Chapter 7
The Quantitative Nature of the Elements

7.1 Introduction

We have already seen throughout the book that the nature of the elements is used as a tool to help define decision activities. It is probably an unnecessary tool for an experienced modeller, but it may be useful for people who start modelling in mathematical programming. The factors that determine the quantitative nature of the element are its data and its properties, as well as the treatment it receives in the description of the problem.

Figure 7.1 defines the decision scheme of the quantitative nature of an element.

The most important nuances that can be considered regarding the definition of the quantitative nature of the elements can be summed up in six cases. Let us see each of them:

- Case 1. Individual element not measurable defined as measurable

An individual element that has a continuous quantity attribute may appear a priori to be measurable due to that continuous attribute and may even determine decision activities where the quantity of the element is measured. However, we can be faced with a system where that attribute of quantity is not measured because it is always used completely or globally as a contribution and not partially. Remember that this methodology does not consider an attribute measurable whose use is logical. Therefore, suitable in these cases is to define the element as unitary (it would be like always working with the continent instead of with the content inside it). However, if we defined the element as measurable individual, we could also model the problem, though not very efficiently.

- Case 2. Measurable Element with both measurable and logical decisions

This occurs when regarding an element, measurable individual or collective, decisions are made about its content and its continent, being the direct object to

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_7

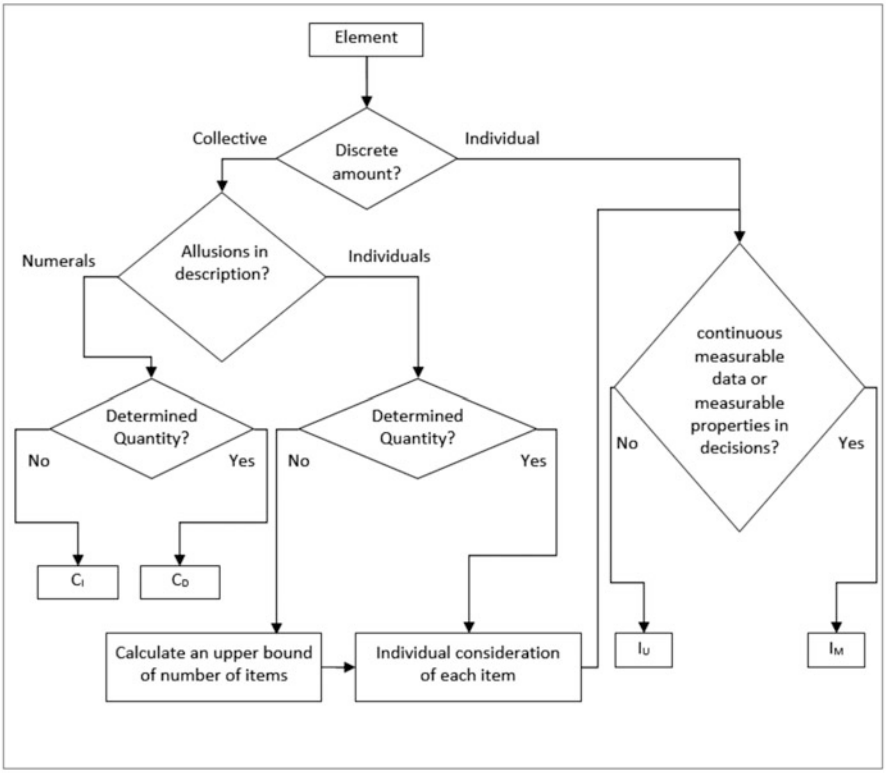

Element
Collective
Discrete amount?
Individual
Numerals
Allusions in description?
Individuals
continuous measurable data or measurable properties in decisions?
Determined Quantity?
No
Yes
Determined Quantity?
No
Yes
No
Yes
C_I
C_D
Calculate an upper bound of number of items
Individual consideration of each item
I_U
I_M

Fig. 7.1 Decision diagram of the nature of an element

the action. Decisions about its content are measurable decisions, while decisions about the global element are logical decisions. We will see how that measurable character can ensure that all decisions can be defined as measurable. We will also propose a configuration on these elements that separates the continent from the content, thus obtaining that the two types of decisions on the same element cannot be considered.

- Case 3. Individual elements with capacity to be grouped in collective elements

In the configuration of the elements of a system, we can find situations where we can have individual elements with the capacity to define themselves as collective, through the modification of the description or through the identification of subgroups of identical elements. In general, the use of collective elements instead of individual elements produces a model with a smaller size and greater efficiency. Therefore, it is good to look for that possibility if the problem does not arise in the first instance. The cases that we can consider are the following:

– Case 3.1. Redefinition of the system

This case occurrs when we have a set of individual elements with the same functionality in the system and identical with respect to their data and values, and we can avoid referring to them individually by redefining the description of the system. In this way we manage to group the elements as a single collective element, which will collect all the data of these elements plus an additional attribute of existence.

With the redefinition of the specifications, we will always convert the particular allusions to the items in references to numerals of a collective element. This process can be simple, with a small change in the description of the specifications or may involve a change of greater scope, which is achieved when you have a rounded knowledge of the problem and extensive experience in the field of operational research. Therefore, we distinguish the two cases:

Case 3.1.1. Redefinition of the system with simple changes
Case 3.1.2. Redefinition of the system with complex changes

– Case 3.2. Grouping into subsets

We have a set of individual elements with the same functionality in the system (same data definition), but their values are not identical in the totality of the elements, although you can define identical subsets. Again, whenever there is no individual reference to each item or those allusions can be eliminated, as in case 3.1, we can create a collective element for each subset.

– Case 3.3. Small changes in the data values

We have a set of individual elements with the same data definition, but the values of the data differ slightly so that the grouping of collective elements cannot be carried out. The solution is to assume small errors in the data through changes in the data values to allow the grouping into collective elements.

- Case 4. Items of indeterminate collective elements that need to be defined individually

This case differs from the previous ones in the sense that it does not express an improvement process but a process necessary for modelling the system. It occurs when the description of a system implies an element as an indeterminate collective, but the statement treats the possible items of the element individually. In that case it is necessary to transform each item of that collective element into an individual element. Since the quantity of the collective element is indeterminate, it will be necessary to calculate an upper bound of the number of items that it could have, and that dimension will be the number of individual elements created.

To illustrate all these cases, we will present each casuistry in sections throughout the chapter.

7.2 Individual Element Not Measurable Defined as Measurable

We are going to illustrate the double treatment of an individual element that has a continuous attribute that is not used partially: in a first version, the correct version, as a unitary element and the second as a measurable element by mistakenly considering this continuous attribute as measurable. It will be shown the suitability of the first version with respect to the second one.

Illustration 7.1

A factory owns a set of 120 sacks of rice grain. Each sack has its own quantity of kilos of rice. It is desired to assign the rice to two production sections where it is processed.

Each sack can only be assigned to a single section, and the full rice content is discharged.

Each section has a capacity. The time to process rice in each section is T1 sec/kilo and T2 sec/kilo. It is desired to maximize the production of rice within the 8 hours of the day.

The following elements are extracted from the statement:

- The factory (the system).
- The two sections.
- Regarding time, the system focuses on the optimization of a day. There are 8 h within the day. The day would be the individual element in which the activity is situated and that we will make explicit by assigning the attribute of 8 h. In the same way, this attribute could have been attributed to the system without declaring the day element. Moreover, the time of 8 h is applicable as the working time of each section, so we are going to consider it an attribute of the sections.
- Regarding rice sacks, they are distinguished from each other so that each sack must be considered as an individual element. The amount of rice in each sack is already discussed as a measurable attribute and yet the phrase "Each sack can only be assigned to a single section and the full rice content is discharged" indicates that the item is used in full in the system and therefore is not measurable.

This is going to be the element that distinguishes the two versions. In the first version, we will define the sacks as unitary elements, while in the second we will define them as measurable elements.

Version 1

Table of Elements (Table 7.1)

Decision Activities

Action: Assign rice sacks to sections (equivalent to processing sacks in sections).

Decision variables:

$\alpha_{ij} = 1$ if sack j is assigned to Section i; 0 otherwise. $i = 1,2$; $j = 1\ldots120$.

Table 7.1 Elements of Illustration 7.1 – Version 1

			Data				
Elements	Set	QN	Name	Param	Type	Belong	Value
Sections	$i = 1...2$	I_U	Capacity	K_i	C (kilos)	W	–
			Process time	T_i	C (sec/kilo)	W	–
			Working time	TT_i	C (hours)	W	8
Rice sacks	$j = 1...120$	I_U	Rice quantity	A_j	C (kilos)	W	–

Table 7.2 Selection rules of decision activity "Assign Sacks to sections"

Activity	Elements selecting	Selectable elements	Selection	Constraints
Assign	Sack $j = 1...120$	Sections	≤ 1	E1
	Section $i = 1,2$	Sacks	–	

Implicit Specifications

I1. Based on data:

– The capacity data of each section:

$$\forall i: \sum_{j=1}^{120} A_j \alpha_{ij} \leq K_i$$

The capacity consumption is carried out by the activity α_{ij}. The unit consumption corresponds to the quantity of rice that the sack has, since it is completely processed in the section.

– The working time attribute in the day affects each section and works as capacity attribute:

$$\forall i: \sum_{j=1}^{120} T_i A_j \alpha_{ij} \leq \mathrm{TT}_i$$

The unit consumption corresponds to the quantity of kilos of rice that the sack has, multiplied by the time it takes to use a kilogram in the section.

I2. Quantitative selection rules: Let us analyze the quantitative norms of activity α_{ij} (Table 7.2).

Each sack goes at the most to a section (E1, as appears explicitly). We are not required to use them all. No quantitative rule is established for each section.

I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

Explicit Specifications

E1. "Each bag can only be assigned to a single section."

Table 7.3 Elements of Illustration 7.1 – Version 2

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Sections	$i = 1\ldots2$	$\mathrm{I_U}$	Capacity	K_i	C (kilos)	W	–
			Process time	T_i	C (sec/kilo)	W	–
			Working time	TT_i	C (hours)	W	8
Rice sacks	$j = 1\ldots120$	$\mathrm{I_M}$	Rice quantity	A_j	C (kilos)	W	–

The selection rule already analyzed:

$$\forall j : \sum_{i=1}^{2} \alpha_{ij} \leq 1$$

Objective Criterion

Maximize processed kilo of rice:

$$\text{Max} \quad \sum_{i=1}^{2} \sum_{j=1}^{120} A_j \alpha_{ij}$$

Version 2

We started to consider the sacks of rice as measurable (Table 7.3).

Decision Activities

Action: Assign Rice sacks to Sections (equivalent to processing rice sacks in sections).

Decision Variables:

x_{ij} = Amount of rice of sack j assigned to Section i. $i = 1{,}2; j = 1\ldots120$.

Implicit Specifications

I1. Based on data:

– The capacity data of each section:

$$\forall i : \sum_{j=1}^{120} x_{ij} \leq K_i$$

– The working time attribute in the day affects each section and works as capacity attribute:

$$\forall i : \sum_{j=1}^{120} T_i x_{ij} \leq \mathrm{TT}_i$$

Unitary consumption corresponds to the time a kilo uses in the section.

– The availability of rice in each sack implicitly implies a capacity specification:

$$\forall j : \sum_{i=1}^{2} x_{ij} \leq A_j$$

I2. Quantitative selection rules: they do not exist.

I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

Explicit Specifications

E1: "Each sack of rice can only be assigned to a single section and the entire rice content is discharged."

It actually expresses two specifications:

"Each sack of rice can only be assigned to a single section."
Logical Proposition: $\forall j$: EITHER $x_{1j} > 0$ OR $x_{2j} > 0$.
"The full content of rice is discharged."
Logical Proposition: $\forall i, \forall j$: IF $x_{ij} > 0$ THEN $x_{ij} = A_i$.

This second specification means that the first one is not necessary. Assigning a sack of rice to one section assigns all the rice, and therefore no more rice can be assigned to any other section.

Objective Criterion

Maximize processed kilos of rice:

$$\text{Max} \quad \sum_{i=1}^{2} \sum_{j=1}^{120} x_{ij}$$

7.3 Measurable Element with Both Measurable and Logical Decisions

It is difficult to find systems where a measurable element in decision activities also has logical decisions. In many cases this situation is due to the fact that the optimization problem encompasses several problems that could be independent. The normal situation is that decisions are measurable and logical calculations are obtained from those decisions. Anyway, when this situation occurs, we will say that the logical decisions are made on the continent of the element and the measurable decisions on the content. Therefore, we will propose the separation of the continent and the content in the table of elements.

We present two illustrations. The first refers to a measurable individual element and the second to a collective one. In the first one, we are going to slightly modify the statement in Illustration 7.1 to assign a mandatory measurable attribute to rice sacks.

Illustration 7.2: Measurable Individual Element
A factory owns a set of 120 sacks of rice grain. Each sack has its own quantity of kilos of rice. It is desired to assign the rice to two production sections where it is

Table 7.4 Elements of Illustration 7.2

			Data				
Elements	Set	QN	Name	Param	Type	Belong	V.
Sections	$i = 1\ldots2$	$\mathrm{I_U}$	Capacity	K_i	C (kilo)	W	–
			Process time	T_i	C (sec/kilo)	S	–
			Working time	TT_i	C (hours)	W	8
Sacks	$j = 1\ldots120$	$\mathrm{I_U}$	Rice quantity	A_j	C (kilos)	S	–
Rice	–	$\mathrm{I_M}$		A_j; K_i; T_i			

processed. Each sack can only be assigned to a single section. Subsequently, the amount of rice to be processed from each sack is free.

Each section has a capacity. The time to process rice in each section is T1 sec/kilo and T2 sec/kilo. It is desired to maximize the production of rice in the 8 hours of the day.

The quantity of rice in each bag becomes a measurable attribute in the decision activities, since the amount of rice that is processed must be decided. In addition to deciding the quantity of rice that is processed, in the system there is a logical decision to assign sacks to the sections. That is, on the one hand sacks are assigned to sections, and on the other the amount of rice processed must be decided.

The table of elements would be the one described in Table 7.3. However, in the cases in which the possibility of using the element can be seen both in a measurable and logical way, it may be more convenient to configure the elements where we separate the content element from the continent element, making the continent unitary and the content measurable, as reflected in Table 7.4.

Sacks of rice become unitary. The measurable amount of rice is shared with the rice element, which acts as primary and supports the measurable nature.

As mentioned, the decision activities would be defined as:

Decision Activities

Action: Assign Sacks to Sections.
Decision variables: $\alpha_{ij} = 1$ if I assign sack j to section i; 0 otherwise.
Action: Process Rice from the Sacks.
Decision variables: $x_j =$ Amount of rice processed from sack j.

It is not necessary to include the participation of the sections in the processing activity, since we already have the activity of assigning sacks to sections, but their inclusion may favor the subsequent modelling of the specifications.

Action: Process Rice from Sacks in Sections.
Decision variables: $x_{ij} =$ Amount of rice from sack j processed in section i.

It could be that we had assigned sack j to section i and we would not have processed any rice from the sack. Of course, this fact is strange in itself. The logical thing would be to think that if you assign the sack to a section, it is because you are going to process rice from it in that section; otherwise, it would be normal not to assign it. If we situate ourselves in this last scenario, we would always include the sections as participants in the processing activity and the allocation of sacks to sections would cease to be a decision activity and become a logical calculation. With this, there would be no logical activity on the measurable element:

Action: Process rice from the sacks in sections.
Decision variables: x_{ij} = Amount of rice from sack j processed in section i.

Binary logical calculation: Assign sack j to section i.
Applied to: Each sack $j = 1\ldots120$; each section $i = 1,2$.
Variables: α_{ij}=1 if I assign sack j to section i; 0 otherwise.
Logical propositions:
$\forall j, i : \alpha_{ij} = 1$ IF AND ONLY IF $x_{ij} > 0$

We will now illustrate the same case but with respect to a collective element, which will have decisions about its content and about the continent.

Illustration 7.3: Collective element
A parts machining factory produces 15 different part models (i = 1...15). For manufacturing, it has 10 machines (j = 1...10). Each machine uses a time of T_{ij} minutes to produce a part of model i.

The weekly planning of the production of parts is P_i units of each model i, although in the system only the daily selection of 5 models is allowed. In the week there are 6 days of production and a capacity of 12 hours of production each day.

It is about assigning the production of parts to the machines, keeping in mind that in any machine you cannot produce more than two models of different parts on any given day.

The objective is to use the least possible time in the production of the parts.

The main elements in the problem are each part model, which are formed by a number of determined units to produce. Each part model would be a collective element because it consists of a set of identical items and without individual reference to each item in the system. That would be the content. The decisions in the system fall on the content, that is, on the number of parts that will be made each day in each machine, although in the description it can also be understood that there is a decision on which models to make each day, since there is a restriction on a maximum of five models per day. According to this, we should also reference the continent, which would be the model of each part. It would be a unitary individual element.

In spite of this, another valid perspective in the problem is to consider only decisions about the content. The decisions about the continent are converted into a logical calculation regarding the content (if I have produced parts of a model 1 day, I have selected that model that day). If I do not opt for the logical calculation option, I

Table 7.5 Elements of Illustration 7.3 – Implementation 7.1

			Data				
Elements	Set	QN	Name	Param	Type	Belong	Value
Models	$i = 1\ldots15$	I_U	Production	P_i	I	S	–
			Time	T_{ij}	C (min)	S	–
Parts	–	C_D		P_i			
Machines	$j = 1\ldots10$	I_U		T_{ij}			
Days	$t = 1\ldots6$	I_U	Available time	H_t	C (min)	W	12*60

Table 7.6 Elements of Illustration 7.3 – Implementation 7.2

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Part models	$i = 1\ldots15$	C_D	Production	P_i	I	W	–
			Time	T_{ij}	C (min)	S	–
Machines	$j = 1\ldots10$	I_U		T_{ij}			
Days	$t = 1\ldots6$	I_U	Available time	H_t	C (min)	W	12*60

would have to express the implicit relationship between the decisions as a specification to select a model and produce parts. Let us see the two implementations:

Implementation 7.1: Decisions on Content and the Continent (Table 7.5)

Decision Activities

Action: Select Models in Days.
Decision variables: $\alpha_{it} = 1$ if I select model i in day t; 0 otherwise.
Action: Produce Parts of Models in Machines in Days.
Decision variables:
x_{ijt} = Number of parts of model i produced in machine j in day t.

Implicit Specification

We indicate the relationship between both decision variables, regardless of whether there are other implicit specifications.

I3. Logical conditions between activities: Between α_{it} and x_{ijt}, the following relationship cannot be ignored:

$\forall i, j, t$: IF $\alpha_{it} = 0$ THEN $x_{ijt} = 0$.

Implementation 7.2: Decisions Made Only About the Content

In this second version, we do not make the continent explicit in the table of elements because the decisions fall on the content exclusively. For this reason, each part model is considered a collective element (Table 7.6).

Decision Activities

Action: Produce parts of models in machines in days.
Decision variables:
x_{ijt} = Number of parts of model *i* produced in machine *j* in day *t*.

Logical Calculation

Binary logical calculation: Select models in days.
Applied to: Each model $i = 1\ldots15$; Each day $t = 1\ldots6$.
Variables: α_{it}=1 if I select model *i* in day *t*; 0 otherwise.
Logical propositions:

$$\forall i, t : \alpha_{it} = 1 \quad \text{IF AND ONLY IF} \quad \sum_{j=1}^{10} x_{ijt} > 0$$

7.4 Individual Elements with Capacity to be Grouped in Collective Elements: Redefining the System with Simple Changes

To illustrate the change of individual elements to items of a collective element, let us take a look at a system of allocating distribution centers to supermarkets. The same system will be defined in three different ways by making small changes in the description. The three systems are equivalent.

Illustration 7.4: System of Allocating Distribution Centers
There is a supermarket company that has 10 locations (i = 1...10) to install a maximum of 3 product distribution centers. The installation cost in each location is established in $\$CI_i$. In a location one center is installed at most. And each installed distribution center must be in a single location.

The Company has 25 supermarkets (j = 1...25) to be supplied from the distribution centers. Each supermarket must be allocated to a distribution center for its supply. Each location has a K_i capacity that is expressed in the number of supermarkets that it can supply.

By legal requirements, if the company installs a center in location 3 and another in location 5, it cannot install any in location 6.

Objective: Minimize the costs of the problem taking into account that if the number of supermarkets allocated to a center exceeds 10, the center is penalized with a cost of $F.

Table 7.7 Elements of Illustration 7.4 – Version 1

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Locations	$i = 1\ldots10$	I_U	Cost	CI_i	I	W	–
			Capacity	K_i	I	W	–
Centers	$j = 1,2,3$	I_U	Penalty	P_j	C	W	F
Supermarkets	$k = 1\ldots25$	I_U					

Version 1

Table of Elements

According to the statement:

- The locations are individual and unitary.
- The supermarkets are identical, but they are referred to individually ("each supermarket must be assigned to a distribution center for its supply"). We consider them unitary.
- The centers are identical as well, but there is a particular reference to them in the statement to express a specification ("each distribution center must be in a single location"). Also, in the objective function, the centers are particularized. We consider the penalty as an attribute of the center, although being a constant of all centers we could have considered it as an attribute of the system (Table 7.7).

Decision Activities

There are two decision activities. On the one hand, install centers in locations and, on the other hand, assign supermarkets to centers.

Action: Install centers in locations.
Decision variables: $\alpha_{ij} = 1$ if center j is installed in location i; 0 otherwise.
Action: Allocate centers to supermarkets.
Decision variables: $\beta_{kj} = 1$ if supermarket k is allocated to center j; 0 otherwise.

Specifications

1. *Implicit Specifications*

I1. Based on data:

Although it could be considered that the specification of capacity is explained in the statement, we will consider it in the implicit section. There is a capacity attribute for the locations, expressing the maximum number of supermarkets that can be supplied from that location (k_i). The activities of the problem do not consume that resource. We need to calculate the supermarkets that are supplied from each location, because those variables will be the ones that consume that capacity. It is a logical calculation:

Binary logical calculation: Supermarket supplied from location.
Applied to: Each supermarket $k = 1\ldots25$ and each location $i = 1\ldots10$.
Variables:$\omega_{ki} = 1$ if supermarket k is supplied from location i.
Logical proposition: The supermarket k is supplied from the location i if the supermarket k is allocated to center j that is installed in location i:

$\forall k, i : \omega_{ki} = 1 \leftrightarrow \ (\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1) \text{ OR } (\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1)$
$\text{ OR } (\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1)$

Proposition modelling:

$\Rightarrow \text{Ref. f}_1 \Rightarrow$

$$\begin{aligned}&\Rightarrow \forall k, i : \text{IF } (\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1) \text{ OR } (\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1)\\ &\text{OR } (\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1) \text{ THEN } \omega_{ki} = 1\end{aligned} \tag{7.1}$$

$$\begin{aligned}&\Rightarrow \forall k, i : \text{IF } \omega_{ki} = 1 \text{ THEN } \ (\beta_{k1} = 1 \text{ AND } \ \alpha_{i1} = 1) \text{ OR } (\beta_{k2} = 1 \text{ AND } \alpha_{i2} = 1)\\ &\text{OR } (\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1)\end{aligned} \tag{7.2}$$

Model of (7.1):

$$\begin{aligned}&\qquad\qquad \Rightarrow \forall k, i : \text{IF } (\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1) \text{ THEN } \omega_{ki} = 1 \Rightarrow \text{Ref.f}_3 \Rightarrow\\ &\Rightarrow \text{Ref.f}_{40} \ \Rightarrow \forall k, i : \text{IF } (\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1) \text{ THEN } \omega_{ki} = 1 \Rightarrow \text{Ref.f}_3 \Rightarrow\\ &\qquad\qquad \Rightarrow \forall k, i : \text{IF } (\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1) \text{ THEN } \omega_{ki} = 1 \Rightarrow \text{Ref.f}_3 \Rightarrow\\ &\Rightarrow \text{IF NOT}(\omega_{ki} = 1) \text{ THEN NOT}(\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1) \Rightarrow \text{Ref.f}_7 \Rightarrow\\ &\Rightarrow \text{IF NOT}(\omega_{ki} = 1) \text{ THEN NOT}(\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1) \Rightarrow \text{Ref.f}_7 \Rightarrow\\ &\Rightarrow \text{IF NOT}(\omega_{ki} = 1) \text{ THEN NOT}(\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1) \Rightarrow \text{Ref.f}_7 \Rightarrow\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN NOT}(\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1) \Rightarrow \text{Ref.f}_{37} \Rightarrow\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN NOT}(\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1) \Rightarrow \text{Ref.f}_{37} \Rightarrow\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN NOT}(\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1) \Rightarrow \text{Ref.f}_{37} \Rightarrow\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN NOT}(\beta_{k1} + \ \alpha_{i1} \geq 2) \Rightarrow \text{Ref.f}_{11} \Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k1} + \ \ \alpha_{i1} \leq 1\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN NOT}(\beta_{k2} + \ \ \alpha_{i2} \geq 2) \Rightarrow \text{Ref.f}_{11} \Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k2} + \alpha_{i2} \leq 1\\ &\Rightarrow \text{IF1} - \omega_{ki} = 1\text{THEN NOT}(\beta_{k3} + \alpha_{i3} \geq 2) \Rightarrow \text{Ref.f}_{11} \Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k3} + \ \ \alpha_{i3} \leq 1\end{aligned}$$

$$\left|\begin{aligned}&\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k1} + \quad \alpha_{i1} \leq 1 \Rightarrow \text{Ref.f}_{14} \Rightarrow \beta_{k1} + \quad \alpha_{i1} \leq 1 + \omega_{ki}\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k2} + \quad \alpha_{i2} \leq 1 \Rightarrow \text{Ref.f}_{14} \Rightarrow \beta_{k2} + \quad \alpha_{i2} \leq 1 + \omega_{ki}\\ &\Rightarrow \text{IF } 1 - \omega_{ki} = 1 \text{ THEN } \beta_{k3} + \quad \alpha_{i3} \leq 1 \Rightarrow \text{Ref.f}_{14} \Rightarrow \beta_{k3} + \quad \alpha_{i3} \leq 1 + \omega_{ki}\end{aligned}\right| \tag{7.3}$$

Model of (7.2):

In this calculation, we just need the case in which we use $\omega_{ki} = 1$, case (7.1). We can give freedom to the system when conditions to use $\omega_{ki} = 1$ do not coincide. That means that it is not necessary to impose (7.2). Anyway, if we want ω_{ki} to be worth 0 when the proposal is not fulfilled, for a formal question, we model the case (7.2):

$$\Rightarrow \forall k,i : \text{IF } \omega_{ki} = 1 \text{ THEN } \quad (\beta_{k1} = 1 \quad \text{AND} \quad \alpha_{i1} = 1)$$
$$\text{OR } (\beta_{k2} = 1 \quad \text{AND} \quad \alpha_{i2} = 1) \qquad \text{OR } (\beta_{k3} = 1 \quad \text{AND} \quad \alpha_{i3} = 1)$$

$$\Rightarrow \text{Ref } f_{37} \Rightarrow \forall k,i : \text{IF } \omega_{ki} = 1 \text{ THEN } (\beta_{k1} + \alpha_{i1} \geq 2) \text{ OR } (\beta_{k2} + \alpha_{i2} \geq 2) \text{ OR } (\beta_{k3} + \alpha_{i3} \geq 2)$$

$\Rightarrow$ Ref f_{35} $\Rightarrow$

$$\left.\begin{array}{l}
\Rightarrow \forall k,i : \lambda_{ki1} = 1 \text{ IF AND ONLY IF } \quad (\beta_{k1} + \quad \alpha_{i1} \geq 2) \\
\Rightarrow \text{Ref } f_{25} \Rightarrow \beta_{k1} + \quad \alpha_{i1} \geq 2\lambda_{ki1}; \beta_{k1} + \quad \alpha_{i1} \leq 1 + \lambda_{ki1} \\
\Rightarrow \forall k,i : \lambda_{ki2} = 1 \text{ IF AND ONLY IF } \quad (\beta_{k2} + \quad \alpha_{i2} \geq 2) \\
\Rightarrow \text{Ref } f_{25} \Rightarrow \beta_{k2} + \quad \alpha_{i2} \geq 2\lambda_{ki2}; \beta_{k2} + \quad \alpha_{i2} \leq 1 + \lambda_{ki2} \\
\Rightarrow \forall k,i : \lambda_{ki3} = 1 \text{ IF AND ONLY IF } \quad (\beta_{k3} + \quad \alpha_{i3} \geq 2) \\
\Rightarrow \text{Ref } f_{25} \Rightarrow \beta_{k3} + \quad \alpha_{i3} \geq 2\lambda_{ki3}; \beta_{k3} + \quad \alpha_{i3} \leq 1 + \lambda_{ki3}
\end{array}\right| \qquad (7.4)$$

$$\Rightarrow \forall\ k,\ i : \text{IF } \omega_{ki} = 1 \text{ THEN } \quad \lambda_{ki1} = 1 \text{ OR } \lambda_{ki2} = 1 \text{ OR } \lambda_{ki3} = 1$$
$$\Rightarrow \text{Ref } f_{34} \Rightarrow \ \forall\ k,\ i : \text{IF } \omega_{ki} = 1 \text{ THEN } \quad \lambda_{ki1} + \lambda_{ki2} + \lambda_{ki3} \geq 1$$

$$\Rightarrow \text{Ref } f_{15} \Rightarrow \forall k,i : \quad \lambda_{ki1} + \lambda_{ki2} + \lambda_{ki3} \geq \omega_{ki} \qquad (7.5)$$

The specification would be:

$$\forall i : \sum_{k=1}^{25} \omega_{ki} \leq K_i \qquad (7.6)$$

I2. Quantitative selection rules:

The two sets of binary variables have associated quantitative selection specifications. Let’s analyze each activity and its participating elements (Table 7.8).

The three specifications are explicitly reflected in the statement.

I3. Logical conditions between activities:

Table 7.8 Selection rules in Illustration 7.4 – Version 1

Activity	Elements selecting	Selectable elements	Selection	Constraints
Install	Locations $i = 1\ldots10$	Centers	≤ 1	E1
	Centers $j = 1\ldots3$	Locations	≤ 1	E2
Allocate	Supermarket $k = 1\ldots25$	Centers	$= 1$	E3
	Center $j = 1\ldots3$	Supermarkets	No imposed selection specification	–

The two decision activities have an obvious logical relationship: in order for a center to supply supermarkets, it must have been installed. We express the logical proposition:

"IF a center has been installed THEN it can supply supermarkets."
As the proposition expresses possibility, we define it negatively (see Sect. 6.8.6):
"IF a center has not been installed THEN It cannot supply supermarkets."

Mathematically:

The specification falls on any center that has not been installed in any location and will not serve any supermarket of the existing ones. Therefore, it falls on the three sets of elements:

$$\forall j, i, k : \text{IF } \alpha_{ij} = 0 \quad \text{THEN} \quad \beta_{kj} = 0$$

But it can also be expressed by reducing the number of propositions in the following way (we change the way of expressing the specification using numerals):

"IF the number of locations in which a center is installed is null THEN the number of supermarkets to which it will supply will be null":

$$\forall j : \text{IF } \sum_{i=1}^{10} \alpha_{ij} = 0 \quad \text{THEN} \quad \sum_{k=1}^{25} \beta_{kj} = 0$$

Expression $\sum_{i=1}^{10} \alpha_{ij}$ is binary, thanks to the specification that a center cannot be installed in more than one location. $\sum_{k=1}^{25} \beta_{kj}$is an integer expression.

We express the specification as:

$$\Rightarrow \text{Ref.f}_7 \Rightarrow \forall j : \text{IF } 1 - \sum_{i=1}^{10} \alpha_{ij} = 1 \quad \text{THEN} \quad \sum_{k=1}^{25} \beta_{kj} = 0 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{\text{LB}} \Rightarrow \forall j : \text{IF } 1 - \sum_{i=1}^{10} \alpha_{ij} = 1 \quad \text{THEN} \quad \sum_{k=1}^{25} \beta_{kj} \leq 0 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow \forall j: \quad \sum_{k=1}^{25} \beta_{kj} \leq 25 \sum_{i=1}^{10} \alpha_{ij} \tag{7.7}$$

I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

Explicit Specifications

E1. "In a location one center is installed at the most."

It is a quantitative selection norm of upper bound for $\forall i$ in α_{ij}:

$$\forall i: \sum_{j=1}^{3} \alpha_{ij} \leq 1 \tag{7.8}$$

E2. "Each installed distribution center must be in a single location."

It is also a quantitative selection rule for $\forall j$ in α_{ij}. It would be an upper bound since there is no obligation to install each center.

$$\forall j: \sum_{i=1}^{10} \alpha_{ij} \leq 1 \tag{7.9}$$

E3. "Each supermarket must be allocated to a distribution center for its supply."

Quantitative selection rule for $\forall k$ in the variables β_{kj}

$$\forall k: \sum_{j=1}^{3} \beta_{kj} = 1 \tag{7.10}$$

E4. "If the company installs a center in location 3 and another in location 5, it cannot install any in location 6."

It is a compound logical proposition. The simple propositions of which it is formed are quantitative selection specifications:

- Install a center in location 3:

  $\sum_{j=1}^{3} \alpha_{3j} = 1$
- Install a center in location 5:

  $\sum_{j=1}^{3} \alpha_{5j} = 1$

- Unable to install a center in location 6:

  $\sum_{j=1}^{3} \alpha_{6j} = 0$

  Compound logical proposition:

$$\text{IF } \sum_{j=1}^{3} \alpha_{3j} = 1 \text{ AND } \sum_{j=1}^{3} \alpha_{5j} = 1 \text{ THEN } \sum_{j=1}^{3} \alpha_{6j} = 1$$

where all simple propositions are binary, by the definition of the number of centers in a location. The modelling would be:

$$\Rightarrow \text{Ref.f}_{37} \Rightarrow \text{IF } \sum_{j=1}^{3} \alpha_{3j} + \sum_{j=1}^{3} \alpha_{5j} \geq 2 \text{ THEN } \sum_{j=1}^{3} \alpha_{6j} = 0 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{3} \Rightarrow \text{IF } \sum_{j=1}^{3} \alpha_{6j} = 1 \text{ THEN } \sum_{j=1}^{3} \alpha_{3j} + \sum_{j=1}^{3} \alpha_{5j} < 2 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{4} \Rightarrow \text{IF } \sum_{j=1}^{3} \alpha_{6j} = 1 \text{ THEN } \sum_{j=1}^{3} \alpha_{3j} + \sum_{j=1}^{3} \alpha_{5j} \leq 1 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow \sum_{j=1}^{3} \alpha_{3j} + \sum_{j=1}^{3} \alpha_{5j} \leq 1 + (2-1)\left(1 - \sum_{j=1}^{3} \alpha_{6j}\right) \Rightarrow$$

$$\Rightarrow \sum_{j=1}^{3} \alpha_{3j} + \sum_{j=1}^{3} \alpha_{5j} \leq 2 - \sum_{j=1}^{3} \alpha_{6j} \tag{7.11}$$

Objective Criterion

Minimize Costs

Costs = Installation costs in locations + Additional cost to allocate more than 10 supermarkets to any center

- Installation costs in locations $= \sum_{i=1}^{10} \sum_{j=1}^{3} \text{CI}_i \alpha_{ij}$.
- Additional Cost: it requires a logical calculation to find out if more than 10 supermarkets have been assigned in each center. The calculation falls on each center:

Binary logical calculation: Center with more than 10 supermarkets.
Applied to: Centers $j = 1\ldots3$.
Variables:

$$\delta_j = \begin{cases} 1 \text{ if center } j \text{ has more than 10 supermarkets} \\ 0 \text{ otherwise} \end{cases}$$

Logical propositions:

$$\forall j : \text{IF } \sum_{k=1}^{25} \beta_{kj} > 10 \text{ THEN } \delta_j = 1$$

[Ref. $S_{V:}$ It implies a cost, so I can stop defining the value $\delta_j = 0$].

Proposition modelling:

$$\Rightarrow \text{Ref.f}_3 \Rightarrow \forall j: \text{IF } \delta_j = 0 \ \text{ THEN } \ \sum_{k=1}^{25} \beta_{kj} \leq 10$$

$$\Rightarrow \text{Ref.f}_7 \Rightarrow \forall j: \text{IF } 1 - \delta_j = 1 \ \text{ THEN } \ \sum_{k=1}^{25} \beta_{kj} \leq 10$$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow \forall j: \ \sum_{k=1}^{25} \beta_{kj} \leq 10 + (25 - 10)\big(1 - \big(1 - \delta_j\big)\big)$$

$$\Rightarrow \forall j: \ \sum_{k=1}^{25} \beta_{kj} \leq 10 + 15\delta_j \tag{7.12}$$

O.F.: Min $\sum_{i=1}^{10} \sum_{j=1}^{3} \text{CI}_i \alpha_{ij} + \sum_{j=1}^{3} F\delta_j$

Version 2

By analyzing the table of elements of Version 1, where the centers are identical elements, we will try to eliminate the specifications that allude to each center in an individual way, to be able to treat it as a collective element without changing the system. Let us take a look at the references to the centers in the statement:

- "Install a maximum of 3 product distribution centers": Here, the centers are referred to with a numeral, so there is no change at all.
- "In a location one center is installed at the most": Here we can also interpret a center as a numeral. It refers to the number of centers that can be installed in one location.
- "Each installed distribution center must be in a single location": Here you make an individual reference about each center. Therefore, it is necessary to modify this reference. What we are going to do is eliminate that specification. Through specifying that the number of centers installed in locations cannot be more than 3 and that in one location a center is installed at most, we assume that the number of centers installed will correspond to the sum of the centers installed in the locations.
- "If the company installs a center in location 3 and another in location 5, it cannot install any in location 6": here again the reference to the center can be considered as a numeral. Written in another way: "if the company installs 1 center in location 3 and 1 center in location 5, the number of centers in 6 must be 0."
- "If the number of supermarkets allocated to a center exceeds 10, the center is penalized . . .."

Here, each center is referred to in a particular way, since the affirmation is made for each center. This statement makes use of one of the decision activities: allocate supermarkets to centers.

Table 7.9 Elements of Illustration 7.4 – Version 2

Elements	Set	QN	Data				
			Name	Param	Type	Belong	Value
Locations	$i = 1\ldots10$	I_U	Cost	CI_i	I	W	8
			Capacity	K_i	I	W	–
			Penalty	P_i	C	W	F
Centers	–	C_I	Max number	N	I	W	3
Supermarkets	$k = 1\ldots25$	I_U					

The only way to eliminate this allusion is to modify that decision activity. This way of describing systems is quite common when there are collective elements whose measurement on an element is at most one. This element ends up taking the place of the individual element with which it is related, although it is the latter on which the activities really fall. The idea is to embed the role of the centers in the locations. The centers act only as an indicator that the location can supply supermarkets. If we have installed a center in the location we can supply, otherwise we cannot. We are going to make the locations assume the role of center regarding the allocation of supermarkets. In that way, we would avoid the particular allusion to centers. The statement will then be as follows:

“There is a supermarket company that has 10 locations ($i = 1\ldots10$) to install a maximum of 3 product distribution centers. The installation cost in each location is established in $\$CI_i$. In a location one center is installed at the most. ~~And each installed distribution center must be in a single location.~~

The Company has 25 supermarkets ($j = 1\ldots25$) to be supplied from the locations with distribution centers. Each supermarket must be allocated to a location with a distribution center for its supply. Each location has a K_i capacity that is expressed in the number of supermarkets that it can supply.

By legal requirements, if the company installs a center in location 3 and another in location 5, it cannot install any in location 6.

Objective: Minimize the costs of the problem taking into account that if the number of supermarkets allocated to a ~~center~~ location exceeds 10, the ~~center~~ location is penalized with a cost of \$F.”

Table of Elements (Table 7.9)

Decision Activities

Action: Install centers in locations.

Decision variables: x_i = number of centers installed in location *i.*

Action: Allocate supermarkets to locations.

Decision variables:

$$\beta_{ki} = 1 \text{ if supermarket } k \text{ is allocated to location } i\text{; 0 otherwise.}$$

Specifications

We are going to review the specifications indicated in Version 1.

Table 7.10 Selection rules in Illustration 7.4 – Version 2

Activity	Elements selecting	Selectable elements	Selection	Constraints
Allocate	Supermarket $k = 1\ldots25$	Locations	$= 1$	E3
	Location $i = 1\ldots10$	Supermarkets	$\leq K_i$	I1

1. Implicit Specifications

I1. Based on data: In this version the modelling associated with the capacity of the locations is simplified, since it is the activity represented in β_{ki} that consumes that capacity. The logical calculation ω_{ki} defined in Version 1 is equivalent to β_{ki}

$$\forall i: \sum_{k=1}^{25} \beta_{ki} \leq K_i \tag{7.13}$$

I2. Quantitative selection rules:

Those associated with centers with locations and centers with supermarkets disappear, and supermarkets with locations appear (Table 7.10).

They are specifications that are expressed explicitly (E3) or already defined as capacity specification (I1).

I3. Logical conditions between activities: The two decision activities maintain the same logical relationship as in Version 1, but it is formulated with the following expression:

IF a center has been installed in a location, THEN that location “can” supply supermarkets.

In a negative way:

IF the number of centers installed in a location is zero, THEN that location cannot supply any supermarket.

Mathematically:

The specification falls on any location and any supermarket: $\forall i, k$: IF $x_i = 0$ THEN $\beta_{ki} = 0$.

And again, the number of propositions could be reduced if we join all the propositions for i:

$$\forall i: \text{IF } x_i = 0 \quad \text{THEN} \quad \sum_{k=1}^{25} \beta_{ki} = 0$$

Modelling: $\Rightarrow$ Ref. $f_7 \Rightarrow \forall i$: IF $1 - x_i = 1$ THEN $\sum_{k=1}^{25} \beta_{ki} = 0 \Rightarrow$

$\Rightarrow$ Ref.$f_{LB} \Rightarrow \forall i$: IF $1 - x_i = 1$ THEN $\sum_{k=1}^{25} \beta_{ki} \leq 0$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow \forall i: \sum_{k=1}^{25} \beta_{ki} \leq 25x_i \tag{7.14}$$

I4. Bounds of discrete measurable activities: explicitly in E1.
I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

E1. "In a location one center is installed at most."

Now it is an upper bound of a measurable activity.

$$\forall i : x_i \leq 1 \tag{7.15}$$

E2. "Install a maximum of 3 product distribution centers."

$$\sum_{i=1}^{10} x_i \leq 3 \tag{7.16}$$

E3. "Each supermarket must be allocated to a location with a distribution center for its supply."

It is a quantitative norm of selection for the β_{ki} variables.

$$\forall k : \sum_{i=1}^{10} \beta_{ki} = 1 \tag{7.17}$$

The rule states that the location you choose must have a center installed, that is, $x_i = 1$. As already explained in Sect. 6.3, this type of condition is excluded from the quantitative rule, where I will consider all the options for the standard. It must be in an additional specification where the selection of an element that does not meet that condition is limited or prevented. That is already reflected and modelled in the implicit specification number 13.

E4. "If the company installs a center in location 3 and another in location 5, it cannot install any in location 6."

Now the simple propositions that make up this compound proposition express assignment specifications:

Compound logical proposition: IF $x_3 = 1$ AND $x_5 = 1$ THEN $x_6 = 0$.

All the functions of simple propositions can be considered as binary, since $\forall i : x_i \leq 1$:

$\Rightarrow$Ref. $\text{f}_{37} \Rightarrow$ IF $x_3 + x_5 \geq 2$ THEN $x_6 = 0$

$\Rightarrow$Ref. $f_3 \Rightarrow$ IF $x_6 = 1$ THEN $x_3 + x_5 < 2$
$\Rightarrow$Ref. $f_4 \Rightarrow$ IF $x_6 = 1$ THEN $x_3 + x_5 \leq 1$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow x_3 + x_5 \leq 2 - x_6 \tag{7.18}$$

Objective Criterion

Minimize Costs
Costs = Installation costs in locations + Additional cost to allocate more than 10 supermarkets to any location.

Installation costs of centers in locations $= \sum_{i=1}^{10} \text{CI}_i x_i$.

Additional cost: Logical calculation to find out if more than 10 supermarkets have been assigned in each location:

Binary logical calculation: Location with more than 10 supermarkets.
Applied to: Locations $i = 1 \ldots 10$.
Variables:

$$\delta_i = \begin{cases} 1 \text{ if location } i \text{ has more than 10 supermarkets} \\ 0 \text{ otherwise} \end{cases}$$

Logical proposition: $\forall i$: IF $\sum_{k=1}^{25} \beta_{ki} > 10$ THEN $\delta_i = 1$ [Ref. S_V].

Proposition Modelling: $\Rightarrow$ Ref. $f_3 \Rightarrow \forall i$: IF $\delta_i = 0$ THEN $\sum_{k=1}^{25} \beta_{ki} \leq 10$.

$\Rightarrow$ Ref.$f_7 \Rightarrow \forall i$: IF $1 - \delta_i = 1$ THEN $\sum_{k=1}^{25} \beta_{ki} \leq 10$

$$\Rightarrow \text{Ref.f}_{14} \Rightarrow \forall i: \sum_{k=1}^{25} \beta_{ki} \leq 10 + 15\delta_i \tag{7.19}$$

O.F: Min $\sum_{i=1}^{10} \text{CI}_i x_i + F\delta_i$

Version 3

In this third version, we are going to consider supermarkets as a collective element. As we have observed in the tables of elements of the previous versions, the 25 supermarkets are identical. Therefore, we must try to modify the description of the system where they are alluded to in a particular way, without changing the system.

To do this, the allocation of each supermarket will be made globally, requiring 25 supermarkets to be assigned to the locations with centers. The statement would now look like this:

Table 7.11 Elements of Illustration 7.4 – Version 3

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Locations	$i = 1\ldots10$	I_U	Cost	CI_i	I	W	8
			Capacity	K_i	I	S	–
			Penalty	P_i	C	W	F
Centers	–	C_I	Max number	N	I	W	3
Supermarkets	–	C_D	Quantity	NS	I	W	25
				K_i			

"There is a supermarket company that has 10 locations ($i = 1\ldots10$) to install a maximum of 3 product distribution centers. The installation cost in each location is established in $\$CI_i$. In a location one center is installed at the most.

The Company has 25 supermarkets ($j = 1\ldots25$) to be supplied from the locations with distribution centers (the Company must assign 25 supermarkets to the locations with installed centers). ~~Each supermarket must be allocated to a location with distribution center for its supply.~~ Each location has a K_i capacity that is expressed in the number of supermarkets that it can supply.

By legal requirements, if the company installs a center in location 3 and another in location 5, it cannot install any in location 6.

Objective: Minimize the costs of the problem taking into account that if the number of supermarkets allocated to a location exceeds 10, the location is penalized with a cost of $F."

The configuration of the model would now be as follows:

Table of Elements (Table 7.11)

The capacity attribute of the locations is shared with the supermarket element, since it refers to the number of supermarkets that can supply.

Decision Activities
Action: Install centers in locations.
Decision variables: x_i = number of centers installed in location *i.*

Action: Allocate supermarkets to locations.
Decision variables: y_i = Number of supermarkets allocated to location *i*.
Specifications

1. Implicit Specification.

I1. Based on data: In this version, the capacity consumption activity of the locations is y_i:

$$\forall i: \quad y_i \leq K_i \tag{7.20}$$

Supermarket inventories are explicitly specified (E3) in the statement.

I2. Quantitative selection rules: they do not exist.

I3. Logical conditions between activities: Between x_i and y_i, there is the same logical condition as in previous versions: If you do not install any center in a location, you cannot supply supermarkets:

The specification falls on any location:

$\forall i : \text{IF } x_i = 0 \quad \text{THEN} \quad y_i = 0$
Modelling: $\Rightarrow$ Ref. $\text{f}_{\text{LB}} \Rightarrow \forall i : \text{IF } x_i \le 0 \quad \text{THEN} \quad y_i \le 0$

$$
\begin{array}{llll}
 & & \forall i : x_i \le 0 + (1-0)(1-\omega_i) & \\
\Rightarrow \text{Ref.f}_{17;}\,\text{f}_{20} & \Rightarrow & \forall i : x_i \ge (0+1)(1-\omega_i) + 0\omega_i & \\
 & & \forall i : y_i \le 0 + (K_i - 0)(1-\omega_i) & \\
\forall i : x_i \le 1-\omega_i & & & \\
\Rightarrow \forall i : x_i \ge 1-\omega_i & \Rightarrow & \begin{array}{l} \forall i : x_i = 1-\omega_i \\ \forall i : y_i \le K_i(1-\omega_i) \end{array} & \left. \Rightarrow \forall i : y_i \le K_i x_i \right| \\
\forall i : y_i \le K_i(1-\omega_i) & & &
\end{array} \tag{7.21}
$$

I4. Bounds of discrete measurable activities: explicitly in I1 and E1.

I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

E1. "In a location one center is installed at the most."

Identical to the one expressed in Version 2.

$$\forall i : x_i \le 1 \tag{7.22}$$

We can consider x_i as binary variable both here as in version 2.

E2. "Install a maximum of 3 product distribution centers."

Identical to the one expressed in Version 2.

$$\sum_{i=1}^{10} x_i \le 3 \tag{7.23}$$

E3. "The Company must assign 25 supermarkets to the locations with installed centers."

In this version, what was a quantitative norm of selection becomes an assignment or equilibrium specification.

$$\sum_{i=1}^{10} y_i = 25 \tag{7.24}$$

E4. "If the company installs a center in location 3 and another in location 5, it cannot install any in location 6"

Same as in Version 2.

$$x_3 + x_5 \leq 2 - x_6 \tag{7.25}$$

Objective Criterion

Regarding Version 2, the defined logical calculation varies slightly:

Binary logical calculation: Location with more than 10 supermarkets
Applied to: Locations $i = 1\ldots10$
Variables:

$$\delta_i = \begin{cases} 1 \text{ if location } i \text{ has more than 10 supermarkets} \\ 0 \text{ otherwise} \end{cases}$$

Logical proposition: $\forall i$: IF $y_i > 10$ THEN $\delta_i = 1$ [Ref. S_V]
Proposition Modelling:
$\Rightarrow$Ref. $f_3 \Rightarrow \forall i$: IF $\delta_i = 0$ THEN $y_i \leq 10$
$\Rightarrow$Ref. $f_7 \Rightarrow \forall i$: IF $1 - \delta_i = 1$ THEN $y_i \leq 10$
$\Rightarrow$Ref.$f_{14} \Rightarrow \forall i$: $y_i \leq 10 + 15\delta_i$ (7.26)

O.F.: Min $\sum_{i=1}^{10} (\mathrm{CI}_i x_i + F\delta_i)$

The following table summarizes the analysis of the three versions regarding their size (Table 7.12).

Version 1 Constraints: (7.3); (7.6) to (7.12).
Version 2 Constraints: (7.13) to (7.19) except (7.15).
Version 3 Constraints: (7.20) to (7.26) except (7.22).
(7.15) and (7.22) do not count when considering x_i as binary.

It can be observed that the optimal size is to make both centers and supermarkets collectives. Making collectives only the centers, the number of variables is slightly reduced. This is because the number of centers is not significant, only 3, which does not lead to a considerable decrease in size.

Table 7.12 Comparison of version sizes

	Version 1	Version 2	Version 3
Number of binary variables	358	270	10
Number of integer variables	0	0	10
Number of constraints	805	57	53

7.5 Individual Elements with Capacity to Be Grouped in Collective Elements: Redefining the System Description with Complex Changes

When a system can be defined as an alternative to an initial description, with significant differences, we are facing systems that require a broad knowledge in the field of operational research and experience in the area to which the problem corresponds, to allow us to develop the specifications differently.

In this illustration, we are going to deal with an interval scheduling problem, the fixed jobs scheduling problem (Arkin and Silverberg 1987). In the first description, the problem specifications prevent the union of items as a collective element, despite being identical. However, through understanding the problem, we can make a substantial change in the specifications to stop referring to each item in particular.

Illustration 7.5: Fixed Job Scheduling Problem

There is a set of n jobs characterized by a starting time, a duration, and a weight. There is also a set of m machines to process the jobs. It is about maximizing the total weight of the processed jobs taking into account that a machine cannot process two jobs that overlap in time. A job, if processed, is processed by a single machine.

Unitary Version

After reading the problem, we can define jobs and machines as elements. Each job has three data of own values, starting time, duration, and weight. Therefore, we consider them individual and also unitary because none of the data are measurable. When processing all the jobs in a single machine, we cannot divide the elements, so we work with it completely.

On the other hand, the machines do not present any data, so they can be considered identical; nor do they possess any property that must be measured in the decision activities. Therefore, we consider each machine as a unitary element a priori or the *m* machines as items of a collective element in the system. This will depend on the specifications of the problem. When analyzing the text, we can see that there is a specification that states that “a machine cannot process two jobs that overlap in time.” When we say “a machine,” since we use the indeterminate article, we refer to any of the machines in a particular way. We could have rewritten the text using “any”: any machine cannot process two jobs that overlap in time. Therefore, machines are referred to in a particular way, and it is necessary to consider each machine as unitary.

Table of Elements (Table 7.13)

Table 7.13 Elements of Illustration 7.5 – Individual Version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Jobs	$i = 1...n$	I_U	Starting time	s_i	C	W	–
			Duration	d_i	C	W	–
			Weight	p_i	C	W	–
Machines	$j = 1...m$	I_U					

Table 7.14 Selection rules in Illustration 7.5

Activity	Elements selecting	Selectable elements	Selection	Constraints
Process	Jobs $i = 1...n$	Machines	≤ 1	E1
	Machines $j = 1...m$	Jobs	–	

Decision Activities

Action: Process jobs in machines.
Decision variables: $\alpha_{ij} = 1$ if I process job i in machine j; 0 otherwise.

Specifications

1. Implicit Specifications

I1. Based on data: they do not exist.
I2. Quantitative selection rules: the only rule for the variables α_{ij} is described explicitly: A job, if processed, is processed by a single machine (Table 7.14).

I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

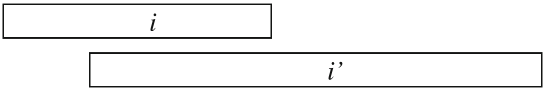
From the statement the following specifications are extracted:

E1. "A job, if processed, is processed by a single machine."

$$\forall i : \quad \sum_{j=1}^{m} \alpha_{ij} \leq 1 \tag{7.27}$$

E2. "A machine cannot process two jobs that overlap in time."

Fig. 7.2 Overlapping jobs



A machine cannot process any pair of jobs that overlap in time. It is a logical proposition with the operator negation that falls on each machine and on each pair of jobs that overlap in time.

The pairs of overlapping jobs can be calculated before the modelling of the specification, since it is information obtained from the data of the jobs. Graphically, two jobs i and i' are overlapped if they are processed together at some point in time (Fig. 7.2).

Mathematically, the overlap occurs when the values of the start and duration data of i and i' comply with:

$$\forall i, i'/i \neq i' : \; s_i \leq s_{i'} \;\; \& \;\; s_i + d_i > s_{i'}$$

The specification would then be as follows:

$$\forall j, \forall i, i'/i \neq i', s_i \leq s_{i'} \;\; \& \;\; s_i + d_i > s_{i'} : \mathrm{NOT}\big(\alpha_{ij} = 1 \quad \mathrm{AND} \;\; \alpha_{i'j} = 1\big)$$

Modelling:

$$\begin{aligned} &\Rightarrow \mathrm{Ref.f}_{37} \Rightarrow \forall j, \forall i, i'/i \neq i', s_i \leq s_{i'} \;\; \& \;\; s_i + d_i > s_{i'} : \mathrm{NOT}\big(\alpha_{ij} \;\; + \;\; \alpha_{i'j} \geq 2\big) \\ &\Rightarrow \mathrm{Ref.f}_{11} \Rightarrow \forall j, \forall i, i'/i \neq i', s_i \leq s_{i'} \;\; \& \;\; s_i + d_i > s_{i'} : \alpha_{ij} \;\; + \;\; \alpha_{i'j} \leq 1 \end{aligned} \tag{7.28}$$

Objective Criterion

$$\mathrm{Max} \;\; \sum_{i=1}^{n} \sum_{j=1}^{m} p_i \alpha_{ij}$$

Collective Version

Let’s modify the statement accordingly: If the modeller has some experience in this kind of problem, they will know that the overlap specification is equivalent to saying that, at any moment of the horizon in which the jobs are processed, the number of jobs that are processed must not exceed the number of existing machines (Kroon et al. 1995).

Let us look at the following example:

Suppose there is a system with six jobs that are represented in a time diagram (Fig. 7.3). The system has two machines.

If in any of the moments that make up the plan we say that no more jobs are processed than existing machines, any solution can be assigned to the machines afterwards without any overlapping conflict, because every time I start a job I will have the security of knowing that there is a machine free, since if it were not so, the

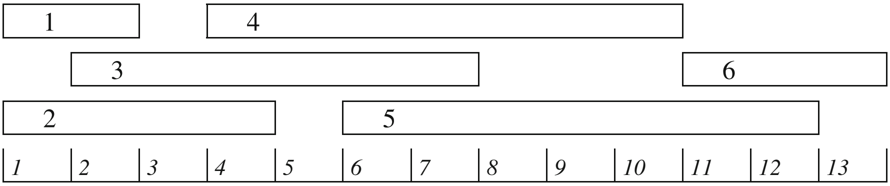


Fig. 7.3 Example

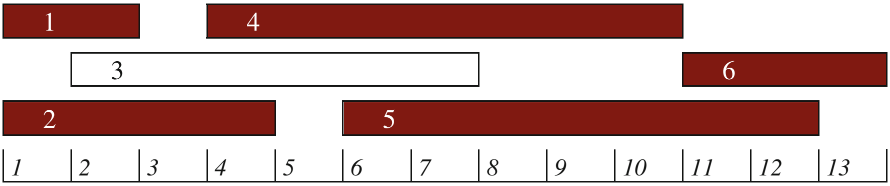


Fig. 7.4 A solution to the problem

selection would not comply with the specification. For our example, a possible solution would be to select the jobs 1,2,4,5,6, discarding job 3 (Fig. 7.4).

Moreover, it is not necessary to impose this specification at any time in the horizon, which would require defining all the instants as an element, but only imposing the overlap specification at times when new overlaps can occur, which are only the starting time of each job.

The solution for this version of the problem will provide the selected jobs, but not the specific assignment of which machine processes each job. This task should be obtained after the resolution, through a simple assignment procedure.

On the other hand, the second specification of the previous version claimed that a job is processed by a single machine at the most. This specification ceases to exist in this version, since the processing of the job in the machines is obtained afterwards and it is in the assignment procedure where that characteristic is imposed.

For all this, the problem would have been stated and modelled in the following way in the collective version:

There is a set of n jobs characterized by a starting time, a duration, and a weight. There is also a set of m machines to process the jobs. For every starting time of a job, the number of jobs that are being processed must not exceed the number of existing machines. It is about maximizing the total weight of the processed jobs ~~taking into account that a machine cannot process two jobs that overlap in time. A job, if processed, is processed by a single machine~~.

Table of Elements (Table 7.15)

Decision Activities

Action: Process jobs in machines.

Table 7.15 Elements of Illustration 7.5 – Collective Version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Jobs	$i = 1\ldots50$	$\mathrm{I_U}$	Starting time	s_i	C	W	–
			Duration	d_i	C	W	–
			Weight	p_i	C	W	–
Machines	–	$\mathrm{C_D}$	Quantity	m	I	W	–

Decision variables: $\alpha_i = 1$ if I process job i in the machines; 0 otherwise.

The jobs are the direct complement of the action. The machines are implicit in the processing and could be excluded from the definition.

Specifications

1. Implicit Specifications

I1. Based on data: they do not exist.
I2. Quantitative selection rules: they do not exist.
I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

A specification is extracted from the statement:

E1. "For every starting time of a job, the number of jobs that are being processed must not exceed the number of existing machines."

It is not necessary to enter the starting times of the jobs in the table of elements as a period element, although it could have been done in that way as well. The reason is that these elements as such are embedded in each job element, and therefore, to allude to them is to refer to each job.

Therefore, the specification falls on each job i. It is a specification of upper bound for the sum of the variables α_k corresponding to jobs k that are processed in the instant of the starting time of job i. These variables will return the total of machines that process those jobs k. This sum assumes that the number of m machines is not exceeded.

$$\forall i: \sum_{k/s_k \le s_i \ \ \& \ \ s_k + d_k > s_i} \alpha_k \le m \tag{7.29}$$

Objective Criterion

Maximize the total weight of the processed jobs

Table 7.16 Comparison of versions of Illustration 7.5

	Individual Version	Collective Version
Number of binary variables	$n \times m$	n
Number of integer variables	0	0
Number of constraints	$n + (m \times R)$	n

O.F.: Max $\sum_{i=1}^{50} p_i \alpha_i$

Let us take a look at a comparison of both models. We consider a general scenario with n jobs and m machines. For the individual version, we consider R as the number of overlapping job pairs $[0 \leq R \leq (n^2 - n)]$ (Table 7.16).

Individual version constraints: (7.27) and (7.28).

Collective version constraints: (7.29).

The number of variables and constraints in the collective version is significantly reduced.

7.6 Individual Elements with Capacity to be Grouped in Collective Elements: Through Grouping into Subsets

The grouping into subgroups of items that have the same data values usually leads to models that are much smaller in size and generally more efficient. We are going to raise the grouping in this section without modifying data values, since it is not necessary. In the illustration, guests of a wedding are grouped according to their characteristics. You will see two versions of modelling, one that does not group and considers each element as individual and the version that groups collective elements.

Illustration 7.6: Assigning Tables to Wedding Guests (Lewis and Carroll 2016)
There is a celebration hall that should seat the guests of a wedding. The living room has tables of two sizes. Tables of 8 seats (18 tables) and tables of 12 seats (12 tables).

200 guests attend the wedding, including seniors, young people, and children. The guests are also classified as either the groom's guest or the bride's guest. Therefore, each guest has the information of age category and origin.

It has been decided that guests who are children must be placed at tables just for them. However, older people and young people can mix at the tables, but what is intended is to have the least number of tables where older and younger people mix (i.e., if at a table they seat young and old people, that table is removed in the objective function).

On the other hand, if a table is used, it must have at least six people seated.

Regarding the groom's guests and the bride's guests, we also intend to mix as little as possible at the wedding. In this way, in the objective function we will minimize the tables where both the mix of older and younger people will be

Table 7.17 Elements of Illustration 7.6 – Unitary Version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Tables	$i = 1\ldots30$	$\mathrm{I_U}$	Capacity	K_i	I	S	–
Seats	–	$\mathrm{C_D}$		K_i			
Guests	$k = 1\ldots200$	$\mathrm{I_U}$	Origin	P_{kp}	B	S	–
			Age category	C_{kc}	B	S	–
Origins	$p = 1{,}2$ 1: Bride 2: Groom	$\mathrm{I_U}$		P_{kp}			
Age categories	$c = 1{,}2{,}3$ 1: Children 2: Young 3: Senior	$\mathrm{I_U}$		C_{kc}			

produced, as groom's guests and bride's guests, but if in a table there are both circumstances, that table only penalizes once, not twice.

For the resolution, we will propose two versions: first of all, a unitary individual version, where each guest is a unitary individual element of the system. In the second version, we will group the guests into collective elements according to their typology, since the statement allows it.

Regarding the tables, both in one version and in another, it is necessary to consider them as individual, although the 18 tables of 8 people are identical, as well as the 12 tables of 12 people. The reason is that they are alluded to in an individual way on several occasions in the problem: the text implies the need to know, with respect to each table, whether or not it penalizes the objective function ("i.e., if both young and old people sit at a table, that table penalizes the objective function"). There is also a specification to sit at least six people at each table that is used. "On the other hand, if you use a table, you must have at least 6 people sitting at it." In addition to all this, each table is individually measurable because the attribute of number of guests acts as capacity attribute.

Individual Version

In this first version, the model is built considering each guest as unitary, since guests are not identical, each one has its age category and its origin as not measurable data.

The table of elements would be as follows:

Table of Elements (Table 7.17)

In the definition of elements, we have made the Origin explicit (groom's guest or bride's guest) and the age category of the guests. We could also have separated the tables according to their size, but to obtain a more simplified model, we have included them all in the same set.

Decision Activities

Table 7.18 Selection rules in Illustration 7.6

Activity	Elements selecting	Selectable elements	Selection	Constraints
Seat	Guests $i = 1\ldots200$	Tables	$= 1$	$\forall k: \quad \sum_{i=1}^{30} \alpha_{ik} = 1$ (7.31)
	Tables $j = 1\ldots30$	Guests	$\leq K_i$	I1

Action: Sitting guests at tables.

Decision variables:

α_{ik}: 1 if I seat guest k at table i; 0 otherwise. $i = 1\ldots30$; $k = 1\ldots200$;

Specifications

1. *Implicit Specifications*

I1. Based on data: There are specifications of capacity consumption for the tables. Each table has a capacity or number of guests that can be seated.

$$\forall i: \quad \sum_{k=1}^{200} \alpha_{ik} \leq K_i \tag{7.30}$$

I2. Quantitative selection rules: It is necessary to analyze the activity of the problem (Table 7.18).

I3. Logical conditions between activities: they do not exist.

I4. Bounds of discrete measurable activities: they do not exist.

I5. Flow balance constraints: they do not exist.

2. *Explicit Specifications*

E1. "Guests who are children have to be placed at tables just for them."

It would be modelled as a logical proposition in which if I seat a child at any table, the guests at that table are all children:

$$\forall i, k/C_{k1} = 1 : \text{IF } \alpha_{ik} = 1 \ \text{ THEN } \sum_{k'=1}^{200} \alpha_{ik'} = \sum_{k'/C_{ik'}=1} \alpha_{ik'} \Rightarrow$$

$$\Rightarrow \forall i, k/C_{k1} = 1 : \text{IF } \alpha_{ik} = 1 \ \text{ THEN } \sum_{k'=1}^{200} \alpha_{ik'} - \sum_{k'/C_{ik'}=1} \alpha_{ik'} = 0 \Rightarrow$$

$$\Rightarrow \text{Ref } \mathrm{f_{LB}} \Rightarrow \forall i, k/C_{k1} = 1 : \text{IF } \alpha_{ik} = 1 \ \text{ THEN } \sum_{k'=1}^{200} \alpha_{ik'} - \sum_{k'/C_{ik'}=1} \alpha_{ik'} \leq 0$$

$$\Rightarrow \text{Ref f}_{14} \Rightarrow \forall i, k/C_{k1} = 1 : \sum_{k'=1}^{200} \alpha_{ik'} - \sum_{k'/C_{ik'}=1} \alpha_{ik'} \leq K_i(1 - \alpha_{ik}) \tag{7.32}$$

This expression is equivalent to defining that: “If a child is at a table, there cannot be young and senior guests at that table”:

$$\forall i, k/C_{k1} = 1 : \text{IF } \alpha_{ik} = 1 \text{ THEN } \sum_{k'/C_{k'1}=0} \alpha_{ik'} = 0$$

$$\Rightarrow \text{Ref.f}_{\text{LB}} \Rightarrow \forall i, k/C_{k1} = 1 : \text{IF } \alpha_{ik} = 1 \text{ THEN } \sum_{k'/C_{k'1}=0} \alpha_{ik'} \leq 0$$

$$\Rightarrow \text{Ref.f}_{37} \Rightarrow \forall i, k/C_{k1} = 1 : \sum_{k'/C_{k'1}=0} \alpha_{ik'} \leq K_i(1 - \alpha_{ik}) \tag{7.33}$$

This proposition could also have been modelled through the statement that children can only sit with children, so we could have expressed it in negative as follows: “a child cannot be sitting at a table with a non-child guest.”

Therefore, the proposition falls on any table, any child and any non-child guest:

$$\forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \text{NOT}(\alpha_{ik} = 1 \text{ AND } \alpha_{ik'} = 1) \Rightarrow$$
$$\Rightarrow \text{Ref.f}_{37} \Rightarrow \forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \text{NOT}(\alpha_{ik} + \alpha_{ik'} \geq 2)$$

$$\Rightarrow \text{Ref.f}_{11} \Rightarrow \forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \alpha_{ik} + \alpha_{ik'} \leq 1 \tag{7.34}$$

It is a simpler restriction, but we need many more to impose the same specification.

E2. “If a table is used, it must have at least 6 people seated at it.”

It is also a logical proposition that uses a logical calculation not defined until now as an input to the condition: “a table is used” or “used table.”

Since the logical calculation is defined as input in the proposition of the specification, we can choose to propose the proposition without defining that logical calculation, since the modelling of the proposition itself will help to define the calculation. We propose it in the following way:

“If there are guests seated at a table, there must be at least 6.”

$$\forall i : \text{IF } \sum_{k=1}^{200} \alpha_{ik} > 0 \text{ THEN } \sum_{k=1}^{200} \alpha_{ik} \geq 6$$

$$\Rightarrow \text{Ref.f}_5 \Rightarrow \forall i : \text{IF } \sum_{k=1}^{200} \alpha_{ik} \geq 1 \text{ THEN } \sum_{k=1}^{200} \alpha_{ik} \geq 6$$

$$\Rightarrow \text{Ref.f}_{18,}\text{f}_{21} \Rightarrow \forall i: \sum_{k=1}^{200} \alpha_{ik} \geq \omega_i \tag{7.35}$$

$$\Rightarrow \forall i: \sum_{k=1}^{200} \alpha_{ik} \leq k_i\omega_i \tag{7.36}$$

$$\Rightarrow \forall i: \sum_{k=1}^{200} \alpha_{ik} \geq 6\omega_i \tag{7.37}$$

ω_i is the logical calculation that responds to whether a table has been used or not. The third group 7.37 makes the group 7.35 unnecessary.

Objective Criterion

Minimize mixed tables.

To establish the expression of the objective function, we need to find out if each table has been mixed, that is, if it penalizes in the objective function.

The mix can occur by mixing guests of age category 2 and 3, or by mixing guests from different origins. Everything can be collected in a logical calculation per table:

Binary logical calculation: Mixed table.
Applied to: Tables $i = 1\ldots30$.
Variables:

$$\delta_i = \begin{cases} 1 \text{ if table } i \text{ is mixed} \\ 0 \ \text{ otherwise} \end{cases}$$

Logical proposition: We can raise it with multiple conditionals to determine value 1 of δ_i (value 0 is not necessary to determine it because value 1 supposes cost [Ref. Sv]) or by means of a smaller proposition:

Multiple conditionals:

$$\forall i: \text{IF } (\alpha_{i1} = 1 \ \text{ AND } \ \alpha_{i2} = 1 \ \text{ AND } \ (C_{12} \neq C_{22} \text{ OR } C_{13} \neq C_{23} \text{ OR } \ P_{11} \neq P_{21})) \text{ OR}$$
$$(\alpha_{i1} = 1 \ \text{ AND } \ \alpha_{i3} = 1 \ \text{ AND } \ (C_{12} \neq C_{32} \text{ OR } C_{13} \neq C_{33} \text{ OR } \ P_{11} \neq P_{31}) \text{ OR} \ldots$$
$$\text{THEN} \quad \delta_i = 1$$

$$\Rightarrow \text{Ref.f}_{40} \Rightarrow \begin{array}{l} \forall i, k, k' : \text{IF } \alpha_{ik} = 1 \ \text{ AND } \ \alpha_{ik'} = 1 \\ \text{AND } \ (C_{k2} \neq C_{k'2} \text{ OR } C_{k3} \neq C_{k'3} \text{ OR } \ \text{P}_{k1} \neq P_{k'1}) \ \text{ THEN } \quad \delta_i = 1 \end{array}$$

In a smaller proposition:

$$\forall i: \text{IF} \left(\sum_{k/C_{k2}=1} \alpha_{ik} > 0 \ \text{ AND } \ \sum_{k/C_{k3}=1} \alpha_{ik} > 0 \right)$$

$$\text{OR}\left(\sum_{k/P_{k1}=1}\alpha_{ik}>0\ \ \text{AND}\ \ \sum_{k/P_{k2}=1}\alpha_{ik}>0\right)\ \ \text{THEN}\quad \delta_i=1 \tag{7.38}$$

Proposition modelling: (We use Proposition 7.38)

$$\Rightarrow \text{Ref.f}_5 \Rightarrow\Rightarrow \begin{array}{l} \forall i:\text{IF}\left(\sum_{k/C_{k2}=1}\alpha_{ik}\geq 1\ \ \text{AND}\ \ \sum_{k/C_{k3}=1}\alpha_{ik}\geq 1\right) \\ \text{OR}\left(\sum_{k/P_{k1}=1}\alpha_{ik}\geq 1\ \ \text{AND}\ \ \sum_{k/P_{k2}=1}\alpha_{ik}\geq 1\right)\ \ \text{THEN}\quad \delta_i=1 \end{array}$$

$$\text{Ref.f}_{35}\Rightarrow \boxed{\begin{array}{l} \forall i:\lambda_{i1}=1\quad \text{IF AND ONLY IF}\ \sum_{k/C_{k2}=1}\alpha_{ik}\geq 1 \\ \forall i:\lambda_{i2}=1\quad \text{IF AND ONLY IF}\ \sum_{k/C_{k3}=1}\alpha_{ik}\geq 1 \\ \forall i:\lambda_{i3}=1\quad \text{IF AND ONLY IF}\ \sum_{k/P_{k1}=1}\alpha_{ik}\geq 1 \\ \forall i:\lambda_{i4}=1\quad \text{IF AND ONLY IF}\ \sum_{k/P_{k2}=1}\alpha_{ik}\geq 1 \end{array}}$$

$$\Rightarrow \text{Ref.f}_{25}\Rightarrow \boxed{\begin{array}{l} \forall i:\ \sum_{k/C_{k2}=1}\alpha_{ik}\geq\lambda_{i1} \\ \forall i:\ \sum_{k/C_{k2}=1}\alpha_{ik}\leq K_i\lambda_{i1} \\ \forall i:\ \sum_{k/C_{k3}=1}\alpha_{ik}\geq\lambda_{i2} \\ \forall i:\ \sum_{k/C_{k3}=1}\alpha_{ik}\leq K_i\lambda_{i2} \end{array}} \tag{7.39}$$

$$\begin{array}{l} \forall i:\ \sum_{k/P_{k1}=1}\alpha_{ik}\geq\lambda_{i3} \\ \forall i:\ \sum_{k/P_{k1}=1}\alpha_{ik}\leq K_i\lambda_{i3} \\ \forall i:\ \sum_{k/P_{k2}=1}\alpha_{ik}\geq\lambda_{i4} \\ \forall i:\ \sum_{k/P_{k2}=1}\alpha_{ik}\leq K_i\lambda_{i4} \end{array} \tag{7.40}$$

$$\text{Ref. f}_{37}\Rightarrow\ \forall\, i:\text{IF}\ (\lambda_{i1}+\lambda_{i2}\geq 2)\ \ \text{OR}\ (\lambda_{i3}+\lambda_{i4}\geq 2)\ \ \text{THEN}\quad \delta_i=1$$

$$\text{Ref.f}_{35}\Rightarrow \begin{array}{l} \forall i:(\lambda_{i1}+\lambda_{i2}\geq 2)\ \ \text{IF AND ONLY IF}\ \ \pi_{i1}=1 \\ \forall i:(\lambda_{i3}+\lambda_{i4}\geq 2)\ \ \text{IF AND ONLY IF}\ \ \pi_{i2}=1 \end{array}$$

$$\Rightarrow \text{Ref.f}_{25} \Rightarrow \boxed{\begin{array}{l} \forall i: \lambda_{i1}+\lambda_{i2} \geq 2\pi_{i1} \\ \forall i: \lambda_{i1}+\lambda_{i2} \leq (1-\pi_{i1})+2\pi_{i1} \\ \forall i: \lambda_{i3}+\lambda_{i4} \geq 2\pi_{i2} \\ \forall i: \lambda_{i3}+\lambda_{i4} \leq (1-\pi_{i2})+2\pi_{i2} \end{array}} \tag{7.41}$$

$\Rightarrow$Ref. $f_{34} \Rightarrow\ \forall\ i$: IF π_{i1} + $\pi_{i2} \geq 1$ THEN $\delta_i = 1$
$\Rightarrow$Ref. $f_3 \Rightarrow\ \forall\ i$: IF $\delta_i = 0$ THEN π_{i1} + $\pi_{i2} \leq 0$

$$\begin{aligned} &\text{Ref.f}_7 \Rightarrow \forall i: \text{IF } 1-\delta_i = 1 \text{ THEN } \quad \pi_{i1}+\pi_{i2} \leq 0 \\ &\Rightarrow \boxed{\forall i: \pi_{i1}+\pi_{i2} \leq 2\delta_i} \end{aligned} \tag{7.42}$$

O.F.: Min $\sum_{i=1}^{30} \delta_i$

Collective Version

It is easy to organize guests into six different groups (Table 7.19).

Within each group, there is a set of identical items that do not have any specification or calculation on each individually in the system.

Obviously, as I know the guests, the values N_1–N_6 are known. For the table of elements, we use a set with those six groups of guests.

Table of Elements (Table 7.20)

Decision Activities

Action: Sitting guests at tables.
Decision variables: x_{ik} = Number of guests of group k sitting at table i.

Specifications

1. *Implicit Specifications*

I1. Based on data:

I1.1. Table capacity:

$$\forall i: \quad \sum_{k=1}^{6} x_{ik} \leq K_i \tag{7.43}$$

I1.2. The new data of existence of each group of guests gives rise to an assignment specification based on an action of determined value, since it is necessary to seat the invited Nk of each group:

Table 7.19 Subgroups of guests in Illustration 7.6

Guests					
Groom's guest			Bride's guest		
Children	Young	Seniors	Children	Young	Seniors
$1\ldots N_1$	$1\ldots N_2$	$1\ldots N_3$	$1\ldots N_4$	$1\ldots N_5$	$1\ldots N_6$

Table 7.20 Elements of Illustration 7.6 – Collective Version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Tables	$i = 1\ldots30$	$\mathrm{I_U}$	Capacity	K_i	I	S	–
Seats	–	$\mathrm{C_D}$		K_i			
Groups of guests	$k = 1\ldots6$	$\mathrm{C_D}$	Number of guests	N_k	I	W	–
			Origin	P_{kp}	B	S	–
			Age category	C_{kc}	B	S	–
Origins	$p = 1{,}2$	$\mathrm{I_U}$		P_{kp}			
Age categories	$c = 1{,}2{,}3$	$\mathrm{I_U}$		C_{kc}			

$$\forall k : \quad \sum_{i=1}^{30} x_{ik} = N_k \tag{7.44}$$

I2. Quantitative selection rules: they do not exist.
I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

E1. "Guests who are children have to be placed at tables just for them."

This logical proposition is stated with guest numerals: There cannot be a positive number of non-child guests at a table with a positive number of children.

Therefore, the proposition falls on any table, the two groups of children and the four groups that are not children:

$$\forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \mathrm{NOT}(x_{ik} > 0 \ \text{ AND } \ x_{ik'} > 0) \Rightarrow$$

$$\Rightarrow \mathrm{Ref.f}_5 \Rightarrow \forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \mathrm{NOT}(x_{ik} \geq 1 \ \text{ AND } \ x_{ik'} \geq 1)$$

$$\Rightarrow \mathrm{Ref.f}_{35} \Rightarrow \begin{array}{l} \forall i, k/C_{k1} = 1 : x_{ik} \geq 1 \ \text{ IF AND ONLY IF } \ \beta_{ik} = 1 \\ \forall i, k'/C_{k'1} = 0 : x_{ik'} \geq 1 \ \text{ IF AND ONLY IF } \ \delta_{ik'} = 1 \end{array}$$

$$\Rightarrow \left.\begin{array}{l} \forall i, k/C_{k1} = 1 : x_{ik} \geq \ \beta_{ik} \\ \forall i, k/C_{k1} = 1 : x_{ik} \leq K_i\beta_{ik} \\ \forall i, k'/C_{k'1} = 0 : x_{ik'} \geq \ \delta_{ik'} \\ \forall i, k'/C_{k'1} = 0 : x_{ik'} \leq \ K_i\delta_{ik'} \end{array}\right| \tag{7.45}$$

$$\Rightarrow \text{Ref.f}_{37} \Rightarrow \forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \text{NOT}(\beta_{ik} + \ \delta_{ik'} \geq 2)$$

$$\Rightarrow \text{Ref.f}_{11} \Rightarrow \forall i, k/C_{k1} = 1, k'/C_{k'1} = 0 : \beta_{ik} + \ \delta_{ik'} \leq 1 \tag{7.46}$$

E2. "If a table is used, it must have at least 6 people seated at it."

As stated in the unitary version:

"If there are guests seated at a table, there must be at least 6."

$$\forall i : \text{IF } \sum_{k=1}^{6} x_{ik} > 0 \quad \text{THEN } \sum_{k=1}^{6} x_{ik} \geq 6$$

$$\Rightarrow \text{Ref.f}_5 \Rightarrow \forall i : \text{IF } \sum_{k=1}^{6} x_{ik} \geq 1 \text{ THEN } \sum_{k=1}^{6} x_{ik} \geq 6 \Rightarrow$$

$$\Rightarrow \text{Ref.f}_{18,}\text{f}_{21} \Rightarrow \forall i : \sum_{k=1}^{6} x_{ik} \geq \omega_i \tag{7.47}$$

$$\Rightarrow \forall i : \sum_{k=1}^{6} x_{ik} \leq k_i\omega_i \tag{7.48}$$

$$\Rightarrow \forall i : \sum_{k=1}^{6} x_{ik} \geq 6\omega_i \tag{7.49}$$

ω_i is the logical calculation that responds to whether a table has been used. The group of constraints (7.49) makes the group (7.47) unnecessary.

Objective Criterion

Minimize mixed tables.

To establish the expression of the objective function, we need to find out if each table has been mixed, that is, if it penalizes in the objective function.

Binary Logical Calculation: Mixed table.
Applied to: Tables $i = 1\ldots30$.
Variables:

$$\delta_i = \begin{cases} 1 \text{ if table } i \text{ is mixed} \\ 0 \text{ otherwise} \end{cases}$$

Logical Propositions:

$$\forall i : \text{IF} \left(\sum_{k/C_{k2}=1} x_{ik} > 0 \ \text{ AND } \sum_{k/C_{k3}=1} x_{ik} > 0 \right) \qquad [\text{Ref. S}_\text{V}]$$

$$\text{OR} \left(\sum_{k/P_{k1}=1} x_{ik} > 0 \ \text{ AND } \sum_{k/P_{k2}=1} x_{ik} > 0 \right) \ \text{ THEN } \quad \delta_i = 1$$

Proposition modelling:

$$\Rightarrow \text{Ref.f}_5 \Rightarrow\Rightarrow \begin{array}{l} \forall i : \text{IF} \left(\sum_{k/C_{k2}=1} x_{ik} \geq 1 \ \text{ AND } \sum_{k/C_{k3}=1} x_{ik} \geq 1 \right) \\ \text{OR} \left(\sum_{k/P_{k1}=1} x_{ik} \geq 1 \ \text{ AND } \sum_{k/P_{k2}=1} x_{ik} \geq 1 \right) \ \text{ THEN } \quad \delta_i = 1 \end{array}$$

$$\Rightarrow \text{Ref.f}_{35} \Rightarrow \boxed{\begin{array}{ll} \forall i : \lambda_{i1} = 1 \quad \text{IF AND ONLY IF} & \sum_{k/C_{k2}=1} x_{ik} \geq 1 \\ \forall i : \lambda_{i2} = 1 \quad \text{IF AND ONLY IF} & \sum_{k/C_{k3}=1} x_{ik} \geq 1 \\ \forall i : \lambda_{i3} = 1 \quad \text{IF AND ONLY IF} & \sum_{k/P_{k1}=1} x_{ik} \geq 1 \\ \forall i : \lambda_{i4} = 1 \quad \text{IF AND ONLY IF} & \sum_{k/P_{k2}=1} x_{ik} \geq 1 \end{array}}$$

$$\Rightarrow \text{Ref.f}_{25} \Rightarrow \boxed{\begin{array}{l} \forall i : \sum_{k/C_{k2}=1} x_{ik} \geq \lambda_{i1} \\ \forall i : \sum_{k/C_{k2}=1} x_{ik} \leq K_i \lambda_{i1} \\ \forall i : \sum_{k/C_{k3}=1} x_{ik} \geq \lambda_{i2} \\ \forall i : \sum_{k/C_{k3}=1} x_{ik} \leq K_i \lambda_{i2} \end{array}} \tag{7.50}$$

$$\begin{array}{l} \forall i : \sum_{k/P_{k1}=1} x_{ik} \geq \lambda_{i3} \\ \forall i : \sum_{k/P_{k1}=1} x_{ik} \leq K_i \lambda_{i3} \\ \forall i : \sum_{k/P_{k2}=1} x_{ik} \geq \lambda_{i4} \\ \forall i : \sum_{k/P_{k2}=1} x_{ik} \leq K_i \lambda_{i4} \end{array} \tag{7.51}$$

$$\Rightarrow \text{Ref. f}_{37} \Rightarrow \ \forall\, i : \text{IF } (\lambda_{i1} + \lambda_{i2} \geq 2) \ \text{ OR } (\lambda_{i3} + \lambda_{i4} \geq 2) \ \text{ THEN } \quad \delta_i = 1$$

$$\Rightarrow \text{Ref.f}_{35} \Rightarrow \begin{array}{l} \forall i : (\lambda_{i1} + \lambda_{i2} \geq 2) \ \text{ IF AND ONLY IF } \ \pi_{i1} = 1 \\ \forall i : (\lambda_{i3} + \lambda_{i4} \geq 2) \ \text{ IF AND ONLY IF } \ \pi_{i2} = 1 \end{array}$$

$$\Rightarrow \text{Ref.f}_{25} \Rightarrow \boxed{\begin{array}{l} \forall i : \lambda_{i1} + \lambda_{i2} \geq 2\pi_{i1} \\ \forall i : \lambda_{i1} + \lambda_{i2} \leq (1 - \pi_{i1}) + 2\pi_{i1} \\ \forall i : \lambda_{i3} + \lambda_{i4} \geq 2\pi_{i2} \\ \forall i : \lambda_{i3} + \lambda_{i4} \leq (1 - \pi_{i2}) + 2\pi_{i2} \end{array}} \quad (7.52)$$

$\Rightarrow$Ref. $f_{34} \Rightarrow \ \forall\, i$: IF $\pi_{i1} + \pi_{i2} \geq 1$ THEN $\delta_i = 1$
$\Rightarrow$Ref. $f_3 \Rightarrow \ \forall\, i$: IF $\delta_i = 0$ THEN $\pi_{i1} + \pi_{i2} \leq 0$

$$\text{Ref.f}_7 \Rightarrow \forall i : \text{IF } 1 - \delta_i = 1 \ \text{ THEN } \ \pi_{i1} + \pi_{i2} \leq 0$$
$$\Rightarrow \boxed{\forall i : \pi_{i1} + \pi_{i2} \leq 2\delta_i} \quad (7.53)$$

F.O.: Min $\sum_{i=1}^{30} \delta_i$

Let us take a look at a comparison of both models (Table 7.21).

Constraints considered:

Individual Version: 7.30, 7.31, 7.32, 7.36, 7.37, 7.39, 7.40, 7.41, 7.42

Collective Version: 7.43, 7.44, 7.45, 7.46, 7.48, 7.49, 7.50, 7.51, 7.52, 7.53

The collective version is much more efficient with respect to the size than the individual version, especially in the number of variables.

7.7 Individual Elements with Capacity to Be Grouped in Collective Elements: Through Small Changes in the Data Values

These are systems in which it is permissible to make small variations in the data, in order to simplify the problem. With this we can make groups of elements as items of a collective element. It may also be necessary to make small changes in the statement to avoid referring to each item in particular and to do so by means of numerals of the collective element. Let us see an illustration that groups the citizens of a city according to their address, assuming identical distances of citizens who live in the same street or stretch of street. Related problems in Wang et al. (2018)

Illustration 7.7: Allocation of Health Centers to Citizens

There is a city formed by five health centers. Due to changes in locations, it has been decided to restructure the allocation of health centers to its 50,000 citizens. We know

Table 7.21 Comparison of version sizes in Illustration 7.6

	Individual Version	Collective Version
Number of binary variables	6240	420
Number of integer variables	0	180
Number of constraints	680+ (30 × N° of children)	606

the address of each citizen, and therefore, the system allows us to know the distance from his address to each health center. Each health center has a capacity that is expressed as the number of citizens that can be attended to per day. It is estimated that 1% of people go to the doctor daily. The objective is to minimize the sum of the distances from each citizen address to the assigned health center.

In the statement each citizen must be considered as an individual element of the system, since each one will have a distance from each health center, without there being many coincidences. On the other hand, they are unitary because they do not have measurable data or properties. In the second version, we will group citizens who share a common area, such as a street or a section of street. These citizens will therefore have the same values of distance to health centers.

Individual Version

The table of elements would be as follows:

Table of Elements (Table 7.22)

In the definition of data, we logically ignore the address, since it is not a numeric attribute.

Decision Activities

Action: Allocate citizens to health centers.
Decision variables:
α_{ij}: 1 if I allocate citizen i to health center j; 0 otherwise.
$i = 1...50.000; j = 1...5.$

Specifications

1. *Implicit Specifications*

I1. Based on data: Capacity specification of each health center (although it can also be considered as explicit by the description of the capacity).

$$\forall j: \quad \sum_{i=1}^{50000} 0,01\alpha_{ij} \leq K_j \tag{7.54}$$

I2. Quantitative selection rules: Variables α_{ij} define quantitative selection rules (Table 7.23).

Table 7.22 Elements of Illustration 7.7 – Individual version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Citizens	$i = 1\ldots 50.000$	$\mathrm{I_U}$	Distance	D_{ij}	C	S	–
Health centers	$j = 1\ldots 5$	$\mathrm{I_U}$	Capacity	K_j	I	W	–
				D_{ij}			

Table 7.23 Selection rules in Illustration 7.7

Activity	Elements selecting	Selectable elements	Selection	Constraints
Allocate	Citizens $i = 1\ldots 50{,}000$	Health centers	$= 1$	$\forall i: \quad \sum_{j=1}^{5} \alpha_{ij} = 1$ (7.55)
	Health centers $j = 1\ldots 5$	Citizens	$\leq K_i$	I1

I3. Logical conditions between activities: they do not exist.

I4. Bounds of discrete measurable activities: they do not exist.

I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

They do not exist.

Objective Criterion

$$\text{O.F.: MIN} \sum_{i=1}^{50000} \sum_{j=1}^{5} D_{ij}\alpha_{ij}$$

Collective Version

In the collective version, we assign the same distance value to the citizens who live in the same area, street or street section.

This version requires a slight modification of the statement, to stop referring to each citizen in particular. We can refer in particular to each group of citizens, but we must avoid referring to each citizen individually.

On the other hand, we need to assign an attribute to each group of citizens with the number of citizens it has. Assuming that a total of m groups has been formed, we will use the data N_i, $i = 1\ldots$m to name the number of citizens of each group.

The statement would look like this:

There is a city formed by five health centers. Due to changes in locations, it has been decided to restructure the allocation of health centers to its 50,000 citizens. We know the address of each citizen, and therefore, the system allows us to know the distance from his area to each health center. Each health center has a capacity that is expressed as the number of citizens that can be attended to per day. It is estimated that 1% of people go to the doctor daily. The objective is to minimize the sum of the distances from citizens to the assigned health centers.

Table of Elements (Table 7.24)

Table 7.24 Elements of Illustration 7.7 – Collective Version

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Groups of citizens	$i = 1 \ldots m$	$\mathrm{C_D}$	Distance	D_{ij}	C	S	–
			Quantity	N_i	I	W	–
Health centers	$j = 1 \ldots 5$	$\mathrm{I_U}$	Capacity	K_j	I	W	–
				D_{ij}			

Table 7.25 Elements of Illustration 7.7 – Collective Version with continent and content elements

			Data				
Elements	Set	QN	Name	Param	Type	Belonging	Value
Group of citizens	$i = 1 \ldots m$	$\mathrm{I_U}$	Distance	D_{ij}	C	S	–
			Quantity	N_i	I	S	
Health centers	$j = 1 \ldots 5$	$\mathrm{I_U}$	Capacity	K_j	I	S	–
				D_{ij}			
Citizens	–	$\mathrm{C_D}$		K_j, N_i			

In the table of elements, we have considered each group of citizens as a collective, and we have introduced them all in a set of m elements. This scenario, as in Illustration 7.2 on elements where we could make the content and the continent explicit, also allows a table of elements in which we consider each group as unitary (continent) and the citizen as collective (content). In that case, the table would have the following form (Table 7.25).

In this table we have grouped the content in a single collective element since the citizens are identical with respect to their own data. The capacity attribute of each health center is shared with the Citizens element. This integer attribute had been own, while there was no need to create the collective element Citizens. Either table is valid to represent the elements of the problem.

Decision Activities

Action: Allocate groups of citizens to Health centers.
Decision variables:
x_{ij}: Number of citizens of group i allocated to Health center j.

Specifications

1. Implicit Specifications

I1. Based on data:

I1.1. Capacity of each health center: We keep it as implicit. Now the activity is x_{ij}.

$$\forall j: \quad \sum_{i=1}^{m} 0,01 x_{ij} \leq K_j \tag{7.56}$$

I1.2. The citizen stock data of each group generates a balance specification:

$$\forall i: \quad \sum_{j=1}^{5} x_{ij} = N_i \tag{7.57}$$

I2. Quantitative selection rules: they do not exist.
I3. Logical conditions between activities: they do not exist.
I4. Bounds of discrete measurable activities: they do not exist.
I5. Flow balance constraints: they do not exist.

2. Explicit Specifications

They do not exist.

Objective Criterion

O.F.: MIN $\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{5} D_{ij} x_{ij}$

Comparison of both models (Table 7.26).

7.8 Items of Indeterminate Collective Elements that Need to Be Defined Individually

According to most of the illustrations seen so far, if a system has an element with a number of indeterminate items, that element is considered indeterminate collective, by default. However, if the system description refers individually to the possible items that the element will have, it is necessary to consider each possible item as an individual element. This means that the collective element must be converted into a set of individual elements. As we do not know how many items we can have, from the description of the system, it is necessary to calculate an upper bound of the number of items, keeping it as small as possible to avoid an unnecessary increase in the number of variables that can be generated in the problem.

By slightly modifying statement 7.6, we can illustrate this case.

Illustration 7.8: Illustration 7.6 with an Indeterminate Number of Tables
There is a celebration hall that should place the guests of a wedding. The living room can use tables of two sizes (tables of 8 seats and tables of 12 seats).

Table 7.26 Comparison of versions of Illustration 7.7

	Individual Version	Collective Version
Number of binary variables	250.000	0
Number of integer variables	0	5 *m*
Number of constraints	50.005	*m* + 5

200 guests attend the wedding, including seniors, young people, and children. The guests are also classified as the groom's guest or the bride's guest. Therefore, each guest has the information of age category and origin.

It has been decided that guests who are children must be placed at tables just for them. However, older people and young people can mix at the tables, but what is intended is to have the least number of tables where older and younger people mix (i.e., if at a table they seat young and old people, that table penalizes in the objective function).

On the other hand, if a table is used, it must have at least six people seated at it.

Regarding the groom's guests and bride's guests, we also intend to mix as little as possible at the wedding. In this way, in the objective function we will minimize the tables where both the mix of older and younger people will be produced, as groom's guests and bride's guests, but if in a table there are both circumstances, that table only penalizes once, not twice.

The modification we have made has been to disregard a certain number of tables, both 8 and 12 seats. Therefore, the tables become indeterminate, which could lead us to visualize each type of table as an indeterminate collective element. However, as in the statement there are several specifications that allude to (tables) individual items (if a table is used, it must have at least six people seated at it, if at a table they seat young and old people, that table penalizes in the objective function), it is necessary to keep the tables as individual elements. The problem is that its number is indeterminate; therefore it is necessary to calculate an upper bound of the number of tables of 8 and 12 people that could be used without losing possible solutions to the problem. If we have 200 guests, the maximum number of tables of 8 that could be used is obtained by rounding the quotient between the number of guests and the number of places of that type of tables: $\left\lceil \frac{200}{8} \right\rceil = 25$tables. The same applies to tables of 12 places: $\left\lceil \frac{200}{12} \right\rceil = 17$tables. In this way, the table of elements for the individual guest version would have a total of 42 tables:

Table of Elements (Table 7.27)

Let us take a look at another illustration to finish the chapter.

Illustration 7.9

There is a set of n tasks that must be processed in a machining factory. The processing is done in a machine model for which we must subcontract units. Each task has a process time and is processed completely on the same machine. The

Table 7.27 Elements of Illustration 7.8

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Tables	$i = 1\ldots42$	I_U	Capacity	K_i	I	S	–
Seats	–	C_D		K_i			
Guests	$k = 1\ldots200$	I_U	Origin	P_{kp}	B	S	–
			Age category	C_{kc}	B	S	–
Origins	$p = 1,2$	I_U		P_{kp}			
Age categories	$c = 1,2,3$	I_U		C_{kc}			

Table 7.28 Elements of Illustration 7.9

Elements	Set	QN	Data				
			Name	Param	Type	Belonging	Value
Factory	–	I_U	Time limit	TL	C	W	–
Tasks	$i = 1\ldots n$	I_U	Process time	t_i	C	W	–
Machines	$j = 1\ldots n$	I_U					

objective of the problem is to minimize the number of machines needed to process all the tasks in a given time limit.

The machines could be considered as an indeterminate collective element, but the system needs its individual consideration, since it is necessary to know in which machine each task is processed. Therefore, an upper bound of its number is necessary. This level can be obtained in various ways. The simplest is to use as many machines as jobs. The tasks are processed completely in the same machine, so they must be considered non-divisible and therefore unitary.

Table of Elements (Table 7.28)

References

Arkin, E. M., & Silverberg, E. B. (1987). Scheduling jobs with fixed start and end times. *Discrete Applied Mathematics, 18*, 1–8.

Kroon, L. G., Salomon, M., & Van Wassenhove, L. N. (1995). Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research, 82*, 190–205.

Lewis, R., & Carroll, F. (2016). Creating seating plans: A practical application. *Journal of the Operational Research Society, 67*, 1353–1362.

Wang, L., Huan, S., & Lu, G. (2018). Healthcare facility location-allocation optimization for China’s developing cities utilizing a multi-objective decision support approach. *Sustainability, 10*, 4580.