

Chapter 2

Mixed-Integer Linear Programming

This chapter is from the book:

Castillo, E., Conejo A.J., Pedregal, P., García, R. and Alguacil, N. (2002). Building and Solving Mathematical Programming Models in Engineering and Science, Pure and Applied Mathematics Series, Wiley, New York.

Copyright 2002 by John Wiley and Sons, Inc.

This material is used by permission of John Wiley and Sons, Inc.

2.1 Introduction

In Chapter 1 we dealt with linear programming problems where the variables involved were real numbers. However, in many cases of real life, some variables are not real but integers, or they are even more restricted, as binary variables, that take values 0 or 1 only. We shall see in Chapter 7 that using integer variables adds more difficulties to the linear programming problem, because of the lack of continuity.

In this chapter we give some real examples of integer linear programming problems (ILPP), in some of which we use binary variables.

2.2 The 0–1 Knapsack Problem

An important class of integer programming problems are those where the variables of the problem can take only two values. This situation can be formulated using the *0–1 variables*. Each value is associated with one of the possibilities of

a binary choice:

$$x = \begin{cases} 1 & \text{if the event occurs} \\ 0 & \text{otherwise} \end{cases}$$

A classical problem involving this type of variable is the *0-1 knapsack problem*. Consider a person who must pack a hike. Assume that there exist a set of items that have a utility for this person, and there exist a limitation on the number of items the person can carry. The problem consists of choosing a subset of items to maximize the sum of the utilities while not exceeding the carrying capacity of the hiker.

The problem has the following elements:

1. Data

- n : the number of objects
- a_j : the weight of the object j
- c_j : the utility of object j
- b : the capacity of the knapsack (hiker)

2. Variables

$$x_j = \begin{cases} 1 & \text{if the object } j \text{ is put in the knapsack} \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

3. Constraints.

The capacity is not exceeded:

$$\sum_{j=1}^n a_j x_j \leq b$$

4. Function to be maximized.

The objective of this problem is to maximize the utility, which can be stated as

$$Z = \sum_{j=1}^n c_j x_j$$

Example 2.1 (The ship owner). A ship owner has a freighter with a capacity of 700 tons. The firm transports containers of different weights for a specific route. On the current trip the ship owner could ship some of the following containers:

Container	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
Weight	100	155	50	112	70	80	60	118	110	55

The decisionmaker's firm would determine the freight such that it maximizes the transported load.

This problem could be formulated as a 0–1 knapsack problem. The variables are:

$$x_j = \begin{cases} 1 & \text{if container } j \text{ is shipped} \\ 0 & \text{otherwise} \end{cases}$$

The objective is to maximize the freight that will be transported by the freighter:

$$\begin{aligned} Z = & 100x_1 + 155x_2 + 50x_3 + 112x_4 + 70x_5 \\ & + 80x_6 + 60x_7 + 118x_8 + 110x_9 + 55x_{10} \end{aligned}$$

and the constraint is that the freight cannot exceed the capacity of the ship:

$$\begin{aligned} 100x_1 + 155x_2 + 50x_3 + 112x_4 + 70x_5 + 80x_6 \\ + 60x_7 + 118x_8 + 110x_9 + 55x_{10} \leq 700 \end{aligned}$$

Note that here $a_i = c_i; \forall i$, because the utility coincides with the weight.

The optimal freight consists of using the containers: $c_1, c_3, c_4, c_5, c_6, c_7, c_8, c_9$. The optimal value is 700, which means that the ship is full. ■

2.3 Identifying Relevant Symptoms

Let $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ be a given set of possible diseases, and assume that physicians, when identifying the diseases associated with a set of patients, usually base their decisions on a set of symptoms $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$. Assume that we want to identify a minimal subset of symptoms $\mathcal{S}_a \subset \mathcal{S}$, such that all diseases can be perfectly distinguished from each other according to the levels of symptoms in \mathcal{S}_a . Finding the minimum set of symptoms is important because it implies minimizing the cost of the diagnosis process.

The problem has the following elements:

1. Data

\mathcal{D} : the set of diseases

\mathcal{S} : the set of symptoms

n : the number of diseases (cardinal of \mathcal{D})

m : the number of symptoms (cardinal of \mathcal{S})

c_{ij} : the level of symptom j associated with disease i

d_{ikj} : discrepancy between diseases i and k due to symptom j

a : the minimum required discrepancy level (to be explained below)

2. Variables

$$x_j = \begin{cases} 1 & \text{if the symptom } j \text{ belongs to } \mathcal{S}_a \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

3. **Constraints.** The subset \mathcal{S}_a must be sufficient for a clear distinction of all diseases:

$$\sum_{j=1}^m x_j d_{ikj} \geq a; \quad \forall i, k \in \{1, 2, \dots, n\}, i \neq k \quad (2.3)$$

where

$$d_{ikj} = \begin{cases} 1 & \text{if } c_{ij} \neq c_{kj} \\ 0 & \text{if } c_{ij} = c_{kj} \end{cases} \quad (2.4)$$

measures the discrepancy between diseases D_i and D_k in terms of the symptoms in \mathcal{S}_a , and $a > 0$ is the discrepancy level we desire. Note that the larger the value of a , the larger the number of required symptoms (cardinal of \mathcal{S}_a). In this case

$$\sum_{j=1}^m x_j d_{ikj}$$

coincides with the number of symptoms in \mathcal{S}_0 that take different levels for diseases D_i and D_k , and a is the corresponding minimum number, for any pair (D_i, D_k) of diseases, that are required to have an acceptable subset \mathcal{S}_a . This means that $a - 1$ symptoms can be missing and we still can differentiate any pair of diseases (D_i, D_k) .

4. **Function to be minimized.** The objective of this problem is to minimize the number of selected symptoms, the cardinal of the set \mathcal{S}_0 :

$$Z = \sum_{j=1}^m x_j.$$

The problem as stated above allows us to determine a minimal subset \mathcal{S}_0 , associated with $a = 0$, of symptoms of the set \mathcal{S} which allows identification of the diseases in the set \mathcal{D} . However, if the diseases are to be identified with some missing information, the set \mathcal{S}_0 can become useless. So, we normally use $a > 0$.

Once we have selected the relevant symptoms to identify all diseases, we can determine the relevant symptoms associated with disease i . This can be done by minimizing

$$Z = \sum_{j=1}^m x_j$$

subject to

$$\sum_{j=1}^m x_j d_{ikj} > a; \quad k \in \{1, 2, \dots, n\}, i \neq k \quad (2.5)$$

Table 2.1: Symptoms associated with all diseases in Example 2.2

Disease	Symptoms							
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
D_1	2	3	1	1	1	2	1	2
D_2	1	1	1	1	3	1	2	1
D_3	3	4	2	3	2	2	3	2
D_4	2	2	2	2	2	1	2	3
D_5	1	1	1	2	1	1	1	2

Table 2.2: Relevant symptoms for all diseases in Example 2.2 for $a = 1$

Disease	Relevant symptoms
D_1	$\{2\}$
D_2	$\{5\}$
D_3	$\{2\}$
D_4	$\{2\}$
D_5	$\{2, 5\}$

In other words, we find the minimal subset of $\mathcal{S}_{ai} \subseteq \mathcal{S}$ such that disease i has different symptoms when compared with all other diseases. This subset is called the *set of relevant symptoms* for disease i .

Example 2.2 (Identifying relevant symptoms). Assume that we have the set of diseases $\mathcal{D} = \{D_1, D_2, D_3, D_4, D_5\}$ and the set of symptoms $\mathcal{S} = \{S_1, S_2, \dots, S_8\}$. Assume also that the symptoms associated with the different diseases are those listed in Table 2.1.

Then, minimizing the sum $Z = \sum_{j=1}^m x_j$ subject to (2.3), and two values of a , we conclude that the set of symptoms $\{2, 5\}$ is a minimal sufficient set of symptoms able to distinguish the 5 diseases. However, if we use a discrepancy level of $a = 3$, the required set is $\{1, 2, 4, 5, 7\}$. Note that in this case we can have two missing symptoms and the diagnostic would be still correct.

Finally, Table 2.2 shows the required set of relevant symptoms for each disease and $a = 1$. Note in Table 2.1 that symptom 2 is sufficient to identify diseases D_1, D_3 , and D_4 , and that symptom 5 is sufficient to identify disease D_2 . However, we need symptoms 2 and 5 to identify disease D_5 . ■

2.4 The Academy Problem

The Academy of Engineering has m members and is involved in the process of selecting r new members among a set of J candidates. To this end, each

actual member is allowed to support from a minimum of 0 to a maximum of r candidates. The r candidates with the largest number of supports are incorporated to the academy.

Before the final selection process, a previous test is performed to know the degree of support of each candidate. In this process each actual member can assign the scores in the list \mathbf{p} to a maximum of S candidates, but need not to assign all scores.

Only the sum of scores of each candidate is known. The problem consists of knowing the minimum and maximum number of final supports of each candidate based on the results of the test, assuming that assigning a score to one candidate is equivalent to supporting such candidate by the actual member assigning the score.

The problem has the following elements:

1. Data

I : the actual number of members in the Academy of Engineering

J : the number of candidates

S : the number of different scores that can be assigned

p_s : the s -th score

C_j : the total score associated with candidate j

2. Variables

x_{ijs} : a binary variable that takes value 1 if member i assigns score p_s to candidate j ; otherwise, it takes value 0

3. Constraints

- Each member can assign at the most one score to each candidate:

$$\sum_{s=1}^S x_{ijs} \leq 1; \forall i \in \{1, 2, \dots, I\}, j \in \{1, 2, \dots, J\}$$

- Each member can assign score p_s to at most one candidate:

$$\sum_{j=1}^J x_{ijs} \leq 1; \forall i \in \{1, 2, \dots, I\}, s \in \{1, 2, \dots, S\}$$

- The total score obtained by each candidate must be the given value:

$$\sum_{i=1}^I \sum_{s=1}^S p_s x_{ijs} = C_j; \forall j \in \{1, 2, \dots, J\}$$

Table 2.3: Total scores received by the 8 candidates in Example 2.3

Candidate	1	2	3	4	5	6	7	8
Received score	71	14	139	13	137	18	24	8

4. **Function to be optimized.** The objective of this problem consists of minimizing and maximizing this function for each candidate:

$$Z_j = \sum_{i=1}^I \sum_{s=1}^S x_{ijs}, \quad j \in \{1, 2, \dots, J\} \quad (2.6)$$

Example 2.3 (Academy problem). Assume that the Academy of Engineering has 20 members and that $r = 4$ new members are to be selected among $J = 8$ candidates, and that $\mathbf{p} \equiv \{10, 8, 3, 1\}$, which implies $S = 4$.

The available information consists of the last row in Table 2.3, namely, the total scores received by each candidate in the first round, and we look for the number of supports of each candidate (see the second last row in Table 2.4).

The actual scores received by each candidate from each actual member are those given in Table 2.4 (note that this information is not available, but has been given only for illustration).

If we minimize and maximize (2.6) for all candidates, we get the results shown in Table 2.5. The following conclusions can be drawn from this table:

1. Only candidates 3 and 5 have at least 15 guaranteed supports. Note that the next one, candidate 1, has only 8 guaranteed supports.
2. It is not clear from Table 2.5 that candidates 3 and 5 enter the academy, since candidates 6, 1 and 7 have a maximum of 18, 20, and 20 guaranteed supports, and they can get only 15, 16, or 17.
3. To know, before the final election, whether candidate 3 enters the academy, it is necessary to add new constraints to the problem. For example, adding that the total number of supports of candidates 1, 5, 6, and 7 are larger than the total number of supports of candidate 3:

$$\begin{aligned} \sum_{i=1}^I \sum_{s=1}^S x_{i1s} &\geq \sum_{i=1}^I \sum_{s=1}^S x_{i3s} \\ \sum_{i=1}^I \sum_{s=1}^S x_{i5s} &\geq \sum_{i=1}^I \sum_{s=1}^S x_{i3s} \\ \sum_{i=1}^I \sum_{s=1}^S x_{i6s} &\geq \sum_{i=1}^I \sum_{s=1}^S x_{i3s} \\ \sum_{i=1}^I \sum_{s=1}^S x_{i7s} &\geq \sum_{i=1}^I \sum_{s=1}^S x_{i3s} \end{aligned}$$

Table 2.4: Scores received by the 8 candidates in Example 2.3

Actual member	Candidate							
	1	2	3	4	5	6	7	8
1	3	-	10	-	8	1	-	-
2	1	-	10	-	8	3	-	-
3	-	1	-	3	10	-	8	-
4	-	3	10	-	8	1	-	-
5	3	-	8	-	10	-	1	-
6	1	-	10	-	8	-	3	-
7	10	-	8	-	3	1	-	-
8	3	-	10	1	8	-	-	-
9	8	-	3	-	10	1	-	-
10	-	3	10	-	1	-	8	-
11	8	-	1	-	10	-	3	-
12	-	-	-	-	10	-	-	-
13	-	-	10	-	8	-	-	-
14	10	-	-	1	3	-	-	8
15	3	-	10	-	8	-	1	-
16	10	-	1	-	8	-	3	-
17	1	3	10	8	-	-	-	-
18	1	3	8	-	10	-	-	-
19	1	-	10	-	3	8	-	-
20	8	1	10	-	3	-	-	-
Number of supports	15	6	17	4	19	7	6	1
Total score	71	14	139	13	137	18	24	8

Table 2.5: Actual and bounds for the number of supports for the 8 candidates in Example 2.3

Supports	Candidate							
	1	2	3	4	5	6	7	8
Minimum	8	3	15	2	15	2	3	1
Maximum	20	14	20	13	20	18	20	8
Actual	15	6	17	4	19	7	6	1
Scores	71	14	139	13	137	18	24	8

Since this leads to an unfeasible problem, then we can guarantee that candidate 3 enters the Academy of Engineering.

■

2.5 School Timetable Problem

This example is a simple instance of the “school timetable problem”. It consists of allocating classrooms and teaching hours for the subjects of one academic program divided in blocks.

We assume that n_c classrooms and n_h teaching hours are available, respectively, to teach n_s subjects. These subjects are grouped by (1) academic blocks and (2) instructors. Binary variable $v(s, c, h)$ is equal to 1 if subject s is taught in classroom c at hour h , and 0 otherwise.

We denote the set of all subjects, by Ω , the set of the n_i subjects taught by instructor i , by Ω_i , and the set of the n_b subjects grouped in academic block b , by Δ_b . Indices s, c, h, i , and b indicate respectively subject, classroom, hour, instructor and block.

The problem has the following elements:

1. Data

n_c : the number of classrooms

n_h : the number of available teaching hours

n_s : the number of subjects

n_i : the number of subjects taught by instructor i

n_b : the number of academic blocks

Ω : the set of all subjects to be taught

Ω_i : the set of subjects taught by instructor i

Δ_b : the set of subjects belonging to academic block b

2. Variables

$v(s, c, h)$: a binary variable that takes value 1 if subject s is taught in classroom c at hour h , and 0 otherwise

3. Constraints

(a) Every instructor teaches all his/her subjects:

$$\sum_{s \in \Omega_i} \sum_{c=1}^{n_c} \sum_{h=1}^{n_h} v(s, c, h) = n_i, \quad \forall i \quad (2.7)$$

(b) Every instructor teaches at most 1 subject every hour:

$$\sum_{s \in \Omega_i} \sum_{c=1}^{n_c} v(s, c, h) \leq 1, \quad \forall h, \quad \forall i \quad (2.8)$$

(c) Every subject is taught once:

$$\sum_{c=1}^{n_c} \sum_{h=1}^{n_h} v(s, c, h) = 1, \quad \forall s \quad (2.9)$$

(d) In every classroom-hour combination at most 1 subject is taught:

$$\sum_{s \in \Omega} v(s, c, h) \leq 1, \quad \forall c, \quad \forall h \quad (2.10)$$

(e) At every hour, at most 1 subject of any academic block is taught:

$$\sum_{s \in \Delta_b} \sum_{c=1}^{n_c} v(s, c, h) \leq 1, \quad \forall h, \quad \forall b \quad (2.11)$$

4. **Function to be optimized.** Formulating an appropriate objective function to be minimized is not an easy task. However, in this example we consider a very simple objective function. The target is to produce a compact timetable. It can be formulated by minimizing

$$\sum_{s \in \Omega} \sum_{c=1}^{n_c} \sum_{h=1}^{n_h} (c + h) v(s, c, h)$$

subject to constraints (2.7)–(2.10).

This optimization function has been chosen because it penalizes the $v(s, c, h)$ variables taking on value 1 for high values of c and h . Thus, it tries to compact the teaching classrooms and hours. The smallest the classroom number and the hour the better.

Example 2.4 (School timetable problem). Consider 3 classrooms, 5 teaching hours, 8 subjects, 2 instructors, and 2 course blocks. The set of all subjects is $\Omega = \{s_1, s_2, \dots, s_8\}$, the set of subjects of instructor 1 is $\Omega_1 = \{s_1, s_2, s_8\}$, the set of subjects of instructor 2 is $\Omega_2 = \{s_3, s_4, s_5, s_6, s_7\}$, the set of subjects of academic block 1 is $\Delta_1 = \{s_1, s_2, s_3, s_4\}$, and the subjects of academic block 2 is $\Delta_2 = \{s_5, s_6, s_7, s_8\}$. Note that $\Omega_1 \cup \Omega_2 = \Omega$ and $\Omega_1 \cap \Omega_2 = \emptyset$, and $\Delta_1 \cup \Delta_2 = \Omega$ and $\Delta_1 \cap \Delta_2 = \emptyset$.

The solution is provided in the tables below:

	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
$c = 1$	s_7	s_6	s_3	s_4	s_5
$c = 2$	s_2	s_1	s_8	–	–
$c = 3$	–	–	–	–	–

The schedule for instructor 1 is

	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
$c = 1$	–	–	–	–	–
$c = 2$	s_2	s_1	s_8	–	–
$c = 3$	–	–	–	–	–

The schedule for instructor 2 is

	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
$c = 1$	s_7	s_6	s_3	s_4	s_5
$c = 2$	–	–	–	–	–
$c = 3$	–	–	–	–	–

The schedule for academic block 1 is

	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
$c = 1$	–	–	s_3	s_4	–
$c = 2$	s_2	s_1	–	–	–
$c = 3$	–	–	–	–	–

The schedule for academic block 2 is

	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$
$c = 1$	s_7	s_6	–	–	s_5
$c = 2$	–	–	s_8	–	–
$c = 3$	–	–	–	–	–

■

2.6 Models of Discrete Location

In this example we describe one of the *discrete location models* or, more precisely, the *capacitated facility location problem*. This deals with deciding where to locate facilities within a finite set of sites, taking into account the needs of the clients to be served, and optimizing certain economic criteria. Usually, setting up a facility involves significant costs that do not depend on the production level of the facility.

The problem is motivated by a number of potential applications. For example, several plants are to be set up at some points of a transportation system to maximize the benefit by means of minimizing the production and shipment costs. Figure 2.1 shows a solution to the problem of locating plants that provide service to a set of clients.

Thus, the main elements in this problem are

1. Data

I : a set $\{1, \dots, n\}$ of n clients

J : a set $\{1, \dots, m\}$ of m sites where facilities can be located

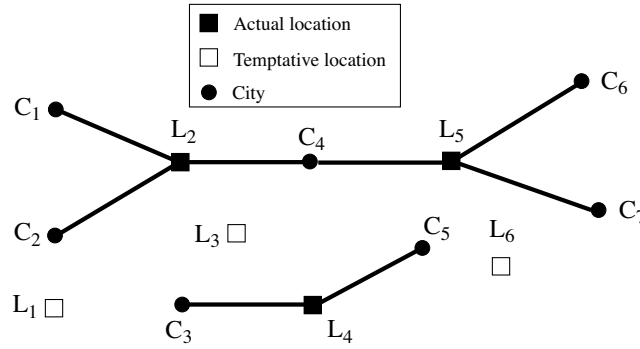


Figure 2.1: Solution of the example of the capacitated facility location problem.

f_j : the fixed cost of the opening facility placed at j for $j \in J$

c_{ij} : the profit per unit of sale of the goods originated at facility j to client i . Usually, the c_{ij} depend on the production costs at facility j , the demand and selling price for client i , and the transportation costs between client i and facility j

u_j : the capacity of the facility located at j

b_i : the demand of the i th client

2. **Variables.** The variables involved in this problem are the following:

y_j : a binary variable to model the choice of “opening” a facility at the site j . This is defined as follows:

$$y_j = \begin{cases} 1 & \text{if facility } j \text{ is open} \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

x_{ij} : the quantity of commodities sent from facility j to client i .

3. **Constraints.** The constraints in this problem are as follows. Each client’s demand must be satisfied:

$$\sum_{j \in J} x_{ij} = b_i, \quad \forall i \in I \quad (2.13)$$

Since a client i cannot be served from j unless a facility is placed at j , we have the following constraints:

$$\sum_{i \in I} x_{ij} \leq u_j y_j, \quad \forall j \in J \quad (2.14)$$

These linear inequalities take into account that the client i can be served from j only if a facility is located at node j , since $y_j = 0$ implies that

Table 2.6: City demands

City	Demand
C_1	1.5
C_2	2.0
C_3	3.0
C_4	4.0
C_5	2.5
C_6	1.0
C_7	2.0

$x_{ij} = 0, \forall i$ and $y_j = 1$ yields the constraint $\sum_{i \in I} x_{ij} \leq u_j$, which means that the production level of the facility j cannot exceed its capacity. In addition, the variable constraints are

$$y_j \in \{0, 1\}, \forall j \in J \quad (2.15)$$

$$x_{ij} \geq 0 \forall i \in I, \forall j \in J \quad (2.16)$$

4. **Function to be optimized.** In the so-called strong formulation of the uncapacitated facility location problem we maximize

$$Z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} - \sum_{j \in J} f_j y_j \quad (2.17)$$

In this model, the allocation problem in the case of unlimited capacity of the sites is solved easily. In reality, in the presence of a feasible set of locations, the allocation problem is solved by means of assigning each client to the most profitable open facility. However, it may be unrealistic to assume that a facility can supply any number of clients. Thus, limited capacities must be dealt with.

Example 2.5 (Location of industrial plants). A company wishes to build various industrial plants to supply 7 cities with a certain product. The demand of these cities based on demographic factors and social characteristics is estimated. These values are shown in Table 2.6.

A study has indicated 6 possible sites for these industrial plants. It is supposed that all the plants have the same characteristics. The maximum production capacity per plant is 6 units. The cost of investment recovering has been calculated in 10 monetary units for the period of study.

Table 2.7 shows the profit achieved by selling, to city i , one unit manufactured in one plant located at site j .

The decisionmaker needs to determine the amount of plants and their locations, so that the cities demand is satisfied, and locations are such that the demands are satisfied, and a maximum financial benefit is obtained. The optimization problem associated with this decision process consists of maximizing

Table 2.7: Benefits according to different locations

Locations (L_j)	Cities (C_i)						
	C_1	C_2	C_3	C_4	C_5	C_6	C_7
L_1	4.0	4.5	2.5	0.5	1.0	0.5	-3.5
L_2	4.0	4.5	2.5	4.2	3.5	1.5	-0.5
L_3	3.5	5.0	4.0	3.5	4.5	1.5	0.0
L_4	1.3	3.0	5.0	3.3	5.5	1.8	1.3
L_5	0.5	1.0	1.5	5.0	4.0	5.5	3.0
L_6	-1.0	0.0	1.5	3.3	4.0	4.5	2.0

the total benefit including amortization costs, subject to the constraints. So, the problem can be stated as follows, by maximizing

$$Z = \sum_{i=1}^7 \sum_{j=1}^6 c_{ij} x_{ij} - \sum_{j=1}^6 10y_j$$

subject to

$$\begin{aligned} \sum_{j=1}^6 x_{1j} &= 1.5; & \sum_{j=1}^6 x_{2j} &= 2.0 \\ \sum_{j=1}^6 x_{3j} &= 3.0; & \sum_{j=1}^6 x_{4j} &= 4.0 \end{aligned} \tag{2.18}$$

$$\sum_{j=1}^6 x_{5j} = 2.5; \quad \sum_{j=1}^6 x_{6j} = 1.0$$

$$\sum_{j=1}^6 x_{7j} = 2.0$$

and

$$\sum_{i=1}^6 x_{ij} \leq 6y_j; \quad j = 1, \dots, 7 \tag{2.19}$$

$$y_j \in \{0, 1\}; \quad j = 1, \dots, 6 \tag{2.20}$$

$$x_{ij} \geq 0; \quad i = 1, \dots, 7; \quad j = 1, \dots, 6$$

where (2.18) and (2.19) are the demand and the production capacity constraints, respectively.

The solution of this problem is plotted in Figure 2.1 consists of placing 3 industrial plants in locations L_2 , L_4 , and L_3 , and the production distribution by cities is the Table 2.8. ■

Table 2.8: Amount of production of each operating plant to be provided to each city

Locations	Cities						
	C_1	C_2	C_3	C_4	C_5	C_6	C_7
L_2	1.5	2.0		1.0			
L_4			3.0		2.5		
L_5				3.0		1.0	2.0

2.7 Unit Commitment of Thermal Power Units

The cost of starting up an electric power thermal unit after being offline for a couple of days is approximately half the cost of buying a 100 m² apartment in a distinguished neighborhood. Therefore, the planning of the startups and shutdowns of any thermal unit should be done carefully. The electric power thermal unit commitment problem consists of determining, for a planning horizon, the startup and shutdown schedule of every unit so that the electric demand is served and total operating costs are minimized, while satisfying different technical and security constraints.

A typical planning horizon is one day divided in hours. If time intervals are denoted by k , the planning horizon consists of the periods

$$k = 1, 2, \dots, K \quad (2.21)$$

where K is typically equal to 24.

The startup cost is an exponential function of the time the unit has been offline, but it will be considered constant (this is a reasonable simplification in most cases). Every time a unit is started up, its startup cost is incurred, and this can be expressed as

$$C_j y_{jk} \quad (2.22)$$

where C_j is the startup cost of unit j and y_{jk} is a binary variable that is equal to 1 if unit j is started up at the beginning of period k and 0, otherwise.

The shutdown cost can be expressed in a similar fashion as the startup cost; thus

$$E_j z_{jk} \quad (2.23)$$

where E_j is the shutdown cost of unit j and z_{jk} a binary variable that is equal to 1 if unit j is shut down at the beginning of period k , and 0 otherwise.

The running costs consist of a fixed cost and a variable cost. The fixed cost can be expressed as

$$A_j v_{jk}, \quad (2.24)$$

where A_j is the fixed cost of unit j and v_{jk} is a binary variable that is equal to 1 if unit j is online during period k and 0, otherwise.

The variable cost can be considered proportional to the unit output power:¹

$$B_j p_{jk} \quad (2.25)$$

where B_j is the variable cost of unit j and p_{jk} the output power of unit j during period k .

Thermal units cannot operate below a minimum output power and above a maximum output power. These technical constraints can be expressed as

$$\underline{P}_j v_{jk} \leq p_{jk} \leq \overline{P}_j v_{jk} \quad (2.26)$$

where \underline{P}_j and \overline{P}_j are respectively the minimum and maximum output powers of unit j .

The left-hand side of the preceding constraint expresses that if unit j is online during period k ($v_{jk} = 1$), its output power should be above the minimum output power. Analogously, the right-hand-side of the constraint above expresses that if unit j is online during period k ($v_{jk} = 1$), its output power should be below the maximum output power. If $v_{jk} = 0$, the preceding constraint forces $p_{jk} = 0$.

From one time period to the next one, any power unit cannot increase its output power above a maximum power increment, called the rampup limit. This can be written as

$$p_{jk+1} - p_{jk} \leq S_j \quad (2.27)$$

where S_j is the maximum rampup power increment of unit j .

For the first period of the planning horizon the above constraint becomes

$$p_{j1} - P_j^0 \leq S_j \quad (2.28)$$

where P_j^0 is the output power of unit j just before the first period of the planning horizon.

Similarly, any power unit cannot decrease its output power above a maximum power decrement, which is called the rampdown power limit. Therefore

$$p_{jk} - p_{jk+1} \leq T_j \quad (2.29)$$

where T_j is the maximum rampdown power decrement of unit j .

For the first period of the time horizon, the constraint above becomes

$$P_j^0 - p_{j1} \leq T_j \quad (2.30)$$

Any unit that is online can be shut down but not started up, and analogously, any unit that is offline can be started up but not shut down. This can be expressed as

¹A more precise modeling requires the variable cost to be a quadratic or cubic function of the power output.

$$y_{jk} - z_{jk} = v_{jk} - v_{jk-1} \quad (2.31)$$

For the first period the above constraint becomes

$$y_{j1} - z_{j1} = v_{j1} - V_j^0 \quad (2.32)$$

where V_j^0 is a binary constant that is equal to 1 if unit j is online the period preceding the first period of the planning horizon, and 0 otherwise. The reader is encouraged to verify these two conditions using examples.

In every period the power demand should be satisfied, so

$$\sum_{j=1}^J p_{jk} = D_k \quad (2.33)$$

where J is the number of power units and D_k the demand in period k .

For security reasons, the total output power available online should be larger than the actual demand by a specified amount. This is formulated as

$$\sum_{j=1}^J \bar{P}_j v_{jk} \geq D_k + R_k \quad (2.34)$$

where R_k is the amount of required reserve (over the demand) in period k .

The main elements in this problem are:

1. Data

K : the number of time intervals

C_j : the startup cost of unit j

E_j : the shutdown cost of unit j

A_j : the fixed cost of unit j

B_j : the variable cost of unit j

\underline{P}_j : the minimum output power of unit j

\bar{P}_j : the maximum output power of unit j

S_j : the maximum rampup power increment of unit j

P_j^0 : the output power of unit j just before the first period of the planning horizon

T_j : the maximum rampdown power decrement of unit j

V_j^0 : a binary constant that is equal to 1; if unit j is online the period preceding the first period of the planning horizon, and 0, otherwise

J : the number of power units

D_k : the demand in period k

R_k : the amount of required reserve (over the demand) in period k

2. **Variables.** The variables involved in this problem are the following:

y_{jk} : a binary variable that is equal to 1, if unit j is started up at the beginning of period k and 0, otherwise

z_{jk} : a binary variable that is equal to 1, if unit j is shut down at the beginning of period k , and 0, otherwise

v_{jk} : a binary variable that is equal to 1, if unit j is online during period k and 0, otherwise

p_{jk} : the output power of unit j during period k

3. **Constraints.** The constraints in this problem are as follows. Any unit at any time should operate above its minimum output power and below its maximum output power, then

$$\underline{P}_j v_{jk} \leq p_{jk} \leq \bar{P}_j v_{jk} \quad \forall j, k \quad (2.35)$$

Rampup constraints should be satisfied:

$$p_{jk+1} - p_{jk} \leq S_j, \quad \forall j, k = 0, \dots, K-1 \quad (2.36)$$

where

$$p_{j0} = P_j^0$$

Rampdown constraints should also be satisfied:

$$p_{jk} - p_{jk+1} \leq T_j, \quad \forall j, k = 0, \dots, K-1 \quad (2.37)$$

The logic of status changes (from online to offline and vice versa) should be preserved; therefore

$$y_{jk} - z_{jk} = v_{jk} - v_{jk-1}, \quad \forall j, k = 1, \dots, K \quad (2.38)$$

where

$$v_{j0} = V_j^0, \quad \forall j$$

The demand should be satisfied in every period; thus

$$\sum_{j=1}^J p_{jk} = D_k, \quad \forall k \quad (2.39)$$

Finally, security constraints should be satisfied in all periods of the planning horizon; then

$$\sum_{j=1}^J \bar{P}_j v_{jk} \geq D_k + R_k, \quad \forall k. \quad (2.40)$$

4. **Function to be minimized.** The objective of the unit commitment problem is to minimize total costs; the objective is therefore to minimize

$$Z = \sum_{k=1}^K \sum_{j=1}^J [A_j v_{jk} + B_j p_{jk} + C_j y_{jk} + E_j z_{jk}] \quad (2.41)$$

The problem illustrated in (2.35)–(2.41) is a simplified version of the electric power thermal unit commitment problem. It should be noted that it is a binary mixed-integer linear programming problem.

Example 2.6 (Unit commitment). A 3-hour planning horizon is considered. The demands in these hours are respectively 150, 500, and 400. Reserves are respectively 15, 50, and 40. Three power units are considered. Data for these units are given below:

Power unit number	1	2	3
Maximum output power	350	200	140
Minimum output power	50	80	40
Rampup limit	200	100	100
Rampdown limit	300	150	100
Fixed cost	5	7	6
Startup cost	20	18	5
Shutdown cost	0.5	0.3	1.0
Variable cost	0.100	0.125	0.150

All units are offline before the planning horizon. The unit output powers for the optimal solution are

	Hour		
Unit	1	2	3
1	150	350	320
2	—	100	080
3	—	050	—
Total	150	500	400

The minimum cost is 191. Unit 1 is started up at the beginning of hour 1 and remains online for the 3 hours. Unit 2 is started up at the beginning of hour 2 and remains on line during hours 2 and 3. Unit 3 is started up at the beginning of hour 2 and shut down at the beginning of hour 3. ■

Exercises

- 2.1 Walter builds two types of transformers and has available 6 tons of ferromagnetic material and 28 hours of working time. Transformer 1 requires 2

tons of ferromagnetic material and 7 hours of work, and transformer 2 requires 1 unit of ferromagnetic material and 8 hours of work. Selling prices of transformers 1 and 2 are respectively 120 and 80 thousand Eurodollars. How many transformers of each type should manufacture Walter to maximize his benefits? Solve the problem graphically and analytically.

- 2.2 Consider a transportation network where several cities are connected by roads. The problem becomes one of finding the shortest route between two cities. We assume that the distance between two directly connected cities is known. Formulate this problem using integer linear programming (*The Shortest Path Problem*).
- 2.3 Consider a salesperson who wants to find the minimum cost tour that visits each of n given cities exactly once and returns to the originating city. Formulate this problem as an integer linear programming problem (*The Traveling Salesperson Problem*).
- 2.4 Consider the problem of finding the maximum number of paths in a communication network from the origin p to destination q such that they are link-disjoint. That means that two different paths do not have common links. Formulate this problem as an integer linear programming problem.
- 2.5 Given a connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where N is the set of nodes, A is the set of links, and a special set of “terminal vertices” $\mathcal{T} = \{t_1, t_2, \dots, t_k\} \subset \mathcal{N}$, the network reliability problem consists of evaluating the probability of communication between the elements of \mathcal{T} under a random degradation of the network. Let $p(e)$ be the probability of link e to be *operative*. The network reliability is the probability of all pairs of terminals being connected at least by an operative path. Consider the problem that one has to design the least-cost network that satisfies the requirement of a given reliability. This design consists of the selection of a subset of links of the original network. More specifically, consider as the original network the one shown in Figure 2.2 with terminal nodes $T = \{1, 2\}$ and assume that the desired network reliability is 0.90. The reliability and cost of the links are given in Table 2.9. Formulate the corresponding problem as an integer linear programming problem.
- 2.6 An electricity producer should plan its hourly energy production to maximize its profits from selling energy during a planning horizon of a given number of hours. Formulate a mixed-integer linear programming problem taking into account that:
 - (a) The producer does not produce before the planning horizon.
 - (b) Hourly energy prices can be forecasted and are considered known.
 - (c) If running, the minimum and maximum energy productions of the producer are known quantities, and the minimum is greater than zero.

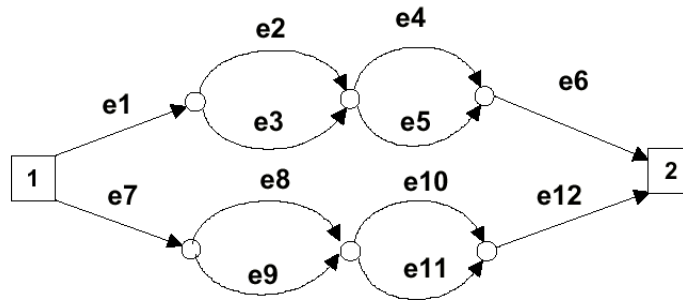


Figure 2.2: Network topology.

Table 2.9: Reliability and cost of the components of the network of Figure 2.2

Link	Reliability ($p(e_i)$)	Cost
e_1	0.90	1.90
e_2	0.85	1.85
e_3	0.95	1.95
e_4	0.70	1.70
e_5	0.80	1.80
e_6	0.90	1.90
e_7	0.90	1.90
e_8	0.50	1.35
e_9	0.60	1.45
e_{10}	0.60	1.20
e_{11}	0.30	1.30
e_{12}	0.90	1.90

- (d) Energy productions in two consecutive hours cannot differ in more than a prespecified amount.
- (e) Producer production cost is linear.

2.7 The manufacture of the two parts, A and B , of a certain machine requires the processes L , S , D , M , and G . The time of each process to operate on each part and the number of available processes are given in Table 2.10 (hours per unity). Each one can be used during 8 hours, 30 days per month.

- (a) Determine the optimal production strategy to maximize the total number of parts A and B manufactured in a month.
- (b) If the number of parts A must be equal to the number of parts B , what is the optimal strategy?

Table 2.10: Times required for different processes

Part	Process				
	<i>L</i>	<i>S</i>	<i>D</i>	<i>M</i>	<i>G</i>
<i>A</i>	0.6	0.4	0.1	0.5	0.2
<i>B</i>	0.9	0.1	0.2	0.3	0.3
Availability	10	3	4	6	5

2.8 A hospital manager should plan the working timetable for the hospital staff. Determine the minimum weekly cost associated with the staff of this hospital if

- (a) The daily working time is structured in 3 shifts.
- (b) In every shift there should be at least 1 physician, 2 (male) nurses, and 3 assistants.
- (c) The maximum total number of employees needed on every shift is 10.
- (d) The salaries are: \$50/shift for a physician, \$20/shift for a nurse, and \$10/shift for an assistant.
- (e) The total number of employees is: 15 physicians, 36 nurses, and 49 assistants.
- (f) Each employee should rest during at least two consecutive shifts.