

Chapter 2

Structure of a Mathematical Programming Model



2.1 Environment of an Optimization Problem

Optimization problems arise within diverse environments of any nature. In this book we will denominate the environment as a system, so that the system is where the optimization problem is defined, and it is the system that has the elements that participate in the problem, the specifications and the criterion that directs the optimization.

Aracil and Gordillo (1997) refer to the system as an aspect of reality endowed with a certain complexity because it consists of interacting parts. We can speak of a system when we refer to any organized environment on which a specific management of its elements is proposed, whose objective is to achieve goals under an objective criterion. In our environment, we will work with static systems, those in which the information of their elements does not change over time. There are also dynamic systems, which are solved with other techniques.

Static systems can be considered and solved by optimization techniques. The objective of this methodology is to convert the description of the system, that is, the description of an optimization problem, to a mathematical language usable by optimization techniques in mathematical programming, which is known in terms of operational research such as “modelling the problem.”

Traditionally, whenever an optimization problem is defined, a management situation is described within a given environment. The model is the mathematical representation of the problem. It is perfectly admissible, and just as appropriate, to refer to the term “model the system” instead of “model the problem,” meaning “modelling the system,” how to model its operation or model its management. The result in any case will be a mathematical model that we can refer to as a mathematical programming model or optimization model, which will later be solved to obtain a solution for its management.

2.2 Components of an Optimization Problem

In a system where an optimization problem is defined, the following components are distinguished:

Elements

The elements are all the actors that participate in the system. They are diverse in nature, from people, tools, places, time, etc.

They usually have associated information that we will call data, and that must be numerical information or to be defined numerically.

The elements participate in the actions that occur in the system and also support its specifications. They are closely related to the activities that occur in the system, and sometimes the joint identification of both components is effective.

The elements are configured taking into account their nature, their associated data, and their reference in the specifications.

Decision Activities

These are direct actions that occur in the system for which it is necessary to decide their value, which is not determined. They are associated with the elements. The decision activities are simple actions. They cannot be the result of a logical calculation, simple function, or combination of other decision activities.

Decision activities define the main variables of a model. The values that a variable can have will be a binary value (1/0), an integer value, or a continuous value.

Calculations

Based on the decision activities, a system may need additional information that is obtained from these decisions through a calculation process. Even calculations can be generated from other calculations, not only from the base decisions of the system. The calculations are also represented as variables, and in classical formulation, they are also considered decision variables, although they are really only decision variables for the resolution processes not for the operation of a system.

The way of expressing the relationship with the variables on which it depends can be through a mathematical function, as a result of a logical proposition, or acting as a bound of those variables. According to this, we will have three types of calculations:

Auxiliary Calculations

Represent the calculations obtained from the result of a linear mathematical function. Normally, they are used for a concept of comfort and clarity in the construction of the model, without being obliged to use it, but they are mandatory when we want to discretize the value of a function, that is, when we want the value of a function to be integer. They give rise to the auxiliary variables of a problem.

Logical Calculations

They are used to obtain the result of a logical proposition (Klement 2006). Its use is mandatory. The variables generated will be referred to as logical variables.

Lower/Upper Bound Calculations

They are used to obtain the upper or lower bound of a set of variables. Its use is also necessary.

Specifications

Regulations, impositions or limitations that must be fulfilled in the system. They give rise to the constraints of the problem, but specification should not be confused with constraint. Constraint is a single mathematical expression, whereas the specification is a characteristic that the system must fulfill and that is implemented in one or more constraints.

Objective Criterion

This can be understood as one more specification of the system, expressing the criteria that guide the resolution of the system. The objective will give rise to a linear function of costs, costs that are associated to the elements of the system and that is known as objective function. In certain objectives, the criterion will also lead to the use of calculations and the definition of specific constraints.

This book will follow a sequential approach to the learning of each of the components, practically following the order presented now. However, to explain certain concepts that will appear throughout the methodology, it will sometimes be necessary to refer to components that have not yet been seen. We will try to do this as simply as possible, leaving the most advanced concepts for when all the components have been studied.

Figure 2.1 represents the relationships between the different components. The elements, owners of the system information, relate to all the components of the system with a special influence on the activities. On them fall the specifications, calculations, and the optimization criterion. And these components also determine the typology of the elements, so the relationship is bidirectional.

The decision activities participate in the specifications and the objective criterion. The calculations that are defined as variables also participate in specifications and the objective function. They can be necessary to define specifications and the objective criterion. Likewise, the definition of a calculation is represented as a specification (Fig. 2.1).

Relationship of Proposed Components and Classical Components

As is logical, the classical components (García Sabater 2015; Castillo et al. 2002) are quite equivalent to the components proposed in this methodology, with hints that are displayed in the following figure (Fig. 2.2):

The elements provide all the data, parameters and subscripts that the problem will have, with an organized structure that will allow its subsequent implementation in an optimization library. The elements are also used with special relevance in the definition of problem decision activities, and its quantitative nature will determine the type of variable. The quantitative nature will depend on its ability to be measured in the system, its degree of determination, its data, and how the element is identified in the system.

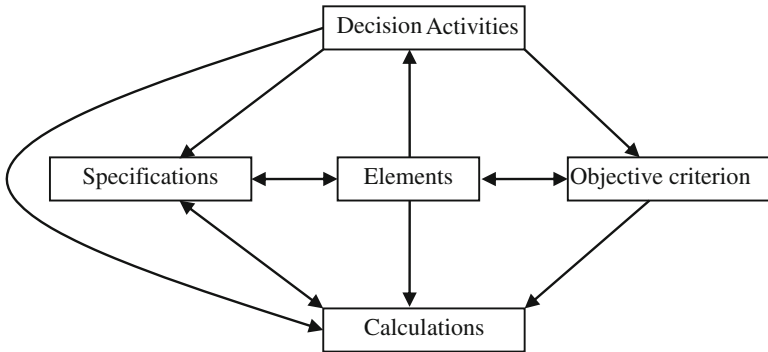


Fig. 2.1 Relationship between the proposed components

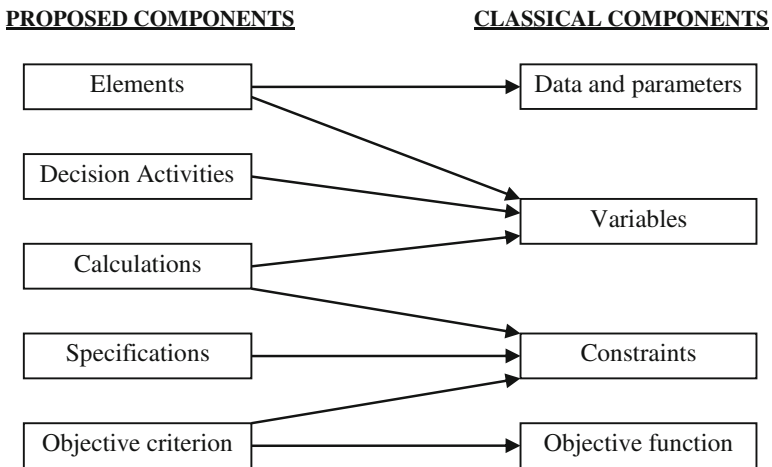


Fig. 2.2 Relationship between components

The decision activities correspond to variables of the problem. The calculations also generate variables and always have associated constraints. The specifications, as already mentioned, define constraints. The objective criterion defines the objective function and may also come with constraints.

2.3 Examples

To illustrate the components of a system, let us look at a couple of simple examples. The first one was modelled in Chap. 1. We use it here simply to repeat its modelling but with the components of this methodology. The second system is a sales system. Both will be used as illustrations for many explanations in future chapters.

Illustration 2.1: Production of Butter

A butter production factory 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 h and 6 h for the whipping machine.

The time (in minutes) consumed by each machine to obtain a kilo of butter is shown in the following table:

Table. Butter processing times (in minutes)

	Sweet butter	Raw butter
Pasteurization	3	3
Whipping	3	6

1. Elements

Analyzing the statement, we can identify the following participating elements:
Regarding the resources used in production:

- Pasteurization machine: The numerical information associated with this element in the system is:
 - The usage time = 3.5 h.
 - The time that a kilo of sweet butter uses in the machine = 3 min.
 - The time spent in the machine by kilo of raw butter = 3 min.
- Whipping machine: The associated information coincides with the information of the pasteurization machine, that is, the time of use (6 h), the time used by a kilo of sweet butter (3 min), and the time used by a kilo of raw butter (6 min).

Regarding the concepts produced in the system:

- Sweet Butter: The information of this element in the system has already been partly reflected, and it is the time that a kilo of this element needs in each machine. In addition to these times, there is also a profit per kilo = \$10.
- Raw Butter: Element similar to the other butter and with the same information.

We could also consider as an element another actor that appears in the text, the factory itself, which would play the role of system. We can verify that it does not have any characteristic that is translated into a certain value, nor will there be any activity that must necessarily be associated with it. In any case, its definition as an element does not complicate the elaboration of the model.

2. Decision Activities

To recognize the decision activities, it is necessary to explore the actions that take place in the system. To these actions we must assign a type of value (binary, integer, or continuous) and that the value of the action is not determined.

The methodology we will use from the next chapter to obtain all the components of the system will allow us to obtain the activities and assign them the appropriate type of value. In our example we identify as decision activities:

Production of sweet butter

It is an action associated with the sweet butter element.

It is necessary to examine what type of value this activity has:

If we propose a binary value (1, sweet butter is produced; 0, no sweet butter is produced), we would not obtain complete information of the activity. As we will study in the chapter on decision activities, the characteristics of the element discard this type of value for the activity.

If we propose an integer value, we would be restricting the set of values to only integer values. However, production is measured in kilos and this is a continuous measure. Therefore, the appropriate value for the activity is the last of the options, a continuous value.

Defined the type of value, it is necessary to ask whether it is a known value, a value to be determined by a function or an indeterminate value. Being an indeterminate value, we can affirm that it is a decision activity in the system.

Production of raw butter

Activity of identical nature to the previous one.

A decision variable is generated from each activity. Therefore, the decision activities associated with producing butter would give rise to two variables:

x_1 : Kilos of sweet butter that are produced; $x_1 \geq 0$

x_2 : Kilos of sweet butter that are produced; $x_2 \geq 0$

3. Calculations

It is not necessary to perform calculations on the problem. Nor is it necessary to provide clarity to the model since its size is very small.

4. Specifications

Extracting the specifications is the task for which a greater clarity in the description of the system is needed.

In the study of the types of specifications that we will carry out in Chap. 6, we will see that a very common specification occurs when some element acts as a resource. In that case, the element must have associated a numerical characteristic that represents availability or capacity, and there must be decision activities that consume that resource.

In our case, both the pasteurization machine and the whipping machine act as resources.

Regarding pasteurization:

Capacity = 3.5 h = 210 min

Consumption:

x_1 (sweet butter production) consumes 3 min for each unit that the variable takes.
 x_2 (raw butter production) consumes 3 min for each unit that the variable takes.

Constraint generated: Consumption \leq Capacity

$$3x_1 + 3x_2 \leq 210$$

(It is necessary to work with the same unit of measurement, in this case, minutes.)

Regarding whipping, a specification of the same nature and with the same participating activities is applied. Constraint generated: $3x_1 + 6x_2 \leq 360$.

5. Objective Criterion

The criterion is based on maximizing profits. The two existing activities provide profits:

- The production of sweet butter, x_1 , provides a unit profit of \$10 (profit per unit of the variable).
- The production of raw butter, x_2 , provides a unit profit of \$15 (profit per unit of the variable).

Profit expression: $10x_1 + 15x_2$.

The objective function would be Max $10x_1 + 15x_2$.

Complete Model

$$\text{Max } 10x_1 + 15x_2$$

s.t.

$$3x_1 + 3x_2 \leq 210$$

$$3x_1 + 6x_2 \leq 360$$

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

Illustration 2.2 (Source: Meléndez 2019): Sale of Batches

A department store has 200 shirts and 100 trousers from the previous season. They launch two batch offers, A and B. The Offer A consists of a batch of one shirt and one trousers, which is sold at \$30; Offer B consists of a batch of three shirts and one trousers, which is sold for \$50. They do not want to launch less than 20 batches of Offer A. On the other hand, they also have the option of transferring pants to an outlet for a price of \$18/trousers, agreeing to pay a fee to the outlet of \$80 if they transfer more than 50 units. The objective is to maximize the profit.

1. Elements

Analyzing the statement, we can identify as participating elements:

- Shirts: The numerical information associated with this element in the system is:
 - Stocks of Shirts = 200.

As we will justify in the chapter dedicated to Elements, it would be an error in this case to consider each shirt individually as an element:

- Number of shirts in a Batch Offer A = 1

Here is something that we will also analyze in the next chapter – numerical characteristics that can be associated with more than one element:

- Number of shirts in a Batch Offer B = 3
- Trousers: Similar to the previous element. The associated information is:
 - Stocks of trousers = 100
 - Number of trousers in a Batch Offer A = 1
 - Number of trousers in a Batch Offer B = 1
 - The transfer value to the Outlet = \$18
 - Maximum number of trousers sold to the outlet without paying fee = 50
 - Fee for transferring trousers to the outlet = \$80
- Batch Offer A: The associated information is:
 - Number of shirts = 1

As we have already mentioned, this feature is also associated with the Shirts element:

- Number of trousers = 1
- Sale price = \$30
- The minimum number of batches = 20
- Batch Offer B: It has the same characteristics as Batch Offer A, except the minimum number of batches, but with different values.
- Outlet:
 - Sale price of trousers to the outlet = \$18

Since the system we model is associated with the department stores, it is more logical to use descriptions regarding the system, not with respect to each element. For this reason, we use the name of the attribute “Sale price” and not Purchase price, despite the fact that the Outlet buys trousers.

- Maximum number of trousers sold to the outlet without paying fee = 50
- Fee for transferring trousers to the outlet = \$80

2. Decision Activities

To recognize the decision activities, it is necessary to explore the actions that occur in the system.

A department store has 200 shirts and 100 trousers from the previous season. They **launch two batch offers**, A and B. The Offer A consists of a batch of one shirt and one trousers, which **is sold** at \$30; Offer B consists of a batch of three shirts and one trousers, which **is sold** for \$50. They do not want to launch less than 20 batches of Offer A. On the other hand, you also have the option of **selling trousers to an outlet** for a price of \$18/trousers, agreeing **to pay a fee** to the outlet of \$ 80 if they transfer more than 50 units. The objective is to **maximize the profit**.

The actions of launching batches and selling batches can be considered equivalent in the system. They refer to the same process.

Launch Batches A and B

It is an action associated with both the Batch Offer A element and the Batch Offer B element, independently. This generates two decision activities:

- Launch Batch Offer A
- Launch Batch Offer B

The fact that the action corresponds to a decision activity is in the analysis of the action, with respect to whether it has a known or unknown value, calculated or not calculated. The first should be to examine what kind of value that activity has. If we proposed a binary value, we could not determine the number of Offers A and B launched. If instead we assign an integer value, the semantics of the action should refer to the number of batches that are launched. This is something indeterminate in the system; we do not know the number of batches that we are going to launch, nor can we calculate it from other decision-making activities. This type of value would determine the decision variables of the problem.

Finally, it would not make sense to use a continuous value since the batches are discrete elements.

Therefore, the decision activities associated with launching batches would give rise to two variables:

x_A : Number of **Batch Offer A launched**

x_B : Number of **Batch Offer B launched**

For this same decision activity, the statement also uses the verb sell. It is common in many descriptions to use different verbs to express the same action.

Transfer Trousers to the Outlet

We are faced with an action related to the element Trousers and the Outlet element jointly. Again, the value that we can associate with this action is an Integer value, and the variable that is generated could be defined as:

x_{PO} : Number of **Trousers transferred** to the **Outlet**.

Pay a Fee

It cannot be considered a decision activity because its value is obtained by calculation with respect to other decision activities.

Maximize the Profit

It is about the objective function. It will correspond to a variable function of the problem. It is therefore not an action that can be considered a decision activity.

3. Calculations

As mentioned, there is a cost associated with a calculation, in this case a logical calculation. The value of paying the fee of \$80 will be defined by a logical function or proposition.

For the modelling of a logical calculation, it is necessary, first of all, to define the variable that will collect the result of the calculation. In this case it is a calculation with binary value ($= 1$ is to pay the fee; $= 0$ is not to pay the fee).

Let the variable $\alpha = 1$ if we pay the fee; 0 otherwise.

And secondly, it is necessary to state the logical proposition that defines the calculation values:

“If the number of trousers sold to the Outlet is greater than 50, we pay a fee of \$80. In another case, we do not pay anything.”

That would be mathematically expressed as:

IF $x_{PO} > 50$, THEN $\alpha = 1$; IF $x_{PO} \leq 50$, THEN $\alpha = 0$.

Chapter 5 is devoted to the calculations of a system, and in Chap. 6 we will learn to model logical propositions. Specifically, this proposition could be modelled with a single constraint:

$$x_{PO} \leq 100\alpha$$

4. Specifications

From the statement and the elements of the problem, we extract the following specifications:

An imposed rule is easily identified: “You do not want to launch less than 20 batches of Offer A.”

This specification refers to the activity of launching batches of Offer A. It imposes a lower bound to the value that the variable x_A can take:

$$x_A \geq 20$$

The elements Shirts and Trousers have data of availability or capacity and then must generate specifications that ensure that the consumption of that availability in the system does not exceed the availability data:

Regarding the Shirts:

Capacity = 200

Consumption:

x_A (Launch Batch Offer A) consumes one unit (one shirt) for each unit that the variable takes (each batch that is sold).

x_B (Launch Batch Offer B) consumes three units for each unit that the variable takes.

Constraint generated: $x_A + 3x_B \leq 200$

Regarding the Trousers:

Capacity = 100

Consumption:

x_A (Launch Batch Offer A) consumes one unit for each unit that the variable takes.

x_B (Launch Batch Offer B) consumes one unit for each unit that the variable takes.

x_{PO} (Sell Trousers to the Outlet) consumes one unit for each unit of the variable.

Constraint generated: $x_A + x_B + x_{PO} \leq 100$

5. Objective Criterion

The criterion is based on maximizing the benefit. Three existing activities provide income and a logical calculation provides cost.

The Offer A batches, x_A , provide a unit income of \$30 (profit for each unit sold).

The Offer B batches, x_B , provide a unit income of \$50.

The transfer of trousers, x_{PO} , provides a unit income of \$18.

Pay the fee: α provides a unit cost of \$80.

Profit expression: $10x_A + 15x_B + 18x_{PO} - 80\alpha$.

The objective function would be $\text{Max } 10x_A + 15x_B + 18x_{PO} - 80\alpha$.

Complete Model

$\text{Max } 10x_A + 15x_B + 18x_{PO}$

s.t.

$x_{PO} \leq 100\alpha$

$x_A \geq 20$

$3x_A + x_B \leq 200$

$x_A + x_B + x_{PO} \leq 100$

$x_A \geq 0, x_B \geq 0, x_{PO} \geq 0$ integers; α binary

References

- Aracil, J., & Gordillo, F. (1997). *Dinámica de Sistemas*. V 168 Alianza Universidad Textos. 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.
- 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.
- Klement, K. C. (2006). Propositional logic. In J. Fieser, & B. Dowden (Eds.), *Internet encyclopedia of philosophy*
- Meléndez, I. (2019). <https://www.monografias.com/trabajos96/distribucion-redes-administracion-proyectos/distribucion-redes-administracion-proyectos.shtml#bibliograa>. La distribución de redes y la administración de proyectos. Accessed June 2019.