed as before: $-G_i(X) \leq -b_i$.
Therefore, $G_i(X) = b_i$ is equivalent to $G_i(X) \leq b_i$ and $-G_i(X) \leq -b_i$.
So, any model can be expressed with sign ≤.

On the other hand, the use of the greater-than and the less-than sign is not allowed. In the case that a system finds a constraint that is defined with those signs, a small modification must be made to express it correctly. Imagine a generic constraint $G_i(X)<b_i$. The way of operating is as follows:

If all the variables are integer, the problem is solved by defining as independent term b_i-1, since it is the maximum value that the expression $G_i(X)$ could reach. In the case of the greater-than sign, $G_i(X)>b_i$, the equivalent constraint is defined as $G_i(X)\geq b_i + 1$.

If there are continuous variables, then for the case of <, it may be that the expression $G_i(X)$ will take values between $b_i -1$ and b_i, so the previous transformation could not be performed. In this case, it is necessary to commit a minimum controlled error in the constraint, defining the value ε as small as we want, and the constraint is defined as $G_i(X) \leq b_i-\varepsilon$. In the case of >, $G_i(X) \geq b_i+\varepsilon$.

The adjustment of the value of ε is made according to the precision of the values that the continuous variables could take and therefore the function $G_i(X)$.

Finally, if an original variable is coming defined as a negative variable $x_j\leq 0$, then we do a substitution of variables. We let $y_i = -x_i$. Then $y_i \geq 0$. And we substitute $-y_i$ for x_i wherever x_i appears in the model. And in the case where x_i is a free variable, unconstrained in sign, we substitute in the model the free variable x_i by the difference of two nonnegative variables, $x_i = e_i - v_i$, $e_i\geq 0$ and $v_i\geq 0$. If necessary, we can impose a propositional logic specification in the model so as not to allow the two variables e_i and v_i to take positive values.

1.6 Justification of the Use of Mathematical Programming Models

The advancement of technology and optimization libraries contributes significantly to the importance of using mathematical models as a way of solving this kind of problems. For decades, mathematical models have been formulated to represent any optimization problem, but in most cases, these models have been used as an insubstantial mathematical contribution, which only allowed for obtaining solutions for instances with few data. Nowadays, thanks to technology, it is possible to solve larger problems. For this reason, optimization libraries have proliferated with a more advanced methodological development, which allow the possibility of solving optimization problems counting only on the mathematical model, without the intervention of the user and without the need to elaborate complex exact or heuristic methods to obtain solutions.

When we talk about solving mathematical programming models, we talk about the search for optimal solutions, that is, exact methods of resolution. In an auxiliary way, mathematical programming can participate in approximate procedures, but on subproblems of the original one, or using time as a factor in the completion of the search process of the optimal solution and collecting the best solution found. But in a primary way, the resolution of mathematical programming models is focused on the search for an optimal solution.

There are three fundamental factors that affect the choice of how to solve a problem:

- The computational complexity of the problem: referred to if the problem fits within the problems of class P (polynomial) or class NP (non-polynomial, and not class P) (Dean 2016).
- The size of the problem: the size of a problem is determined by the number of elements involved in it.
- The temporary nature of the problem: there are operational problems for which a solution is needed in a very limited time. The problem is solved very often, either because the solution is used in a short period of time or because it is necessary to test many alternatives with respect to the data. On the other hand, there are tactical and strategic problems in which the solution is going to be used for a long time and we are allowed the license to be able to wait a while until obtaining a solution.

The entire optimization problems framed in the computational class P can be solved optimally without the need of the mathematical model. Above them, the heuristic resolution does not fit, regardless of the size and temporary nature of the problem. Likewise, the use of the model to obtain optimal solutions is also usually a feasible option.

The problems of linear programming, only with continuous variables, are considered non-complex problems and are solved optimally with the mathematical model. All existing exact procedures are standards that make use of the model.

When the problem is not considered class P and it is considered NP, then the size of the problem and its temporary nature come into play. In the first place, it must be mentioned that each optimization problem in this typology needs its own analysis regarding the resolution time to obtain optimal solutions. There are NP problems for which the mathematical model presents very good behavior, even for instances of a considerable size. On the other hand, there are others where, even for a small size, the exact resolution procedure uses a lot of time (e.g., many scheduling problems). For these types of problems, the option of obtaining the optimal solution requires the mathematical model in the vast majority of cases. There is a very limited group on which other exact techniques can be used, such as dynamic programming, but this possibility is very limited.

The other resolution option for NP problems is the heuristic resolution, for which we do not need the mathematical model. It is the option to choose when there are time limitations and the exact method uses too much time. It may even be the best

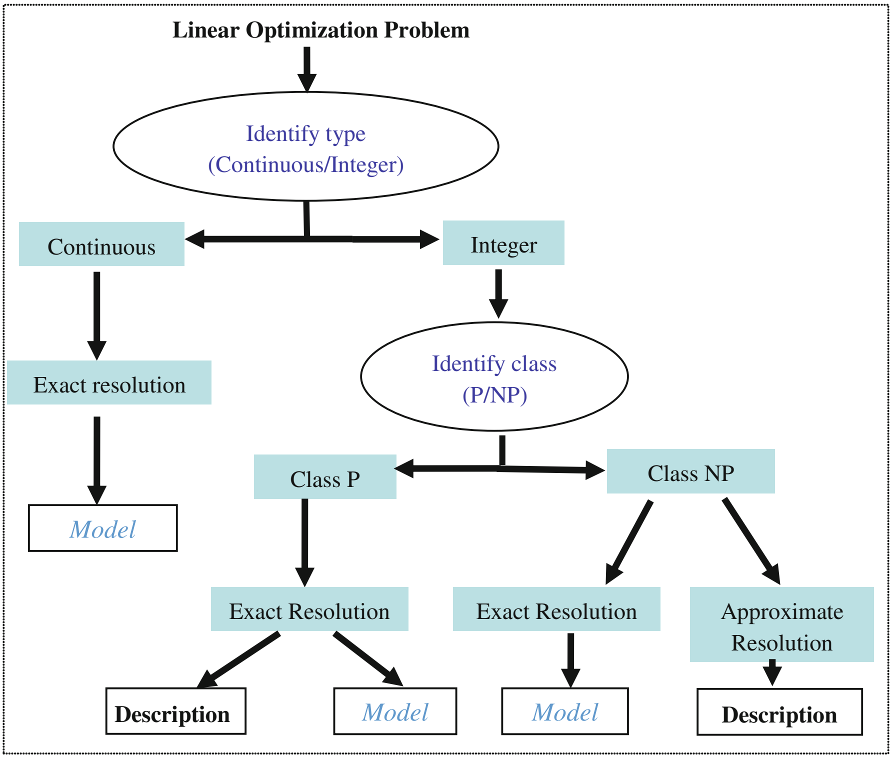

Fig. 1.1 Resolution alternatives

option if the system data are unreliable or simply because it is not so necessary to obtain the optimal solution.

Figure 1.1 summarizes the resolution options for an optimization problem.

Real systems tend to have a large volume of data and specifications. Almost all of them usually have a computational complexity of NP type. Class P problems represent a very small percentage of optimization problems. All this contributes to the fact that nowadays heuristic methodologies are the first option for solving a problem. In any case, I would always recommend the use of the mathematical model and the exact resolution with optimization libraries as a first resource to try to solve a problem.

From a futuristic perspective, both the arrival of increasingly faster processors and the evolution and the rise of computer networks that allow parallel and distributed computing will contribute to the exact resolution of problems using existing techniques (such as the branch and bound method) being an increasingly important option.

It is essential to model with the resolution in mind and to provide a model that obtains solutions as quickly as possible, so to make the most appropriate model for the later resolution is a job for the modeller.

References

Aracil, J. (1983). *Introducción a la dinámica de sistemas*. Madrid: Alianza Editorial.

Castillo, E., Conejo, P., & Pedregal P. (2002). Formulación y resolución de modelos de programación matemática en ingeniería y ciencia. Universidad de Castilla La Mancha Ediciones.

Colin, L. (1973). *Models in planning: An introduction to the use of quantitative models in planning*. Oxford: Pergamon Press.

Dean, W. (2016). Computational complexity theory. In *The Stanford encyclopedia of philosophy*. Stanford: Metaphysics Research Lab, Stanford University.

García Sabater, J. P. (2015). http://personales.upv.es/jpgarcia/LinkedDocuments/modeladomatematico.pdf. Modelado y Resolución de Problemas de Organización Industrial mediante Programación Matemática Lineal. Accessed May, 2020.

Lowry, I. S. (1965). A short course in model design. *Journal of the American Institute of Planners, 31*, 158.

Pidd, M. (2010). *Tools for thinking: Modelling in management science* (3rd ed.). Chichester: Wiley.