M. M. Baldi – R. Tadei

Problem Solving through Linear Programming

Contents

1

Introduction to Linear Programming

2

Advanced modeling I: linearizable objective functions

3

Advanced modeling II: logical constraints

3.1 Introduction

A *proposition* is any statement or sentence which, in Logic, may be true or false. We use capital letters to indicate propositions.

Example 1. *These are some examples of propositions.*

A *Life is beautiful*

B *Today I am happy*

C *Rome is in Italy.*

♣

Note that, in principle, propositions might not be objectively true or have sense.

Example 2. *Propositions like*

D *New York is in Italy*

E *Florence is in Portugal*

F *The sun is a mushroom*

are clearly false.

♣

LOGICAL CONNECTIVE	CORRESPONDING SYMBOL
Or	$A \vee B$
And	$A \wedge B$
Nand	$A \uparrow B$
Xor	$A \oplus B$
Not	$\bar{A}$
Implies (If A then B)	$A \longrightarrow B$
If and only if	$A \longleftrightarrow B$

Table 3.1: Connectives and their symbols

Nevertheless, although prepositions like those in Example 2 are not true, this does not mean that they are not propositions. Propositions met so far are simple. In fact, they are called *atomic statements*. Propositions can be linked together by means of the so-called *connectives*. Table 3.1 shows some connectives with their corresponding symbols.

Example 3 shows some applications of connectives to propositions.

Example 3. *Combining the propositions in Example 1 and 2 by means of connectives we can get many resulting propositions like the following ones.*

- $A \wedge B$*: Life is beautiful* and *today I am happy*
- $\bar{E}$*: Florence is* not *in Portugal*
- $A \longrightarrow B$*:* If *life is beautiful,* then *today I am happy.*

♣

Propositions in Example 3 are longer than the original atomic statements of Examples 1 and 2. Indeed, these propositions are named *compound statements* because they are a combination of atomic statements through connectives.

It is also possible to associate the so-called truth tables to connectives. Table 3.2 shows the truth tables of the connectives presented in Table 3.1.

A	B	$A \vee B$	$A \wedge B$	$A \uparrow B$	$A \oplus B$	$\bar{A}$	$A \longrightarrow B$	$A \longleftrightarrow B$
0	0	0	0	1	0	1	1	1
0	1	1	0	1	1	1	1	0
1	0	1	0	1	1	0	0	0
1	1	1	1	0	0	0	1	1

Table 3.2: Truth table of some connectives

Although we have introduced several connectives, it would have been enough just to present two of them: $\wedge$ and $^{-}$. This means that the set $\{\wedge, ^{-}\}$ is a complete set

of connectives. All the remaining connectives can be expressed by the elements of this complete set through the De Morgan's laws:

$$\overline{A \vee B} = \bar{A} \wedge \bar{B} \tag{3.1}$$
$$\overline{A \wedge B} = \bar{A} \vee \bar{B}. \tag{3.2}$$

This briefing in Logic allows us to introduce logical constraints which compare in a number of problems. Some examples of these constraints are reported in Example 4.

Example 4.
- $P1$*: at least 5 nodes must be opened in a telecommunication network, provided that some flow flows through them*
- $P2$*: if product* A *is manufactured, then products* B *and* C *cannot be manufactured*
- $P3$*: I take the airplane if I take the train or the bus.*

♣

Logical constraints have a double nature: by one side we have to deal with *logical* propositions, while by another side we have to deal with a *linear* constraints. Therefore, logical constraints theory represents the art of translating a logical proposition into a linear constraints that completely describes it. The first step of this translation consists of associating boolean variables to a statement.

Example 5. *Consider the following proposition:*

A*: I take the train*

Then, we can introduce the following logical variable:

$$y_A = \begin{cases} 1 & \textit{if I take the train} \\ 0 & \textit{if I do not take the train.} \end{cases}$$

♣

The logical variable y_A is also called *indicator variable*.

At this point, we want to point out that logical constraints are different from Boolean algebra expressions. In fact, as stated before, the goal of logical constraints theory (and also the aim of this chapter!) consists of expressing logical propositions into *linear* constraints. Boolean algebra expressions might not be linear, as shown in the next Example.

Example 6. *Consider proposition* $P4$ *in Example 4:*

$P4$*: I take the airplane if and only if I take the train and the bus.*

Then, according to Example 5, in addition to variable y_A, we can also introduce the following indicator variables:

$$y_B = \begin{cases} 1 & \textit{if I take the bus} \\ 0 & \textit{if I do not take the bus,} \end{cases} \qquad y_C = \begin{cases} 1 & \textit{if I take the train} \\ 0 & \textit{if I do not take the train} \end{cases}$$

Proposition $P4$ can be expressed in Boolean algebra like

$$A = B \wedge C, \tag{3.3}$$

which, using the indicator variables y_i already introduced, becomes

$$y_A = y_B \cdot y_C. \tag{3.4}$$

This boolean expression is correct but, unfortunately, it is not linear.

♣

Since we are dealing with linear programs, we have to find another way to translate logical constraints into linear constraints. This will be done in the next sections.

3.2 Indicator variables

We have concluded the last section by introducing indicator variables. Indicator (or boolean or logical) variables are usually associated to propositions. If the proposition is true then the indicator variables assumes value 1 (corresponding to the true value), otherwise the proposition is false and the indicator variables goes to 0 (the false value). We would like to point out a different notation that can be met in several books. Some authors tend to use the letter y for the indicator variables. Sometimes the z letter can be found. Other authors use the Greek letter δ. These indicator variables also appear in several articles in the literature with either letters. Therefore, in the following, we will independently use both the notations.

Again, the difficulty consists of translating the logical proposition into a *linear* constraint. Perhaps the simplest case we are asked to represent is to associate an indicator variable, say y, to the proposition $x > 0$. Therefore, we can explicitly define the indicator variable y as follows.

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0. \end{cases} \tag{3.5}$$

Note that (3.5) can be split in two implications:

$$\text{if } x > 0, \text{ then } y = 1, \tag{3.6}$$

$$\text{if } x = 0, \text{ then } y = 0. \tag{3.7}$$

In turn, (3.6) and (3.7) can also be stated respectively as follows.

$$x > 0 \quad \longrightarrow \quad y = 1 \tag{3.8}$$
$$x = 0 \quad \longrightarrow \quad y = 0. \tag{3.9}$$

Implication (3.6) or (3.8) can be expressed as follows.

$$x \leq My, \tag{3.10}$$

where M is an upper bound for variable x. To prove that constraint (3.10) describes implication (3.6), note that when variable x assumes any positive value, say θ, then (3.10) becomes

$$\theta \leq My. \tag{3.11}$$

As M is definitively greater than 0, we can divide each side of (3.11) by M and get

$$y \geq \frac{\theta}{M}. \tag{3.12}$$

But θ is positive and this implies that $\frac{\theta}{M}$ is also positive. If $\frac{\theta}{M}$ is positive then (3.12) forces y to be positive. But, being y a binary variable, the only positive value it can take is 1 and the implication is well described by (3.10). We also wish to point out that M has to be an upper bound of variable x. If this is not the case, then (3.12) can force y to assume values greater than 1. But this is clearly infeasible, as y is a boolean variable. Note that constraint (3.10) covers only one direction of relation (3.5) i. e., implication (3.6) (or (3.8)). In fact, as we have just proved, when $x > 0$ then y is set to 1. But when $x = 0$, (3.10) becomes $0 \leq My$ which is equivalent to $y \geq 0$. Therefore y is free to take either value 0 or value 1, but we want to ensure y to be 0 when $x = 0$. Sometimes, as in the so called *fixed charge problems*, constraint (3.10) is enough to model a problem because the reverse implication (3.7) (or (3.9)) is implicitly satisfied. Other times constraint (3.10) is not enough and we explicitly have to express the reverse implication (3.7). To do so, we need a new tool called the *contrapositive* and that we thoroughly will use in this chapter. The contrapositive is an equivalence which allows us to transform an implication. In particular, given two propositions A and B and the implication $A \longrightarrow B$, the contrapositive affirms that this relationship can be transformed as follows.

$$A \longrightarrow B \equiv \bar{B} \longrightarrow \bar{A}, \tag{3.13}$$

Where $\equiv$ is the equivalence symbol. This means that writing $A \longrightarrow B$ is the same of writing $\bar{B} \longrightarrow \bar{A}$. Example 7 gives an intuitive confirmation of this equivalence.

Example 7. *Consider the following propositions:*

- A: *It rains*
- B: *I take the umbrella.*

Then, $A \longrightarrow B$ *is read as*

$$\textit{If it rains, then I take the umbrella.} \tag{3.14}$$

In turn, we read $\bar{B} \longrightarrow \bar{A}$ *as*

$$\textit{If I don't take the umbrella, then it does not rain.} \tag{3.15}$$

Intuitively, (3.15) *is equivalent to* (3.14) *because, by* (3.14)*, I have to take the umbrella if it rains. Therefore, if I don't have any umbrella with me then for sure it does not rain. Note that both* (3.14) *and* (3.15) *are just one implication. That means that I might take the umbrella even if it does not rain. Otherwise, we should have the double implication* I take the umbrella if and only if it rains.

♣

We use the contrapositive in order to express (3.7) (or (3.9)). Let us denote by A the proposition $x > 0$ and by B the proposition $y = 0$. Being $[0, +\infty[$ the domain of variable x, then $\bar{A}$ is $\{0\}$ or[1] $x = 0$. Similarly, being y a binary variable, then $\bar{B}$ corresponds to $y = 1$. Therefore, applying the contrapositive to (3.9) we get

$$y = 1 \longrightarrow x > 0, \tag{3.16}$$

which is the opposite implication of (3.9). Therefore, the relationship (3.5) is also equivalent to the following double implication:

$$x > 0 \longleftrightarrow y = 1. \tag{3.17}$$

Unfortunately, because of the strict inequality, it is not possible to completely express (3.16). Nevertheless, we can transform (3.16) into a non strict inequality. Consider a small positive number ϵ, then we can write (3.16) as follows.

$$y = 1 \longrightarrow x \geq \epsilon, \tag{3.18}$$

which is described by the following constraint:

$$x \geq \epsilon y. \tag{3.19}$$

We wish to point out that (3.19) does describe (3.7), which is one of our two initial implications. Recall that we want y to be 0 when x is equal to 0. And if we substitute 0 to x within (3.19) we get $0 \geq \epsilon y$. As $\epsilon > 0$ we can divide side by side by ϵ and without reversing the inequality and get $y \leq 0$. But y is a binary

[1]Recall that $A \cup \bar{A}$ is the so called *universal set*. Since set $[0, +\infty[$ –which corresponds to $x \geq 0$– can be split in $\{0\} \cup [0, +\infty[$ –which corresponds to the union of $x = 0$ with $x > 0$– and $x > 0$ corresponds to $x > 0$, then $\bar{A}$ is $x = 0$.

variable therefore $y \leq 0$ is equivalent to $y = 0$. Nevertheless (3.19) forces y to 0 even for values of x smaller than ϵ. Let θ be a positive number smaller than m. Then, plugging $x = \theta$ into (3.19) we get $\theta \geq \epsilon y$. Again, as $\epsilon > 0$ we can divide side by side by ϵ and without reversing the inequality and get

$$y \leq \frac{\theta}{\epsilon}. \tag{3.20}$$

But $\theta < \epsilon$ and using this information into (3.20) we have

$$y \leq \frac{\theta}{\epsilon} < \frac{\epsilon}{\epsilon} = 1. \tag{3.21}$$

But y is a binary variable therefore $y < 1$ is equivalent to $y = 0$. And this is in contrast with our initial goal, i.e., relationship (3.5), where we want y to be 1 for *any* positive value of x. Actually, even (3.10) does not completely describe (3.5) because reduces the initial domain of variable x from $[0, +\infty)$ to $[0, M]$. This means that, once an upper bound to x has been fixed, then values greater than M are not accepted, otherwise, as we have proved before, (3.10) is violated. Nevertheless, we can accept this negligible modeling flaw by choosing proper values of ϵ and M and finally affirm that relationship (3.5) is described by the following couple of constraints:

$$y = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0. \end{cases} \iff \begin{cases} x & \leq My \\ x & \geq \epsilon y. \end{cases} \tag{3.22}$$

Now we have all the tools in order to model our first problem with logical constraints. This is done in Example 8.

Example 8. ***A kebap shop - I***
The owner of a renowned Turkish kebap shop has to manage the preparation of his specialties. In particular, several foods are available (split in two categories, meats and vegetables) to prepare each kebap. The available meats are: lamb, beef and chicken. The available vegetables are: salad, tomatoes, onion and mint. The owner of the kebap shop needs to determine the best combination of foods for each kebap. Each food which will take part in the kebap will imply a cost (in €/kg) according to Table 3.3.

On the other hand, each kebap will be sold at a price of K €/kg. Formulate a linear programming model able to determine the best composition of each kebap in order to optimize the overall profit (given by the revenue of selling each kebap minus the cost for purchasing the foods) by satisfying the following constraints:

1. *each kebap cannot have more than C kg of meat and V kg of vegetables*
2. *the percentage of onion in each kebap cannot be greater than 10%*
3. *at most 5 foods can be used.*

Index	Food	Cost (€/kg)
1	Lamb	c_1
2	Beef	c_2
3	Chicken	c_3
4	Salad	c_4
5	Tomato	c_5
6	Onion	c_6
7	Mint	c_7

Table 3.3: Cost of each food

Note that it is not requested to determine the number *of kebaps to sell but the optimal* composition *of each kebap. Let* $I = \{1, 2, \ldots, 7\}$ *be the set of available foods, let* $x_i \geq 0$ *be the quantity (in kg) of the* i*-th food, then the overall purchaising cost is* $\sum_{i \in I} c_i x_i$*, while the overall revenue is equal to* $K \sum_{i \in I} x_i$*. Hence, the objective function is*

$$\max K \sum_{i \in I} x_i - \sum_{i \in I} c_i x_i = \max \sum_{i \in I} (K - c_i) x_i.$$

Constraints 1 on the maximum amounts of meat and vegetables can be expressed as follows: $x_1 + x_2 + x_3 \leq C$ *for the meats and* $x_4 + x_5 + x_6 + x_7 \leq V$ *for the vegetables.*

Constraint 2 on the percentage of onion is $x_6 \leq 0.1 \sum_{i \in I} x_i$*.*

Constraint 3 concerns food usage and is a logical constraints. We need to introduce proper indicator variables associated to each food stating whether that food is used in the kebap or not. Thus, we can write

$$\delta_i = \begin{cases} 1 & \text{if food } i \in I \text{ is used} \\ 0 & \text{otherwise.} \end{cases}$$

As for (3.22)*, this relationship is expressed through the following constraints.*

$$x_i \leq M \delta_i$$

$$x_i \geq \epsilon \delta_i,$$

Now, if food $i \in I$ *is used then* $\delta_i = 1$*. Therefore, the constraint on the maximum number of foods can be written as follows.*

$$\sum_{i \in I} \delta_i \leq 5.$$

To summarize, we have the following model:

$$\max \quad \sum_{i\in I}(K-c_i)x_i \tag{3.23}$$

$$\textit{s. t.} \quad x_1+x_2+x_3 \le C \tag{3.24}$$

$$x_4+x_5+x_6+x_7 \le V \tag{3.25}$$

$$x_6 \le 0.1\sum_{i\in I} x_i \tag{3.26}$$

$$x_i \le M\delta_i \qquad \forall\, i\in I \tag{3.27}$$

$$x_i \ge \epsilon\delta_i \qquad \forall\, i\in I \tag{3.28}$$

$$\sum_{i\in I}\delta_i \le 5 \tag{3.29}$$

$$x_i \ge 0 \qquad \forall\, i\in I \tag{3.30}$$

$$\delta_i \in \{0,\,1\} \qquad \forall\, i\in I. \tag{3.31}$$

♣

So far we have seen how the pair of constraints (3.22) is necessary to model the logical condition (3.5). In some problems we can take advantage of each or both of (3.22) if a particular constraint appears in the problem. Now we know that the implication *I1: If* $x=0$ *then* $y=0$ is modeled by constraint (3.19), which is $x \ge \epsilon y$, with $\epsilon > 0$ small enough. If we apply the contrapositive to $I1$, its equivalent proposition is *I2: If* $y=1$ *then* $x>0$ that we approximated to *I2: If* $y=1$ *then* $x \ge \epsilon$. Some problems present constraints on product of usage. Suppose, for instance, that you are at the supermarket and you are uncertain whether to buy your favorite dessert: muffins. You have to face with the following decision: either you don't buy muffins (perhaps because you have changed your mind), either you buy some. But if you decide to buy muffins then you would not be satisfied to eat just one but you would like to purchase at least five to share with your friends. If we name x the integer variable describing the number of muffins you will buy and y the indicator variable stating whether you buy or not the muffins, then these variables are linked as $x \le My$. Furthermore, implication (3.7) together with your "indecision" are modeled by the following constraint:

$$x \ge 5y, \tag{3.32}$$

which is (3.19) with $\epsilon = 5$. Note that we are using (3.32) for two purposes. The first one is to link x and y variables in the sense that if you don't buy muffins then the indicator variables must be zero. The second purpose is to model your second requirement: if you decide to buy muffins, you want to have at least five to share with your friends. In other words, we are expressing two requirements with just one constraint! Note also that it would be a mistake to write $x \ge 5$ instead of (3.32). This is a very small difference but it does change things completely. In fact, (3.32) ensures that you buy at least 5 muffins only if you decide to buy them.

But you might even not buy any. Vice versa, $x \geq 5$ ensures that you will always buy at least 5 muffins.

Example 9. ***A kebap shop - II***
Suppose to add to Example 8 the constraint that if salad is used, at least σ kg have to be present in each kebap. Then, we have to add to model (3.23) - (3.31) *the following constraint.*

$$x_4 \geq \sigma\delta_4.$$

♣

A similar consideration holds for constraint (3.10). Suppose that, because of your budget, you cannot afford to buy more than ten muffins. Then, you would add to your problem the following constraint: $x \leq 10$. Note that this time this constraint does not change the problem. Recall that if we had written $x \geq 5$ instead of $x \geq 5y$ we would have made a mistake because the former inequality implies that you *always* buy 5 muffins while you buy at least 5 muffins *only if* you decide to buy them. This time, writing $x \leq 10$ does not alter the problem because even if you don't buy any muffin ($x = 0$), we would have $0 \leq 10$ and the constraint is satisfied. But now we have two $\leq$ constraints. The first constraint, $x \leq My$ ensures that the indicator variable y is set to 1 when $x > 0$ (i. e., you buy). Then we have the constraint on your budget: $x \leq 10$. The presence of both constraints is correct but we can merge them into just one constraint as follows.

$$x \leq 10y. \tag{3.33}$$

In fact, since you cannot buy more than ten muffins, then 10 becomes the upper bound on x. Recall, indeed, that M is an upper bound to variable x and (3.33) corresponds to (3.10) with $M = 10$. Therefore, even with (3.33) we express two different things: 1) we link the decision variable x with the indicator (logical) variable y and 2) we express a maximum budget (or, depending on the problem, capacity[2]) constraint or any explicit upper bound or availability[3] on x.

Example 10. ***A kebap shop - III***
Suppose that the owner of the kebap shop in Example 8 has a limited amount of salad, say S. Then, we have to add to model (3.23) - (3.31) *the following constraint.*

$$x_4 \leq S\delta_4.$$

♣

The introduction of binary variables allows us to describe by means of linear constraints many connectives. Table 3.4 illustrates how logical connectives can be expressed in terms of linear constraints.

[2]Like in the Facility Location Problem presented in Section 3.2.1.
[3]This approach has also been used in problem 2.4.1, pp. 19-21 in the textbook.

LOGICAL CONNECTIVE	CORRESPONDING LINEAR CONSTRAINT
$A \vee B$	$y_A + y_B \geq 1$
$A \wedge B$	$y_A + y_B = 2$
$A \uparrow B$	$y_A + y_B \leq 1$
$A \oplus B$	$y_A + y_B = 1$
$\bar{A}$	$1 - y_A$
$A \longrightarrow B$	$y_A \leq y_B$
$A \longleftrightarrow B$	$y_A = y_B$

Table 3.4: Equivalence between connectives and logical constraints

Example 11 shows some example of constraints that could be added to the kebap problem in Example 8.

Example 11. ***A kebap shop - IV***
If the owner of the kebap shop in Example 8 wants both the lamb and *the salad to be present in each kebap, then the AND connective must be used. We see from Table 3.4 row 3, that this constraint is expressed as* $y_1 + y_4 = 2$*. In fact, being* y_1 *and* y_4 *binary variables, this expression forces them to be both at one.*

If we want to model the constraint that if beef is used then tomatoes must also be used, we use the if-then (or implies) connective (row 7). This constraint is modeled as $y_2 \leq y_5$*. In fact, when* $y_2 = 1$ *(i.e., beef is used), then the constraint becomes* $y_5 \geq 1$ *which forces the tomatoes to be used.*

We can also combine the implies connective with the not connective in order to represent a similar constraint. Suppose that we want to model the requirement that if beef is used than chicken cannot be present in the kebap. Denote by $P2$ *and* $P3$ *the following propositions:*

$P2$*: beef is used*

$P3$*: chicken is used.*

We want to express the following implication: $P2 \longrightarrow \overline{P3}$*. From row 6 in Table 3.4 we see that* $y_2 \leq y_3$ *represents the implication* $P2 \longrightarrow P3$*, while we want to model* $P2 \longrightarrow \overline{P3}$*. Nevertheless, from row 5 in Table 3.4 we see that the NOT connective applied to* $P3$ *is expressed as* $1 - y_3$*. Therefore, the desired constraint is* $y_2 \leq 1 - y_3$*. In fact, when* $y_2 = 1$ *(i.e., beef is used), then the constraint becomes* $1 \leq 1 - y_3$ *or* $y_3 \leq 0$*, which means that chicken cannot be present in the kebap.*

The two last constraints are two examples of the so called if-then constraints. We will thoroughly address these constraints in Section 3.3.

♣

There are some circumstances where we do not need to explicitly write the whole relationship (3.22) but where only constraint (3.10) is enough to link the indicator variable y with the decision variable x. One example is a class of problems, named fixed charge problems, which are discussed next.

3.2.1 Fixed charge problems

Fixed charge problems represent those situations where, in order to start an activity, a setup (or fixed charge) cost must be payed. Suppose that you have to decide whether to open or not an activity, say a company that produces cars. There is a cost c for producing each car. If we denote by x the decision variable representing the number of manufactured cars, the overall production cost is given by cx. But, usually, a setup cost must be spent in order to start the activity. For instance, you might pay a setup cost f in order to open your car factory. In this case, you pay an overall cost of $f + cx$ i. e., the cost for opening your car factory plus the cost for each manufactured car. Nevertheless, you might not want to open your firm if this activity is not convenient. For instance, the profit yielded by your sales might not be higher than the costs you pay. Clearly, in this case you would not do anything and you pay 0. Therefore, we can introduce an indicator variable y which is equal to 1 if you decide to start the activity and sell your cars (i.e., $x > 0$), 0 otherwise. We can also write a general cost as $fy + cx$ instead of $f + cx$. The latter is payed when $x > 0$ while the former also includes the case where you don't do anything. In this case, in fact, both x and y are equal to 0. As always, x and y are linked like in (3.22) and, since you want to minimize your costs, we can write:

$$\begin{aligned}
\min \quad & fy + cx && (3.34)\\
\text{s. t.} \quad & x \leq My && (3.35)\\
& x \geq \epsilon y && (3.36)\\
& \text{...other constraints} \\
& x \in \mathbb{N} && (3.37)\\
& y \in \{0,\, 1\}. && (3.38)
\end{aligned}$$

Recall that (3.36) is used because we want y to be 0 when x is 0. This behavior is not guaranteed by just writing (3.35) because in this constraint y is free to be either 0 or 1 when x is equal to 0. Nevertheless, in this problem there will never be an optimal solution with $x = 0$ and $y = 1$ because, as we are minimizing the term $fy + cx$, if the optimal solution has $x = 0$ then even y will be set to this value. Therefore, in this case, even if its presence is formal and correct, we do not need to explicitly write constraint (3.36). Actually, we could always avoid to skip constraint $x \geq \epsilon y$ even for problems which are not fixed charge ones. The trick consists of artificially adding the logical variables in the objective function in order to force them to be 0 when their corresponding indicator variables are also at zero. Nevertheless, this strategy must be taken with a lot of care because proper coefficients should be chosen for the indicator variables not to alter the problem. In fact, these coefficient strongly depend on the objective function. Vice versa, this choice is natural for fixed charge problems because the indicator variables explicitly appear within the objective function.

Example 12. *Suppose that you need to buy a number of units of product and you can choose between two suppliers. You have to buy your units of product from at*

least one of the two suppliers. Let $x_i \geq 0$ *denote the units of product bought from supplier* $i \in \{1, 2\}$*. Let also* c_i *denote the cost for buying a unit of product from supplier* $i \in \{1, 2\}$*. We can introduce the indicator variables* y_i *equal to 1 if we buy from supplier* $i \in \{1, 2\}$*, 0 otherwise. In order to minimize the purchasing costs, we can write the following model:*

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 && (3.39)\\ \textit{s. t.} \quad & x_i \leq My_i \quad i \in \{1, 2\} && (3.40)\\ & x_i \geq \epsilon y_i \quad i \in \{1, 2\} && (3.41)\\ & y_1 + y_2 \geq 1. && (3.42) \end{aligned}$$

In order to drop constraints (3.41) *we can modify the objective function* (3.39) *as follows.*

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 + \epsilon y_1 + \epsilon y_2 && (3.43)\\ \textit{s. t.} \quad & x_i \leq My_i \qquad i \in \{1, 2\} && (3.44)\\ & y_1 + y_2 \geq 1. && (3.45) \end{aligned}$$

Since this is a minimization problem, when x_i *is 0 then* y_i *is automatically set to 0. Consider now the case where you get a revenue* r_i *if you go to supplier* $i \in \{1, 2\}$*. In this case model* (3.39)-(3.42) *becomes:*

$$\begin{aligned} \max \quad & r_1x_1 + r_2x_2 && (3.46)\\ \textit{s. t.} \quad & x_i \leq My_i \quad i \in \{1, 2\} && (3.47)\\ & x_i \geq \epsilon y_i \quad i \in \{1, 2\} && (3.48)\\ & y_1 + y_2 \geq 1 && (3.49) \end{aligned}$$

Now, if we want to drop constraints (3.48) *we cannot add* $+\epsilon y_i$ *to the objective function, otherwise (since now we are maximizing) we would exactly get the undesired drawback: that* y_i *is 1 when* x_i *is 0. Therefore, we have to add* $-my_i$ *to the objective function and get the following model.*

$$\begin{aligned} \max \quad & r_1x_1 + r_2x_2 - \epsilon y_1 - \epsilon y_2 && (3.50)\\ \textit{s. t.} \quad & x_i \leq My_i \qquad i \in \{1, 2\} && (3.51)\\ & y_1 + y_2 \geq 1. && (3.52) \end{aligned}$$

♣

From Example 12 we can affirm that it is possible to skip constraints $x \geq \epsilon y$ provided the indicator variables are placed in the objective function with proper coefficients. Moreover, some authors (see, for instance, [4]) present the fixed charge problem in the maximization form as appears in model (3.50)-(3.52).

3.3 If-then constraints

In this section, we present a powerful methodology which allows us to express advanced logical constraints. If-then constraints are used to model those settings in which if a proposition is true, then another proposition must be guaranteed.

Table 3.5 shows several kinds of implications we can meet when dealing with if-then constraints.

#	IMPLICATION
(I1)	$y = 1 \longrightarrow \sum_j a_j x_j \leq b$
(I2)	$\sum_j a_j x_j \leq b \longrightarrow y = 1$
(I3)	$y = 1 \longrightarrow \sum_j a_j x_j \geq b$
(I4)	$\sum_j a_j x_j \geq b \longrightarrow y = 1$
(I5)	$y = 1 \longrightarrow \sum_j a_j x_j = b$
(I6)	$\sum_j a_j x_j = b \longrightarrow y = 1$

Table 3.5: Canonical implications in if-then constraints

These implications are generic because coefficients do not take particular values. Therefore, the method we are going to present is a general one. We suppose that the expression $\sum_j a_j x_j - b$ is bounded from below and from above, i. e., there exist finite values m and M such that

$$m \leq \sum_j a_j x_j - b \leq M. \tag{3.53}$$

The trick consists of always to reduce to the case where the indicator variable y or δ is to the left of the implication symbol "$\longrightarrow$".

Implication (I1) is already in this form:

$$y = 1 \longrightarrow \sum_j a_j x_j \leq b \tag{3.54}$$

We manipulate (3.54) as follows.

$$1 - y = 0 \longrightarrow \sum_j a_j x_j - b \leq 0 \tag{3.55}$$

At this point, we can translate the logical implication in the following linear constraint:

$$\sum_j a_j x_j - b \leq \alpha(1-y), \tag{3.56}$$

with α proper constant to determine. When $y = 1$ we have that $\sum_j a_j x_j - b \leq 0$ or $\sum_j a_j x_j \leq b$. Since $y \in \{0, 1\}$, the only value variable y can take but 1 is 0. And when $y = 0$ we have that

$$\alpha \geq \sum_j a_j x_j - b. \tag{3.57}$$

This means that α is an upper bound to the expression $\sum_j a_j x_j - b$. But from (3.53) we know that this upper bound is equal to M. Therefore, $\alpha = M$ and (**??**) becomes

$$\sum_j a_j x_j - b \leq M(1-y). \tag{3.58}$$

Implication (I2) is not so straightforward to represent because the indicator variable is not to the left of the implication sign. The trick consist of applying the contrapositive to implication (I2) and get the following equivalent implication:

$$y = 0 \longrightarrow \sum_j a_j x_j > b. \tag{3.59}$$

In this implication the indicator variable is to the left of the implication sign but we have a strict inequality which is not accepted in Linear Programming. As done in Section 3.2, we can transform a strict inequality into a non-strict one by accepting a small approximation error. In particular, we can introduce an $\epsilon > 0$ small enough and rewrite (3.59) as

$$y = 0 \longrightarrow \sum_j a_j x_j \geq b + \epsilon. \tag{3.60}$$

Note that, if variables x_j and coefficients a_j are integer, then ϵ can be set to 1. Taking everything to the left of the $\geq$ side in (3.60), we get

$$y = 0 \longrightarrow \sum_j a_j x_j - b - \epsilon \geq 0. \tag{3.61}$$

This implication can be represented by the following linear constraint:

$$\sum_j a_j x_j - b - \epsilon \geq \alpha y, \tag{3.62}$$

with α proper constant to determine. When $y = 0$ the implication is satisfied. But when $y = 1$ we have that

$$\alpha \leq \sum_j a_j x_j - b - \epsilon, \tag{3.63}$$

or

$$\alpha + \epsilon \leq \sum_j a_j x_j - b. \tag{3.64}$$

This means that $\alpha + \epsilon$ is a lower bound to the expression $\sum_j a_j x_j - b$. But from (3.53) we know that this lower bound is equal to m. Therefore, $m = \alpha + \epsilon$ which yields $\alpha = m - \epsilon$ and (3.62) becomes

$$\sum_j a_j x_j - b - \epsilon \geq (m - \epsilon)y. \tag{3.65}$$

Implication (I3) can be represented as follows.

$$1 - y = 0 \longrightarrow \sum_j a_j x_j - b \geq 0 \tag{3.66}$$

This implication can be represented by the following linear constraint:

$$\sum_j a_j x_j - b \geq \alpha(1 - y), \tag{3.67}$$

with α proper constant to determine. When $y = 1$ the implication is satisfied. But when $y = 0$ we have that

$$\alpha \leq \sum_j a_j x_j - b, \tag{3.68}$$

This means that α is a lower bound to the expression $\sum_j a_j x_j - b$. But from (3.53) we know that this lower bound is equal to m. Therefore, $\alpha = m$ and (3.67) becomes

$$\sum_j a_j x_j - b \geq m(1 - y). \tag{3.69}$$

In order to manage (I4) we have to apply the contrapositive because the indicator variable is to the right of the implication simble. We get

$$y = 0 \longrightarrow \sum_j a_j x_j < b, \tag{3.70}$$

which, introducing a small approximation error of at most $\epsilon > 0$, can be transformed as

$$y = 0 \longrightarrow \sum_j a_j x_j \leq b - \epsilon. \tag{3.71}$$

Taking everything to the left of the $\leq$ sign we get

$$y = 0 \longrightarrow \sum_j a_j x_j - b + \epsilon \leq 0. \tag{3.72}$$

This implication can be represented by the following linear constraint:

$$\sum_j a_j x_j - b + \epsilon \leq \alpha y, \tag{3.73}$$

with α proper constant to determine. When $y = 0$ the implication is satisfied. But when $y = 1$ we have that

$$\alpha \geq \sum_j a_j x_j - b + \epsilon, \tag{3.74}$$

or

$$\alpha - \epsilon \geq \sum_j a_j x_j - b. \tag{3.75}$$

This means that $\alpha - \epsilon$ is an upper bound to the expression $\sum_j a_j x_j - b$. But from (3.53) we know that this upper bound is equal to M. Therefore, $M = \alpha - \epsilon$ which yields $\alpha = M + \epsilon$ and (3.74) becomes

$$\sum_j a_j x_j - b + \epsilon \leq (M + \epsilon) y. \tag{3.76}$$

In order to express (I5), we transform the equality $\sum_j a_j x_j = b$ into the following system of inequalities[4]:

$$\begin{cases} \sum_j a_j x_j & \geq b \\ \\ \sum_j a_j x_j & \leq b. \end{cases} \tag{3.77}$$

Therefore, (I5) can be stated as follows

$$\begin{cases} y = 1 & \longrightarrow \sum_j a_j x_j \geq b \\ \\ y = 1 & \longrightarrow \sum_j a_j x_j \leq b, \end{cases} \tag{3.78}$$

which means writing together (3.58) and (3.69):

[4] A similar procedure has been employed in the textbook at page 62 when dealing with the asymmetric form of the dual. In that circumstance, constraints $A\bar{x} = \bar{b}$ where written as $A\bar{x} \geq \bar{b}$ and $-A\bar{x} \leq -\bar{b}$.

$$\begin{cases} \sum_j a_j x_j - b & \leq M(1-y) \\ \\ \sum_j a_j x_j - b & \geq m(1-y). \end{cases} \tag{3.79}$$

Implication (I6) is the most difficult case to treat. A first attempt might be to transform, as done in case (I5), the equality $\sum_j a_j x_j = b$ into a set of inequalities and get

$$\begin{cases} \sum_j a_j x_j \geq b & \longrightarrow y = 1 \\ \\ \sum_j a_j x_j \leq b & \longrightarrow y = 1. \end{cases} \tag{3.80}$$

However, this way of thinking, although valuable in principle, is wrong in this case. In fact, if the set of implications (3.80) holds, then we have that the indicator variable y is always at 1, either when $\sum_j a_j x_j \geq b$ or when $\sum_j a_j x_j \leq b$. But we want y to be 1 *only when* $\sum_j a_j x_j = b$. Therefore, we have to handle implication (I6) in a more complex way. We do split the equality $\sum_j a_j x_j = b$ in two inequalities but with two different indicator variables, y_I and y_{II}, one for each inequality:

$$\begin{cases} \sum_j a_j x_j \geq b & \longrightarrow y_I = 1 \\ \\ \sum_j a_j x_j \leq b & \longrightarrow y_{II} = 1. \end{cases} \tag{3.81}$$

Now, only when y_I *and* y_{II} are equal to 1 the equality holds. And only in this case we want to force y to be 1. Therefore, in order to completely express implication (I6), we have to add to (3.81) the following implication:

$$y_I = 1 \wedge y_{II} = 1 \longrightarrow y = 1. \tag{3.82}$$

By Table 3.4, third row, we see that the compound proposition (with the *and* connective) $y_I = 1 \wedge y_{II} = 1$ can be expressed as $y_I + y_{II} = 2$, and (3.82) becomes

$$y_I + y_{II} = 2 \longrightarrow y = 1. \tag{3.83}$$

Moreover, the two implications in (3.81) can be expressed through (3.76) and (3.65) respectively. The last endeavor consists of expressing implication (3.83)

by means of a linear constraint. Again, we have an equality which implies an indicator variable to be 1, as in our current case, i.e., (I6). Nevertheless, while in (I6) we have a general expression in the form $\sum_j a_j x_j = b$ where x_j is in principle continuous (i.e., $x_j > 0$), now we have to deal with a simpler expression involving binary variables only. In fact, as both y_I and y_{II} are binary variables, it is easy to see that writing $y_I + y_{II} = 2$ is equivalent to write $y_I + y_{II} \geq 2$ and (3.83) becomes

$$y_I + y_{II} \geq 2 \longrightarrow y = 1, \tag{3.84}$$

which can be managed like in (I2). Applying the contrapositive to (3.84) we have:

$$y = 0 \longrightarrow y_I + y_{II} < 2. \tag{3.85}$$

We transform the strict inequality into a non-strict one introducing $\epsilon > 0$ as follows.

$$y = 0 \longrightarrow y_I + y_{II} \leq 2 - \epsilon. \tag{3.86}$$

Since y_I and y_{II} are binary variables and their coefficients are integer, we can set ϵ to 1 and get

$$y = 0 \longrightarrow y_I + y_{II} \leq 1, \tag{3.87}$$

or

$$y = 0 \longrightarrow y_I + y_{II} - 1 \leq 0. \tag{3.88}$$

This implication can be represented by the following linear constraint:

$$y_I + y_{II} - 1 \leq \alpha y, \tag{3.89}$$

with α proper constant to determine. When $y = 0$ the implication is satisfied. But when $y = 1$ we have that

$$\alpha \geq y_I + y_{II} - 1. \tag{3.90}$$

This means that α is an upper bound to the expression $y_I + y_{II} - 1$. A general upper bound could be set equal to M. But in this specific case, as y_I and y_{II} are binary variables, a specific upper bound is found when y_I and y_{II} assume their maximum value which is 1. Thus, the upper bound is 1 + 1 - 1 = 1 and (3.90) becomes

$$y_I + y_{II} - 1 \leq y. \tag{3.91}$$

Finally, we are able to express (I6) as follows.

$$\begin{cases} \sum_j a_j x_j - b - \epsilon \geq (m-\epsilon) y_I \\ \sum_j a_j x_j - b + \epsilon \leq (M+\epsilon) y_{II} \\ y_I + y_{II} - 1 \leq y. \end{cases} \tag{3.92}$$

We can summarize the work done in this section by adding to Table 3.4 the linear constraints corresponding to the treated implications. This is done in Table 3.6.

#	IMPLICATION	CORRESPONDING LINEAR CONSTRAINT(S)
(I1)	$y = 1 \longrightarrow \sum_j a_j x_j \leq b$	$\sum_j a_j x_j - b \leq M(1-y)$
(I2)	$\sum_j a_j x_j \leq b \longrightarrow y = 1$	$\sum_j a_j x_j - b - \epsilon \geq (m-\epsilon) y$
(I3)	$y = 1 \longrightarrow \sum_j a_j x_j \geq b$	$\sum_j a_j x_j - b \geq m(1-y)$
(I4)	$\sum_j a_j x_j \geq b \longrightarrow y = 1$	$\sum_j a_j x_j - b + \epsilon \leq (M+\epsilon) y$
(I5)	$y = 1 \longrightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j - b \leq M(1-y)$ $\sum_j a_j x_j - b \geq m(1-y)$
(I6)	$\sum_j a_j x_j = b \longrightarrow y = 1$	$\sum_j a_j x_j - b - \epsilon \geq (m-\epsilon) y_I$ $\sum_j a_j x_j - b + \epsilon \leq (M+\epsilon) y_{II}$ $y_I + y_{II} - 1 \leq y$

Table 3.6: Summary of general implications met in if-then constraints

We conclude this section by proposing in the next examples several problems containing if-then constraints.

Example 13. *Implications* (**??**) *and* (**??**) *from Example* **??** *are expressed as follows. Applying the contrapositive to* (**??**) *we get*

$$y = 0 \longrightarrow x_1 + x_2 < 10, \tag{3.93}$$

which we transform as

$$y = 0 \longrightarrow x_1 + x_2 \leq 10 - \epsilon. \tag{3.94}$$

But variables and their coefficients are integer and we can set $\epsilon = 1$ *and get*

$$y = 0 \longrightarrow x_1 + x_2 \leq 9, \tag{3.95}$$

which can be expressed as

$$x_1 + x_2 - 9 \leq \alpha y. \tag{3.96}$$

When $y = 0$ *the implication is satisfied, but when* $y = 1$ *we have*

$$\alpha \geq x_1 + x_2 - 9. \tag{3.97}$$

This means that α *is an upper bound for this expression. Since we do not have from Example* **??** *an explicit upper bound of variables* x_1 *and* x_2*, we set* $\alpha = M$ *and get*

$$x_1 + x_2 - 9 \leq My. \tag{3.98}$$

Implication (**??**) *becomes*

$$1 - y = 0 \longrightarrow x_3 + x_4 - 2 \leq 0, \tag{3.99}$$

which can be written as

$$x_3 + x_4 - 2 \leq \alpha(1 - y). \tag{3.100}$$

In order to find the right value for α *we set* y *to 0 and see* α *is an upper bound of* $x_3 + x_4 - 2$*. Even in this case we do not have a specific upper bound for this expression. Therefore, we set* $\alpha = M$ *and get*

$$x_3 + x_4 - 2 \leq M(1 - y). \tag{3.101}$$

with M *big enough to satisfy both* (3.98) *and* (3.101)*.*

♣

Example 14. ***A kebap shop - V***
Suppose that the owner of the kebap shop in Example 8 has also to model the following constraint. If mint is used then at most one vegetable between tomato and onion can be present in each kebap. This if-then constraint can be expressed as follows.

$$\delta_7 = 1 \longrightarrow \delta_5 + \delta_6 \leq 1, \tag{3.102}$$

or

$$1 - \delta_7 = 0 \longrightarrow \delta_5 + \delta_6 - 1 \leq 0. \tag{3.103}$$

This implication can be modeled with the following linear constraint.

$$\delta_5 + \delta_6 - 1 \leq \alpha(1 - \delta_7), \tag{3.104}$$

with α proper constraint to determine. When $\delta_7 = 0$, we have $\alpha \geq \delta_5 + \delta_6 - 1$. Hence, α is an upper bound to the expression $\delta_5 + \delta_6 - 1$. This upper bound is clearly $1 + 1 - 1 = 1$ and the mint constraint becomes

$$\delta_5 + \delta_6 - 1 \leq 1 - \delta_7, \tag{3.105}$$

i.e.,

$$\delta_5 + \delta_6 + \delta_7 \leq 2. \tag{3.106}$$

♣

3.4 A general method

The aim of this section is to show how the method illustrated in Section 3.3 is not only useful to express if-then constraints by means of linear constraints (although this is already a powerful tool), but can also address the theory presented in Section 3.2 of indicator variables. This is proved in Section **??**. Finally, the method introduced in Section 3.3 can also be employed in order to express advanced logical relationships. This yields the great advantage to avoid to memorize them. Thus, advanced formulas can be easily reproduced by applying the simple concepts at the base of if-then constraints of Section 3.3. We show how to reproduce these formulas in Section 3.4.1. Finally, in Section 3.4.2, we show how the methodology proposed in Section 3.3 can also include other alternative if-then constraints formulations that can be found in the literature.

3.4.1 Advanced logical relationships

The if-then constraints method presented in Section 3.2 can also be employed to express advanced logical relationships involving several indicator variables. Two of these are reported in Table 3.7

#	LOGICAL PROPOSITION	CORRESPONDING LINEAR CONSTRAINT
(I7)	$A_1 \vee A_2 \vee \ldots \vee A_n \longrightarrow B$	$\sum_{i=1}^{n} y_i \leq n y_B$
(I8)	$A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow B$	$\sum_{i=1}^{n} y_i \leq (n-1) + y_B$

Table 3.7: Equivalence between some propositions and their logical constraints

In the following, we show that there is no need to memorize (I7) and (I8) but that these formulas can be easily obtained by applying the concepts presented in Section 3.2.

3.4.2 Other notations

Some authors present if-then constraints with a different formulation. In particular, if constraint $f(x_1, \ldots, x_n) > 0$ is satisfied, then even constraint $g(x_1, \ldots, x_n) \geq 0$ must be satisfied as well. Winston [4] formulates this implication with the following couple of constraints.

$$-g(x_1, \ldots, x_n) \leq My \tag{3.107}$$
$$f(x_1, \ldots, x_n) \leq M(1-y). \tag{3.108}$$

In this section, we want to show that the method presented in Section 3.3 is also able to include formulation (3.107) -(3.108). In particular, we express the desired implication as follows.

$$f(x_1, \ldots, x_n) > 0 \longrightarrow g(x_1, \ldots, x_n) \geq 0. \tag{3.109}$$

This implication can be split as

$$f(x_1, \ldots, x_n) > 0 \longrightarrow y = 1 \tag{3.110}$$
$$y = 1 \longrightarrow g(x_1, \ldots, x_n) \geq 0. \tag{3.111}$$

Nevertheless, in order to get constraints (3.107) -(3.108), we slightly modify implications (3.110)-(3.111). It is obvious that we can also split implication (3.109) as follows.

$$f(x_1, \ldots, x_n) > 0 \longrightarrow y = 0 \tag{3.112}$$
$$y = 0 \longrightarrow g(x_1, \ldots, x_n) \geq 0. \tag{3.113}$$

Naturally, M must be chosen big enough such that it is an upper bound both for $f(x_1, \ldots, x_n)$ and $-g(x_1, \ldots, x_n)$.

3.5 Tower Blocks

Each square of a chessboard must be colored in one among the following colors: yellow, green, red or blue.

- yellow squares have higher priorities than green ones
- green squares have higher priorities than red ones
- red squares have higher priorities than blue ones.
- a yellow square must be surrounded by at least one green, one red and one blue square
- a green square must be surrounded by at least one red and one blue square
- a red square must be surrounded by at least one blue square
- no adjacent squares with the same color are allowed in the chessboard.

3.5.1 Terminology

We name K the set of colors defined as follows.

$$K = \{B,\, R,\, G,\, Y\},$$

where B stands for blue, etc.

The only parameters involved in the model are the priorities associated to each color. We denote by p_k the priority of color $k \in K$. Clearly, we must set priority values such that $p_Y > p_G > p_R > p_B$.

We can next introduce a boolean variable for each square in the chessboard and for each color. Therefore, if we have a $m \times n$ chessboard and $k = |K|$ colors, we then have $m \cdot n \cdot k$ boolean variables. We can define each of these variables as follows.

$$x_{ij}^k = \begin{cases} 1 & \text{if square } (i,\, j) \text{ is colored in } k \in K \\ 0 & \text{otherwise.} \end{cases} \tag{3.114}$$

Relationship (3.114) is correct but how do we relate indexes i and j to the position of the associated square in the chessboard? In order to clarify this point, we restrict ourselves to a 5×5 chessboard, like in the real game.

(5, 5)

(0.5, 4.5)	(1.5, 4.5)	(2.5, 4.5)	(3.5, 4.5)	(4.5, 4.5)
(0.5, 3.5)	(1.5, 3.5)	(2.5, 3.5)	(3.5, 3.5)	(4.5, 3.5)
(0.5, 2.5)	(1.5, 2.5)	(2.5, 2.5)	(3.5, 2.5)	(4.5, 2.5)
(0.5, 1.5)	(1.5, 1.5)	(2.5, 1.5)	(3.5, 1.5)	(4.5, 1.5)
(0.5, 0.5)	(1.5, 0.5)	(2.5, 0.5)	(3.5, 0.5)	(4.5, 0.5)

(0, 0)

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]
[3][0]	[3][1]	[3][2]	[3][3]	[3][4]
[4][0]	[4][1]	[4][2]	[4][3]	[4][4]

	x^k_{i-1j}	
x^k_{ij-1}	x^k_{ij}	x^k_{ij+1}
	x^k_{i+1j}	

3.5.2 The objective function

$$\max \sum_{k \in K} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} p_k x^k_{ij}. \tag{3.115}$$

3.5.3 The constraints

Only one color per square

$$\sum_{k \in K} x^k_{ij} = 1 \qquad i \in \{0, \ldots, m-1\}, j \in \{0, \ldots, n-1\}. \tag{3.116}$$

Red constraints

If square $M[i][j]$ is colored red, then at least one among the sorrounding squares must be colored blue.

$$x_{ij}^R = 1 \longrightarrow x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \geq 1. \tag{3.117}$$

$$x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \geq x_{ij}^R. \tag{3.118}$$

Green constraints

If square $M[i][j]$ is colored green, then at least one among the sorrounding squares must be colored red and at least one blue.

$$\begin{aligned} x_{ij}^G = 1 \quad &\longrightarrow \quad x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \geq 1 && (3.119)\\ x_{ij}^G = 1 \quad &\longrightarrow \quad x_{ij-1}^R + x_{i-1j}^R + x_{i+1j}^R + x_{ij+1}^R \geq 1. && (3.120) \end{aligned}$$

$$\begin{aligned} x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \quad &\geq \quad x_{ij}^G && (3.121)\\ x_{ij-1}^R + x_{i-1j}^R + x_{i+1j}^R + x_{ij+1}^R \quad &\geq \quad x_{ij}^G. && (3.122) \end{aligned}$$

Yellow constraints

If square $M[i][j]$ is colored yellow, then at least one among the sorrounding squares must be colored green, at least one red and at least one blue.

$$\begin{aligned} x_{ij}^3 = 1 \quad &\longrightarrow \quad x_{ij-1}^0 + x_{i-1j}^0 + x_{i+1j}^0 + x_{ij+1}^0 \geq 1 && (3.123)\\ x_{ij}^3 = 1 \quad &\longrightarrow \quad x_{ij-1}^1 + x_{i-1j}^1 + x_{i+1j}^1 + x_{ij+1}^1 \geq 1 && (3.124)\\ x_{ij}^3 = 1 \quad &\longrightarrow \quad x_{ij-1}^2 + x_{i-1j}^2 + x_{i+1j}^2 + x_{ij+1}^2 \geq 1. && (3.125) \end{aligned}$$

$$\begin{aligned} x_{ij-1}^0 + x_{i-1j}^0 + x_{i+1j}^0 + x_{ij+1}^0 \quad &\geq \quad x_{ij}^3 && (3.126)\\ x_{ij-1}^1 + x_{i-1j}^1 + x_{i+1j}^1 + x_{ij+1}^1 \quad &\geq \quad x_{ij}^3 && (3.127)\\ x_{ij-1}^2 + x_{i-1j}^2 + x_{i+1j}^2 + x_{ij+1}^2 \quad &\geq \quad x_{ij}^3. && (3.128) \end{aligned}$$

Color constraints

If square $M[i][j]$ is colored with color $k \in K$, then the surrounding squares cannot be colored with the same color.

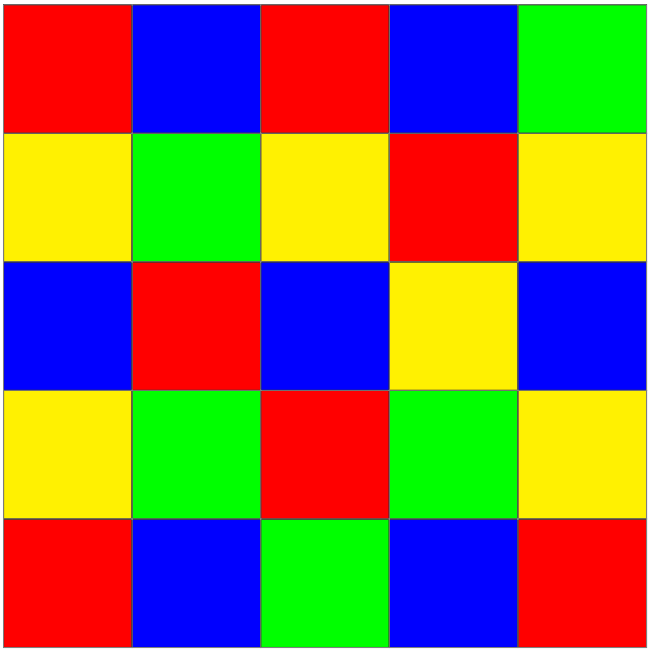
$$x_{ij}^k = 1 \longrightarrow x_{ij-1}^k + x_{i-1j}^k + x_{i+1j}^k + x_{ij+1}^k \leq 0. \tag{3.129}$$

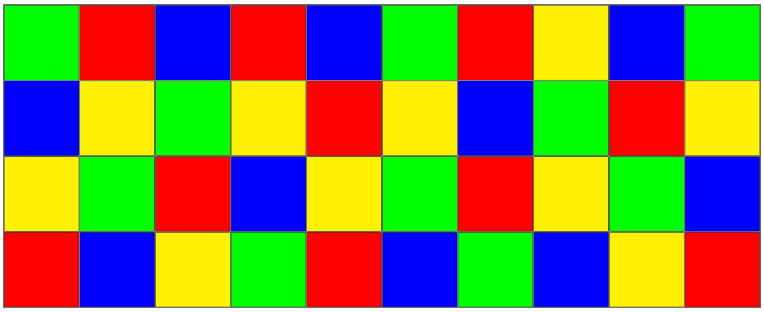
$$x_{ij-1}^k + x_{i-1j}^k + x_{i+1j}^k + x_{ij+1}^k + 4x_{ij}^k \leq 4. \tag{3.130}$$

3.5.4 The final model

$$
\begin{aligned}
\max \quad & \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} p_k x_{ij}^k \\
\text{s. t.} \quad & \sum_{k \in K} x_{ij}^k = 1 && i \in \{0, \ldots, m-1\},\ j \in \{0, \ldots, n-1\} \\
& x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \geq x_{ij}^R && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^0 \geq x_{ij}^G && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^R + x_{i-1j}^R + x_{i+1j}^R + x_{ij+1}^R \geq x_{ij}^G && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^B + x_{i-1j}^B + x_{i+1j}^B + x_{ij+1}^B \geq x_{ij}^Y && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^R + x_{i-1j}^R + x_{i+1j}^R + x_{ij+1}^R \geq x_{ij}^Y && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^G + x_{i-1j}^G + x_{i+1j}^G + x_{ij+1}^G \geq x_{ij}^Y && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij-1}^k + x_{i-1j}^k + x_{i+1j}^k + x_{ij+1}^k + 4x_{ij}^k \leq 4 && i \in \{1, \ldots, m-2\},\ j \in \{1, \ldots, n-2\},\ k \in K \\
& x_{ij}^k \in \{0, 1\} && i \in \{0, \ldots, m-1\},\ j \in \{0, \ldots, n-1\},\ k \in K.
\end{aligned}
$$

3.5.5 The program and some examples





4

The simplex algorithm

As seen in the previous chapters, linear programming problems are ubiquitous in many fields. Sometimes, we also deal with linear programming problems in our lives, without being aware of them. Consider, for instance, the following fitness problem.

Example 15. *You are at the gym and two training exercises are available: the functional training and the cardio workout. The fees for your training are: 2€ per hour of functional training and 4€ per hour of cardio workout. About 400 and 200 calories are burnt respectively for each hour of functional training and cardio workout. Since other people attend the gym, you can train yourself at each session for at most 2 hours. You have a budget of 9€ and you want to burn at most 900 calories.*

♣

You clearly want to stay as much as possible at the gym, but you have to satisfy budget and calories constraints. Therefore, if we let x_1 be the time devoted to the functional training session and x_2 the time spent for the cardio workout, we get the following linear programming model:

$$\begin{aligned}
\max z = \ & x_1 + x_2 && (4.1)\\
\text{s. t.} \quad & 2x_1 + 4x_2 \leq 9 && (4.2)\\
& 4x_1 + 2x_2 \leq 9 && (4.3)\\
& x_1 \leq 2 && (4.4)\\
& x_2 \leq 2 && (4.5)\\
& x_1, \quad x_2 \geq 0. && (4.6)
\end{aligned}$$

In the previous chapters we learned the art of modeling; namely the translation of a problem from a concept expressed in words to a rigorous mathematical formulation. In this chapter, we want to proceed one step farther: given a linear programming model like (4.1)–(4.6), is there a way to solve it? Solving a linear programming model requires to determine 1) whether there exists a solution and 2) if so, find an optimal solution. Coming to our example, this means answering

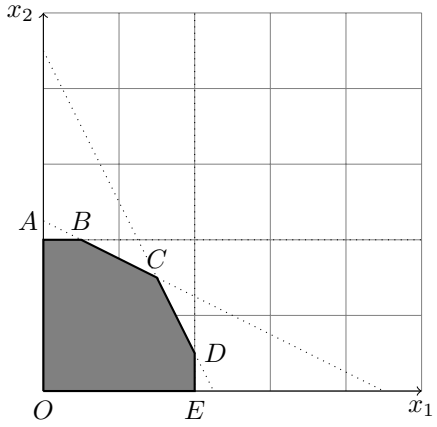


Figure 4.1: The domain of model (4.1)–(4.6).

to the following question: what is the right combination of the time dedicated to the functional training and the cardio workout, such that we can maximize the leisure, while satisfying budget and calories constraints?

In this chapter, we study the simplex algorithm, which is a popular method able to solve linear programming models, which variables are required to be non-negative. We wish to point out that the simplex algorithm solves linear programming problems which variables are *continuous* and non-negative. The special case where the variable are *integer* and non-negative are the so-called integer linear programming problems, which are ways more difficult to solve. Their resolution will be treated in Chapter 4.121. For simple problems, involving two or three variables, the resolution can also be graphical because we are able to draw the domain (given by the constraints of the problem) respectively in a plane or in a three-dimensional space.

Example 16. *The domain of model* (4.1)–(4.6) *is the shaded area depicted in Figure 4.1.*

♣

For the graphical resolution we can consider the objective function in the following form:

$$x_2 = -x_1 + z, \tag{4.7}$$

where we treat z as a free parameter. According to the value of z, we get an infinity of straight lines, some of which are depicted in Figure 4.2. Moreover, the more z increases, the more the parallel dashed line shifts towards the direction of vector $\mathbf{c}^T = (1, 1)$, which are the coefficients of the variables in the objective functions.

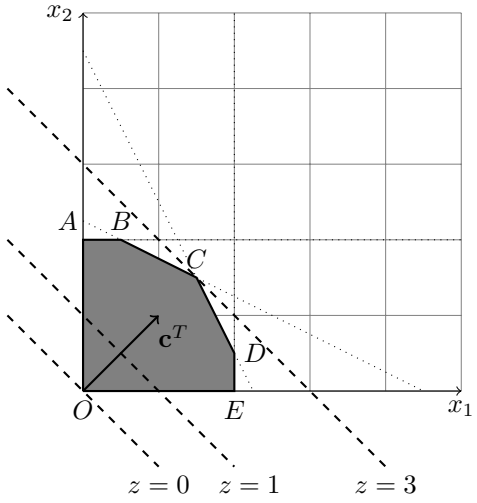


Figure 4.2: Graphical resolution of the fitness problem

We can shift the dashed line as far as it encompasses vertex C (if we went farther, we would not find any feasible solution anymore), which corresponds to $x_1 = x_2 = 1.5$ and $z = 3$. We mark by asterisk * an optimal solution and the optimum (i.e., the value of the objective function corresponding to an optimal solution). Therefore, the optimal solution of the fitness problem $\mathbf{x}^* = (1.5, 1.5)$ consists of 1.5 hours of functional training and 1.5 hours of cardio workout. The optimum z^* consists of 3 hours of overall training at the gym. From Figure 4.2 we note that the optimum occurs at one vertex of polygone $OABCDE$. In Section 4.121 we will provide a formal proof of this behavior, namely that, if an optimal solution of a linear programming problem exists, then it occurs at one vertex of the polytope describe by constraints $\mathbf{Ax} \geq \mathbf{b}$.

4.1 Linear systems

The theory of the simplex algorithm is developed starting from a linear programming problem in standard form. The constraints of a standard form problem are written as $\mathbf{Ax} = \mathbf{b}$, with $\mathbf{x} \geq \mathbf{0}$. Therefore, the basic theory of linear systems and linear algebra is a fundamental starting point for solving a linear programming problem. In this section, we provide the essential linear algebra background required to address the theory of the simplex algorithm. The particularity of linear systems of equations can either have zero, one or infinity solutions. A linear system without any solution is said to be *inconsistent*; otherwise it is *consistent*. The Rouché–Capelli theorem discriminates the nature of a linear system $\mathbf{Ax} = \mathbf{b}$, working on the *rank* of matrix $\mathbf{A}$ and on the rank of the so-called adjoint matrix $[\mathbf{A}|\mathbf{b}]$.

Theorem 1 (The Rouché–Capelli theorem)**.** *Consider the linear system* $\mathbf{Ax} = \mathbf{b}$ *over the field* $\mathbb{E}$*, with* $\mathbf{A} \in \mathbb{E}^{m \times n}$*,* $\mathbf{x} \in \mathbb{E}^n$ *and* $\mathbf{b} \in \mathbb{E}^m$*. Solutions exists if and only if* $rank(\mathbf{A}) = rank([\mathbf{A}|\mathbf{b}])$*. Moreover, if* $rank(\mathbf{A}) = rank([\mathbf{A}|\mathbf{b}])$*, there are* $\infty^{n-rank(\mathbf{A})}$ *solutions to the system* $\mathbf{Ax} = \mathbf{b}$*.*

♣

Intuitively, requiring $rank(\mathbf{A}) = rank([\mathbf{A}|\mathbf{b}])$ means that the linear system does not have degenerate equations of the form $0 = b_i$, where $b_i \neq 0$ is a general i-th element of vector $\mathbf{b}$.

Example 17. *Consider the following linear system:*

$$\begin{cases} 4x_1 - 2x_2 &= 5 \\ -6x_1 + 3x_2 &= 1. \end{cases} \tag{4.8}$$

In order to find the rank of matrix $\mathbf{A}$ *and its adjoint, we reduce the adjoint (which also contains* $\mathbf{A}$*) in echelon form by consistently applying elementary operations on its rows. We have:*

$$\left[\begin{array}{cc|c} 4 & -2 & 5 \\ -6 & 3 & 1 \end{array}\right] \; R_2 + 6/4R_1 \; \sim \left[\begin{array}{cc|c} \textcircled{4} & -2 & 5 \\ 0 & 0 & \textcircled{\frac{5}{2}} \end{array}\right] \tag{4.9}$$

Note that $rank(\mathbf{A}) = 1$ *(there is one pivot corresponding to element 4 in the first row), but* $rank(\mathbf{A}|\mathbf{b}) = 2$*. The second row corresponds to equation* $0x_1 + 0x_2 = \frac{5}{2}$*, which is clearly degenerate and does not have any solution.*

♣

Consider the fitness problem in standard form. To do so, we need to change the sign of the objective function (so that we have a problem of minimum) and introduce slack variables y_1, y_2, y_3 and y_4, respectively associated to constraints (4.2), (4.3), (4.4) and (4.5).

$$\min z = -x_1 - x_2 \tag{4.10}$$

$$\text{s. t.} \quad 2x_1 + 4x_2 + y_1 = 9 \tag{4.11}$$

$$4x_1 + 2x_2 + y_2 = 9 \tag{4.12}$$

$$x_1 + y_3 = 2 \tag{4.13}$$

$$x_2 + y_4 = 2 \tag{4.14}$$

$$x_1, \; x_2, \; y_1, y_2, y_3, y_4 \geq 0. \tag{4.15}$$

The adjoint matrix of system (4.12)–(4.15) is

$$\left[\begin{array}{cccccc|c} 2 & 4 & 1 & 0 & 0 & 0 & 9 \\ 4 & 2 & 0 & 1 & 0 & 0 & 9 \\ 1 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 1 & 2 \end{array}\right], \tag{4.16}$$

which is not in echelon form. Nevertheless, we see that columns C_3, C_4, C_5 and C_6 (those associated to the slack variables) make up an identity matrix. The columns of an identity matrix are the standard unit vectors $\mathbf{e}_i$, which are *linearly independent*. The rank of a matrix is defined as the number of pivots of that matrix in echelon form, but also as the number of linear independent columns making up tre matrix itself. Given an $m \times n$ matrix, its rank can be at most m. Here, the number of rows is 4 ($m = 4$) and we found out that there are 4 linear-independent columns. Therefore, the rank of the adjoint matrix of the fitness problem is 4. The number of columns of matrix $\mathbf{A}$ is $n = 6$. Hence, the Rouché–Capelli theorem tells us that there are $\infty^{6-4} = \infty^2$ solutions of the linear system associated to the fitness problem. This means that we have two degrees of freedom. In this example variables x_1 and x_2 can assume *any* value, say, respetcively, a and b. Thus, the general solution of system (4.10)–(4.15) is

$$\begin{cases} x_1 &= & a \\ x_2 &= & b \\ y_1 &= & 9 - 2a - 4b \\ y_2 &= & 9 - 4a - 2b \\ y_3 &= & 2 - a \\ y_4 &= & 2 - b. \end{cases} \tag{4.17}$$

Among the infinity of values a and b can assume, there is the special combination $a = b = 0$. The general solution (4.121) then becomes

$$\begin{cases} x_1 &= & 0 \\ x_2 &= & 0 \\ y_1 &= & 9 \\ y_2 &= & 9 \\ y_3 &= & 2 \\ y_4 &= & 2. \end{cases} \tag{4.18}$$

This particular solution is said to be a *basic* solution.

Note also that point $(x_1, x_2) = (0, 0)$ is a vertex O of the polygone $OABCD$. In Theorem 4.121 we prove that basic feasible solutions of the linear system $\mathbf{Ax} = \mathbf{b}$ coincide with the vertexes (or extreme points) of the polytope associated to the linear system.

Definition 1 (Basic solution)**.** *Consider the system* $\mathbf{Ax} = \mathbf{b}$ *with* $rank(\mathbf{A}) = rank([\mathbf{A}|\mathbf{b}]) = m$*. Therefore, the Rouché–Capelli theorem ensures us that there are* $\infty^{(n-m)}$ *solutions. Since* $rank(\mathbf{A}) = m$*, we can pick up* m *linearly independent columns and gather them into submatrix* $\mathbf{B}$*, where the letter B stands for basic matrix, because these* m *columns are a* basis *for* $\mathbb{E}^m$*. The remaining columns are gathered into matrix*[1] $\mathbf{D}$*. This partition of matrix* $\mathbf{A}$ *corresponds to the partition of vector* $\mathbf{x}$ *in subvectors* $\mathbf{x_B}$ *and* $\mathbf{x_D}$*, and the initial linear system can now be written as*

[1]Some books like **BAZARAA** use matrix $\mathbf{N}$ because the letter N emphasizes that the columns of this matrix are *non-basic*. The notation $\mathbf{D}$ of the non-basic matrix is mainly due to **LUENBERGER**.

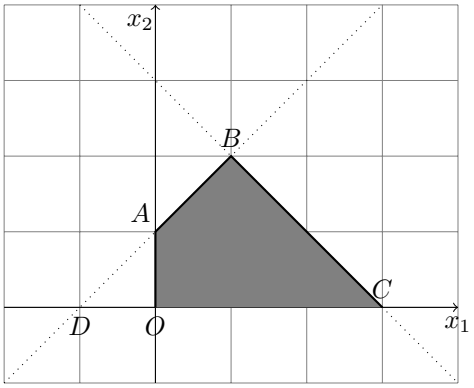


Figure 4.3: The feasible of model (4.20)–(4.23).

$$\mathbf{B}\mathbf{x}_\mathbf{B} + \mathbf{D}\mathbf{x}_\mathbf{D} = \mathbf{b}. \tag{4.19}$$

The particular solution $(\mathbf{x_B}, \mathbf{x_D}) = \left(\mathbf{B}^{-1}\mathbf{b}, \mathbf{0}\right)$ *of this system is said to be a basic solution.*

Not necessarily all the basic components are non-zero. If this is the case, we have a *degenerate basic solutions*. Finally, the system $\mathbf{Ax} = \mathbf{b}$ can have solutions with negative components (see Example 4.121), thus violating the constraints $\mathbf{x} \geq \mathbf{0}$. A basic solution with at least one negative component is an *infeasible basic solution*. Otherwise it is a *feasible basic solution*.

Example 18. *Consider the followin Linear Programming model:*

$$\begin{aligned}
\max \quad & x_1 && (4.20)\\
\textit{s. t.} \quad & x_1 - x_2 \geq -1 && (4.21)\\
& x_1 + x_2 \leq 3 && (4.22)\\
& x_1, \ x_2 \geq 0. && (4.23)
\end{aligned}$$

The domain of its feasible solution is depicted in Figure 4.3.

The model in standard form is

$$\begin{aligned}
\max \quad & -x_1 && (4.24)\\
\textit{s. t.} \quad & -x_1 + x_2 + y_1 = 1 && (4.25)\\
& x_1 + x_2 + y_2 = 3 && (4.26)\\
& x_1, \ x_2, \ y_1, \ y_2 \geq 0. && (4.27)
\end{aligned}$$

The solution $(x_1, x_2, y_1, y_2)^T = (-1, 0, 0, 4)$ *is basic but it is not feasible because* $x_1 < 0$*. This infeasible basic solution corresponds to point* $D(-1, 0)$*, which is the*

intersection of the line $x_1 - x_2 + 1 = 0$ *with the line* $x_2 = 0$*. The other basic solutions are points* $O(0,\, 0)$*,* $A(0,\, 1)$*,* $B(1,\, 2)$ *and* $C(3,\, 0)$*. These basic solutions are all feasible.*

♣

4.2 The fundamental theorem of Linear Programming

In the graphical resolution of the fitness problem, we came to the conclusion that the optimum lies on one vertex of the polyhedron of the constraints of the problem itself. The fundamental theorem of Linear Programming proves this behavior of linear programming problems.

Theorem 2 (The fundamental theorem of Linear Programming)**.** *Given a linear programming model in standard form*

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} && (4.28)\\ \textit{s. t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} && (4.29)\\ & \mathbf{x} \geq \mathbf{0}, && (4.30) \end{aligned}$$

1. *if exists a feasible solution, then it also exists a basic feasible solution;*
2. *if exists an optimal feasible solution, then it also exists an optimal basic feasible solution.*

Proof.
Case 1

Consider a general solution $\mathbf{x}$ of the linear programming problem. Without loss of generality, we can assume that the first p components of $\mathbf{x}$ are positive and the remaining $n - p$ components are equal to zero:

$$\mathbf{x}^T = (x_1,\, x_2,\, \ldots,\, x_p,\, 0,\, \ldots,\, 0)\,. \tag{4.31}$$

Note that the case where all the components are positive corresponds to $p = n$. If, in a solution with p positive component, one zero component appears before position p, then it is possible to change its position by placing it at the last position of vector $\mathbf{x}$ and changing the matrix $\mathbf{A}$ and the cost vector $\mathbf{c}$ accordingly (see Example 19). If $p = m$, the solution is already basic and we are done. If $p < m$, since matrix $\mathbf{A}$ has rank m, we can pick up $p - m$ more linear independent columns of $\mathbf{A}$ and set the corresponding components to 0 and we have a degenerate basic feasible solution. Finally, if $p > m$, the corresponding vector of matrix $\mathbf{A}$ are linearly *dependent*. Therefore, there exists a linear combination with coefficients $y_1,\, y_2,\, \ldots,\, y_p$, with at least one of them different to zero, such that

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \ldots + y_p\mathbf{a}_p = \mathbf{0}. \tag{4.32}$$

At the same time, $\mathbf{x}$ is a solution of the system, therefore,

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_p\mathbf{a}_p + 0\mathbf{a}_{p+1} + \ldots + 0\mathbf{a}_n = \mathbf{b}, \tag{4.33}$$

or simply

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_p\mathbf{a}_p = \mathbf{b}, \tag{4.34}$$

If we multiply (4.121) by a scalar $\varepsilon > 0$ and subtract this new equation from (4.121), we get the following identity:

$$(x_1 - \varepsilon y_1)\mathbf{a}_1 + (x_2 - \varepsilon y_2)\mathbf{a}_2 + \ldots + (x_p - \varepsilon y_p)\mathbf{a}_p = \mathbf{b}, \tag{4.35}$$

Note that (4.35) can also be expressed as

$$\mathbf{A}(\mathbf{x} - \varepsilon\mathbf{y}) = \mathbf{b}, \tag{4.36}$$

meaning that $\mathbf{x} - \varepsilon\mathbf{y}$ is also a solution of the problem.

Note also that at least one of the coefficients y_i is positive. In fact, by definition of linear combination, there is at least one non-zero coefficient in (4.121). If all the non-zero coefficients are negative, it is enough to multiply (4.121) by -1 and then all the non-zero coefficient become positive. Therefore, there is at least one coefficient $x_i - \varepsilon y_i$ which becomes zero for an appropriate value of θ. This value is clearly given by

$$\varepsilon = \min_{i\in\{1,2,\ldots,p\}} \left\{ \frac{x_i}{y_i} \middle| y_i > 0 \right\}. \tag{4.37}$$

Substituting this value into (4.35) or (4.36), we see that $\mathbf{x} - \varepsilon\mathbf{y}$ has $p-1$ non-zero components. If we iteratively repeat this process, we obtain a basic feasible solution.

Case 2

If we have an optimal solution with $p = m$, the solution is already basic and we are done. If we have an optimal solution with $p < m$, again we can buil a degenerate optimal basic feasible solution by setting to 0 $m - p$ independent column vectors of matrix $\mathbf{A}$. The non-trivial case to prove is when $p > m$. The proof holds like in case 1, but we have to ensure that the new solution $\mathbf{x} - \varepsilon\mathbf{y}$ with $p-1$ non-zero components, is also feasible and optimal. Since the initial solution $\mathbf{x}$ is optimal, we must ensure that $\mathbf{c}^T\mathbf{x} = \mathbf{c}^T(\mathbf{x} - \varepsilon\mathbf{y})$, which implies

$$\mathbf{c}^T\mathbf{y} = 0. \tag{4.38}$$

Finally, feasibility is guaranteed noting that $\mathbf{x} - \varepsilon\mathbf{y}$ means moving away from $\mathbf{x}$ towards $-\varepsilon\mathbf{y}$ (see Section4.3). Therefore, if we choose a small-enough value for ε we can move away from $\mathbf{x}$ without violating the constraints of the problem. □

The great advantage of the fundamental theorem of Linear Programming is that we move from searching among ∞^{n-m} solutions of the system $\mathbf{Ax} = \mathbf{b}$ to a limited number of basic solutions of the same system. This number is of the same order of the number combinations of m columns over n, which is

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}. \tag{4.39}$$

Nevertheless, althought limited, the number of solutions to explore can still be huge. The simplex algorithm tries to overcome this issue. Although its efficiency is not polynomial in the worst case (see Section 4.121), it is polynomial in the average case.

Example 19. *Let us build a solution of the fitness problem with 5 non-zero variables over the 6 variables x_1, x_2, y_1, y_2, y_3, y_4, and with the variable in the second position (i.e. x_2) equal to 0. To do so, let's set in* (4.17) $x_1 = 1$ *and* $x_2 = 0$*. Substituting these values in the expressions for the dependent variables, we find*

$$(x_1, x_2, y_1, y_2, y_3, y_4) = (1, 0, 7, 5, 1, 2). \tag{4.40}$$

The number p of non-zero components is 5, but the first five position are not all non-zero. Therefore, we should move x_2 at the end of the vector and consider

$$(x_1, y_1, y_2, y_3, y_4, x_2) = (1, 7, 5, 1, 2, 0). \tag{4.41}$$

But things won't work if we limit ourselves to this substitution. In fact, we have to change the fitness model (4.10)–(4.15) *accordingly as follows:*

$$\min z = -x_1 \qquad -x_2 \tag{4.42}$$

$$s.\ t. \quad 2x_1 + y_1 \qquad +4x_2 = 9 \tag{4.43}$$

$$4x_1 + \quad y_2 \qquad 2x_2 = 9 \tag{4.44}$$

$$x_1 + \qquad y_3 \qquad = 2 \tag{4.45}$$

$$y_4 + x_2 = 2 \tag{4.46}$$

$$x_1,\ y_1, y_2, y_3, y_4,\ x_2 \geq 0 \tag{4.47}$$

The new adjoint matrix of system (4.22)–(4.47) *is*

$$[\mathbf{A}|\mathbf{b}] = \left[\begin{array}{cccccc|c} 2 & 1 & 0 & 0 & 0 & 4 & 9 \\ 4 & 0 & 1 & 0 & 0 & 2 & 9 \\ 1 & 0 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \end{array}\right], \tag{4.48}$$

Note that, while vector $\mathbf{b}$ *has not changed, the matrix* $\mathbf{A}$ *is now different. Same for the cost vector, which now is* $\mathbf{c}^T = (-1, 0, 0, 0, 0, -1)$.

♣

4.3 A graphical interpretation

In this section, we wolud like to show an interesting graphical interpretation of the fundamental theorem of Linear Programming. This interpretation will also be useful in the development of the simplex algorithm. To do so, we define the concept of *feasible direction* in a polytope. Given a point $\mathbf{x}$ of a polytope, a vector $\mathbf{d}$ is a feasible direction for point $\mathbf{x}$ if exists a positive scalar θ such that the point $\mathbf{x}+\theta\mathbf{d}$ belongs to the polytope. Since $\mathbf{x}+\theta\mathbf{d}$ belongs to the polytope, it must satisfy $\mathbf{A}(\mathbf{x}+\theta\mathbf{d}) = \mathbf{b}$. But also $\mathbf{x}$ belongs to the polytope, and this implies $\mathbf{Ax} = \mathbf{b}$. Thus, we have

$$\begin{aligned} \mathbf{A}(\mathbf{x}+\theta\mathbf{d}) &= \mathbf{b} && (4.49)\\ \mathbf{Ax}+\theta\mathbf{Ad} &= \mathbf{b} && (4.50)\\ \mathbf{b}+\theta\mathbf{Ad} &= \mathbf{b}, && (4.51) \end{aligned}$$

which, as $\theta > 0$, implies

$$\mathbf{Ad} = \mathbf{0}. \tag{4.52}$$

Note that (4.49) becomes (4.36) when we set $\theta = \varepsilon$ and $\mathbf{d} = -\mathbf{y}$. Moreover, if we set $\mathbf{d}^T = (-y_1, -y_2, \ldots, -y_p, 0, \ldots, 0)$, then (4.52) becomes (4.32).

Example 20. *Consider the following Linear Programming model:*

$$\begin{aligned} \max \quad & x_1 + x_2 && (4.53)\\ \textit{s. t.} \quad & x_1 + x_2 \leq 4 && (4.54)\\ & x_2 \leq 3 && (4.55)\\ & x_1,\ x_2 \geq 0, && (4.56) \end{aligned}$$

which in standard form becomes

$$\begin{aligned} \min \quad & -x_1 - x_2 && (4.57)\\ \textit{s. t.} \quad & x_1 + x_2 + y_1 = 4 && (4.58)\\ & x_2 + y_2 = 3 && (4.59)\\ & x_1,\ x_2,\ y_1,\ y_2 \geq 0. && (4.60) \end{aligned}$$

The corresponding $\mathbf{A}$ *matrix has rank 2 because the last two columns make up the identity matrix and the number of rows is equal to 2. Thus, a non-degenerate basic solution has two positive components (corresponding to linearly independent vectors of matrix* $\mathbf{A}$*) and the remaining two components equal to 0. Consider point* P *in Figure 4.4. It corresponds to the solution of the system* $\mathbf{x}_P^T = (1, 1, 2, 2)$ *Applying* (4.34) *to this particular case, this means that*

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}. \tag{4.61}$$

Since the four columns corresponding to the non-zero components of $\mathbf{x}_P$ *are linearly dependent, there exist some non-zero* d_1, d_2, d_3 *and* d_4 *such that*

$$d_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + d_2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + d_3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + d_4 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.62}$$

We can arbitrarily choose, for instance, $\mathbf{d}^T = (d_1\ d_2,\ d_3,\ d_4) = (2,\ -1,\ -1,\ 1)$. *Substituting these values in* (4.62)*, multiplying by* $\theta > 0$ *and summing*[2] *to* (4.62)*, we get:*

$$(1+2\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + (1-\theta) \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (2-\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + (2+\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}. \tag{4.63}$$

If θ *is progressively increased starting from 0, the first component to become zero is* x_2*, and this holds for* $\theta = 1$. *When* $\theta = 1$ *we have*

$$3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \tag{4.64}$$

which corresponds to solution $\mathbf{x}_R^T = (3,\ 0,\ 1,\ 3)$*, which corresponds to point* $R(3,\ 0)$ *in Figure 4.4 and has 3 non-zero components. Figure 4.4 also shows a very interesting geometrical interpretation. Moving from* P *to* R *means summing to vector* $\mathbf{x}_P$ *the vector* $\theta\mathbf{d}$ *(in this examples* $\theta = 1$*). The resulting sum is vector* $\mathbf{x}_R$*, which denotes point* R. *Note also that* $\mathbf{c}^T\mathbf{x}_P = -1 - 1 = -2 > \mathbf{c}^T\mathbf{x}_R = -3$. *A similar interpretation holds for the second part of the fundamental theory of linear programming. Nevertheless, vector* $\mathbf{d}$ *must be built with particular care. If we try to solve graphically the problem of this example, we notice that the line* $x_2 = -x_1 - z$ *includes all the edge* BC *rather than just one vertex of the polytope, and this happens when* $z = -4$. *This means that all the points lying on this edge are optimal solutions with* $z^* = -4$. *Among these solutions, we have point* Q*, which corresponds to the solution* $(x_1,\ x_2,\ y_1,\ y_2)^T = (2,\ 2,\ 0,\ 1)$*, which is not a basic solutions because the number of non-zero components is 3. If we want to move towards one of the two vertexes (say* C*), when we have to add to vector* $\mathbf{x}_Q$ *vector* $\mathbf{d}$*, modulated by an appropriate value* θ. *This means that we have to move along the edge* BC *which is perpendicular to vector* $\mathbf{c}$. *Therefore,* $\mathbf{d}$ *must be chosen such that* $\mathbf{c}^T\mathbf{d} = 0$*; which is* (4.38)*. For instance, we can choose* $d_1 = 1$ *and* $d_2 = 2$. *This choice corresponds to vector* $\mathbf{d}$ *in Figure 4.5. The remaining components* $d_3 and d_4$ *cannot be visualized in Figure 4.5, but are chosen considering that* $\mathbf{Ad} = \mathbf{0}$ *(due to* (4.52)*) and* $d_1\mathbf{a}_1 + d_2\mathbf{a}_2 + d_4\mathbf{a}_4 = \mathbf{0}$ *(due to* (4.32)*). This implies* $d_3 = 0$ *and* $d_4 = 1$. *Similarly to the first part of this example, we get*

$$(2+\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + (2-\theta) \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + (1+\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}. \tag{4.65}$$

[2] We are summing because we are using the notation $\mathbf{x}+\theta\mathbf{d}$ of this section, rather than the notation $\mathbf{x} - \varepsilon\mathbf{y}$ of Section 4.2

When $\theta = 2$, the second component becomes zero, and we get the solution $(x_1,\, x_2,\, y_1,\, y_2)^T = (4,\, 0,\, 0,\, 3)$, which is basic and correspond to vertex C. Note also that $\mathbf{c}^T\mathbf{x}_Q = -2 - 2 = \mathbf{c}^T\mathbf{x}_C = -4$. Geometrically speaking, this means summing vector $2\mathbf{d}$ to vector $\mathbf{x}_Q$. In this way we move along edge BC as far as we reach vertex C.

4.4 Basic solutions and extreme points

Theorem 3 (Equivalence theorem)**.** $\mathbf{x}$ *is a basic feasible solution if and only if is an extreme point.*

Proof. Basic feasible solution $\rightarrow$ extreme point

If $\mathbf{x}$ is a basic solution then, without loss of generality, we can write

$$\mathbf{x} = (x_1,\, x_2,\, \ldots,\, x_m,\, 0,\, \ldots,\, 0) \tag{4.66}$$

and

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_m\mathbf{a}_m = \mathbf{b}. \tag{4.67}$$

Let us suppose that $\mathbf{x}$ can be expressed as a convex combination of vectors $\mathbf{y}$ and $\mathbf{z}$, then there exists $\alpha \in (0,\, 1)$ such that

$$\mathbf{x} = \alpha\mathbf{y} + (1 - \alpha)\mathbf{z}. \tag{4.68}$$

The i-th component of vector $\mathbf{x}$ with $m < i \leq n$ is 0. Equation (4.68) applied to this component is

$$0 = \alpha y_i + (1 - \alpha)z_i. \tag{4.69}$$

Since $\alpha \in [0,\, 1]$ and $y_i,\, z_i \geq 0$, then the only values which can satisfy (4.69) are $y_i = z_i = 0$. This means that the last $n - m$ components of vectors $\mathbf{y}$ and $\mathbf{z}$ are equal to 0. Since these vectors are also solutions of the problem, we have that

$$y_1\mathbf{a}_1 + y_2\mathbf{a}_2 + \ldots + y_m\mathbf{a}_m = \mathbf{b} \tag{4.70}$$

and

$$z_1\mathbf{a}_1 + z_2\mathbf{a}_2 + \ldots + z_m\mathbf{a}_m = \mathbf{b}. \tag{4.71}$$

Since vectors $\mathbf{a}_1,\, \mathbf{a}_2,\, \ldots,\, \mathbf{a}_m$ are linearly independent, then there exists a unique linear combinations able to express vector $\mathbf{b}$ in terms of these vectors. Therefore the components x_i, y_i and z_i respectively in (4.67), (4.70) and (4.71) must be equal. In fact, if we subtract (4.70) from (4.67), we have

$$(x_1 - y_1)\mathbf{a}_1 + (x_2 - y_2)\mathbf{a}_2 + \ldots + (x_m - y_m)\mathbf{a}_m = \mathbf{0}. \tag{4.72}$$

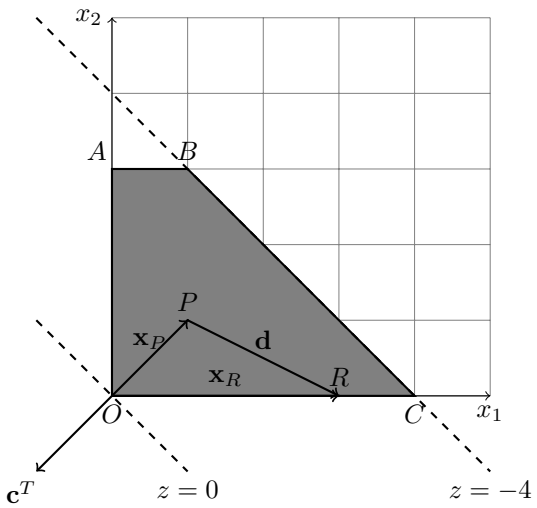


Figure 4.4: A geometrical interpratation of the first part of the fundamental theorem of Linear Programming.

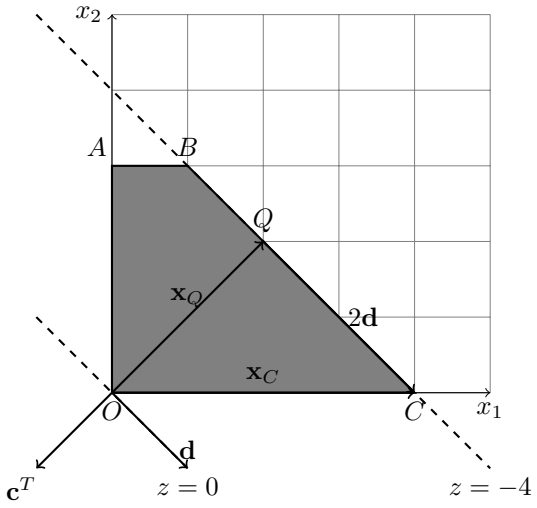


Figure 4.5: A geometrical interpratation of the second part of the fundamental theorem of Linear Programming.

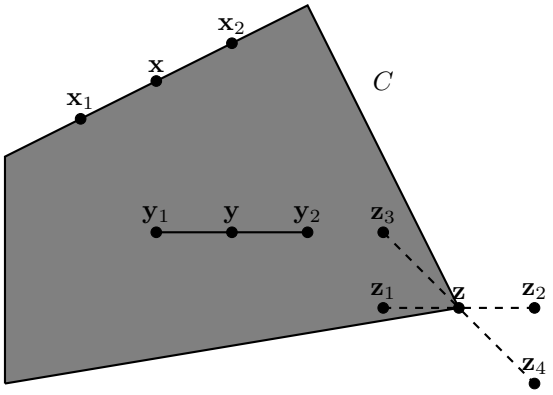


Figure 4.6: Point $\mathbf{x}$ is not an extreme point because lies within the segment connecting $\mathbf{x}_1$ and $\mathbf{x}_2$, both belonging to the convex set C. A similar conclusion holds for point $\mathbf{y}$. Point $\mathbf{z}$ is an extreme point because any point is picked within C, the part of the segment beyond $\mathbf{z}$ is outside C.

Since vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are linearly independent, then the only linear combination which allows us to find the $\mathbf{0}$ vector is the one with the coefficients multiplying each of the vectors $\mathbf{a}_i$ equal to zero, with $i \in \{1, \ldots, m\}$. Thus, it must be $x_i - y_i = 0$, or $x_i = y_i$, for all $i \in \{1, \ldots, m\}$. Combining in the same fashion (4.67) with (4.71) and (4.70) with (4.71), we also find respectively that $x_i = z_i$ and $y_i = z_i$ for all $i \in \{1, \ldots, m\}$.

Since the last $n - m$ components are also all equal, we must have $\mathbf{x} = \mathbf{y} = \mathbf{z}$. Therefore $\mathbf{x}$ cannot be expressed as a convex combination of vectors $\mathbf{y}$ and $\mathbf{z}$. It follows that $\mathbf{x}$ is an extreme point.

Extreme point $\rightarrow$ basic feasible solution

Without loss of generality, we can assume that the non-zero components of the extreme point $\mathbf{x}$ are the first k components; i.e.

$$x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \ldots + x_k\mathbf{a}_k = \mathbf{b}. \tag{4.73}$$

If $\mathbf{x}$ is a basic feasible solution, vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ must be linearly independent. If, by contradiction, these vectors were linearly independent, then there would exists $d_1, d_2, \ldots, d_k$, not all zero, such that

$$d_1\mathbf{a}_1 + d_2\mathbf{a}_2 + \ldots + d_k\mathbf{a}_k = \mathbf{0}. \tag{4.74}$$

We build vector $\mathbf{d}$ setting to 0 the last $n - k$ components as follows:

$$\mathbf{d} = (d_1, d_2, \ldots, d_k, 0, \ldots, 0). \tag{4.75}$$

Then (4.74) becomes

$$\mathbf{A}\mathbf{d} = \mathbf{0}. \tag{4.76}$$

Since the first k components of $\mathbf{x}$ are positive, we can find an appropriate $\theta > 0$ such that

$$\mathbf{x} + \theta\mathbf{d} \geq \mathbf{0} \qquad \text{and} \qquad \mathbf{x} - \theta\mathbf{d} \geq \mathbf{0}. \tag{4.77}$$

We have

$$\mathbf{A}(\mathbf{x} \pm \theta\mathbf{d}) = \mathbf{A}\mathbf{x} \pm \theta\mathbf{A}\mathbf{d} = \mathbf{b} + \theta\mathbf{0} = \mathbf{b}. \tag{4.78}$$

This means that $\mathbf{x} \pm \theta\mathbf{d}$ are also two solutions of the problem. But $\mathbf{x} = \frac{1}{2}(\mathbf{x} + \theta\mathbf{d}) + \frac{1}{2}(\mathbf{x} - \theta\mathbf{d})$, which means that $\mathbf{x}$ can be expressed as a convex combination combination of these two solutions, contradicting the hypothesis that $\mathbf{x}$ is an extreme point. Therefore the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ corresponding to the k nonzero components of $\mathbf{x}$ must be linearly independent. Since the number of linearly independent vectors can be at most m, it follows that $k \leq m$, which means that $\mathbf{x}$ must be a basic feasible solution. Note that if $k < m$, then $\mathbf{x}$ is a degenerate basic feasible solution. □

4.5 Other examples

The domains of the examples presented so far have always been bounded. Therefore, the polytope of the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$ is called *polyhedron*. We wish to conclude this chapter showing two examples of linear programming problems where the polytope associated to the constraints of the problem is unbounded. Two situations can occur:

1. there is still an optimum
2. there is not an optimum and the problem is unbounded because the objective function of the min (max) can be arbitrarily small (large).

Example 4.121 shows these two particular cases.

Example 21. *Consider the following linear programming problems:*

$$\begin{aligned}
\min \quad & z = x_1 + x_2 && (4.79)\\
\textit{s. t.} \quad & 2x_1 - x_2 \geq 0 && (4.80)\\
& x_1 - 2x_2 \geq 0 && (4.81)\\
& x_1 - 4x_2 \geq 0 && (4.82)\\
& x_1 \geq 0 && (4.83)\\
& x_2 \geq 0. && (4.84)\\
& && (4.85)
\end{aligned}$$

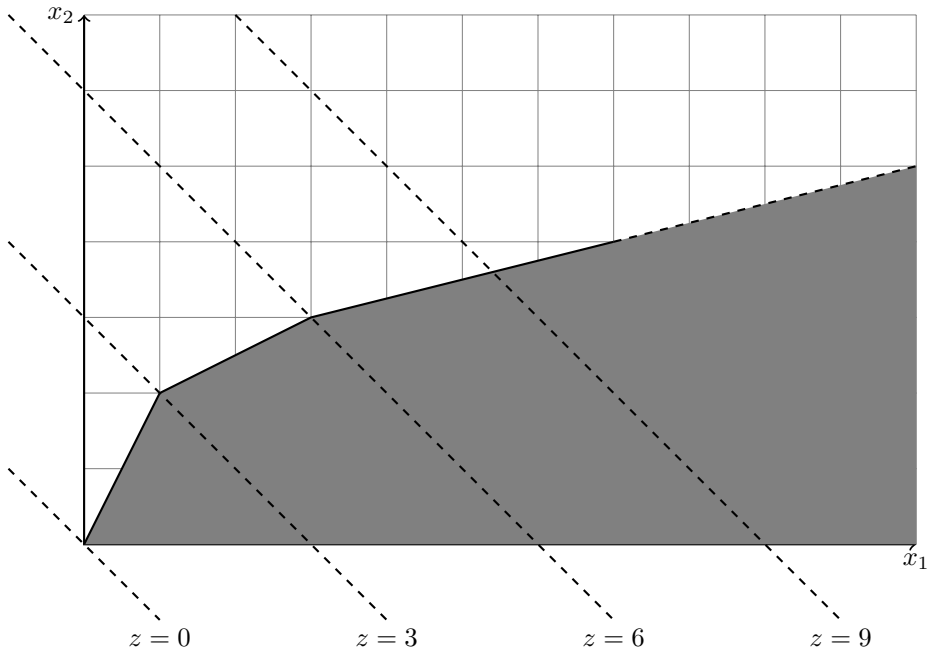


Figure 4.7: Example of two linear programming problems with unbounded domain.

$$\begin{aligned}
\max \quad & z = x_1 + x_2 && (4.86)\\
\textit{s. t.} \quad & 2x_1 - x_2 \geq 0 && (4.87)\\
& x_1 - 2x_2 \geq 0 && (4.88)\\
& x_1 - 4x_2 \geq 0 && (4.89)\\
& x_1 \geq 0 && (4.90)\\
& x_2 \geq 0. && (4.91)\\
& && (4.92)
\end{aligned}$$

The two linear programming problems have the same domain, which is depicted in Figure 4.7. The domain is clearly unbounded. For the minimization problem (4.79)–(4.84) *we have a minimum for* $z = 0$*. Vice versa we can see from Figure 4.7 that for the maximization problem* (4.86)–(4.91)*, the objective function* $z = x_1 + x_2$ *can be arbitrarily large.*

4.6 The canonical form

If we have to solve the linear system $\mathbf{Ax} = \mathbf{b}$, we can perform elementary operations on the adjoint matrix applying the well-known (forward) Gaussian algorithm. The resulting matrix is an *échelon* matrix, where the elements below each pivot are equal to zero. The system can then be solved starting from the last equation (in the last row of the matrix, assuming null rows were deleted) which will be in the form $\alpha x_m = \beta$. Solving for x_m we can replace this value in the previous equation and solve for x_{m-1} in the next to last row. Iterating this procedure, we can go up to x_1 and we are done. This procedure is called the backward Gaussian algorithm.

Requiring a system to be in canonical form is more restrictive than expressing it in echelon form. In fact, in addition to the echelon form, we must have:

1. each pivot equal to 1
2. each element above a pivot equal to 0.

To show the difference between the échelon and canonical form and to recall the forward-backward Gaussian algorithm, let us solve the linear system in Example 22

Example 22. *Consider the linear system*

$$\begin{cases} 3x_1 - x_2 + 6x_3 = 1 \\ 6x_1 + 3x_2 + 10x_3 = 3. \end{cases} \tag{4.93}$$

In matrix form the system becomes

$$[\mathbf{A}|\,\mathbf{b}] = \left[\begin{array}{ccc|c} 3 & -1 & 6 & 1 \\ 6 & 3 & 10 & 3 \end{array}\right] \begin{array}{l} \\ R_2 - 2R_1 \end{array} \sim \left[\begin{array}{ccc|c} \textcircled{3} & -1 & 6 & 1 \\ 0 & \textcircled{5} & -2 & 1 \end{array}\right], \tag{4.94}$$

where the circled elements are the pivots *of the matrix.*

From this échelon form we can solve for x_2 from the second (the last one) equation. This means starting the backward procedure of the Gauss algorithm. We have:

$$5x_2 - 2x_3 = 1 \Longrightarrow 5x_2 = 1 + 2x_3 \Longrightarrow x_2 = \frac{1}{5} + \frac{2}{5}x_3. \tag{4.95}$$

$$3x_1 - x_2 + 6x_3 = 1 \Longrightarrow 3x_1 = 1 + x_2 - 6x_3 \Longrightarrow x_1 = \frac{1}{3} + \frac{1}{3}x_2 - 2x_3 \Longrightarrow$$

$$\Longrightarrow x_1 = \frac{1}{3} + \frac{1}{3}\left(\frac{1}{5} + \frac{2}{5}x_3\right) - 2x_3 \Longrightarrow x_1 = \frac{2}{5} - \frac{28}{15}x_3. \tag{4.96}$$

Thus, the general solution of system 4.121 is

$$\left(\frac{2}{5}-\frac{28}{15}t,\ \frac{1}{5}+\frac{2}{5}t,\ t\right), \tag{4.97}$$

with $t \in \mathbb{R}$.

$$\left[\begin{array}{ccc|c} \textcircled{3} & -1 & 6 & 1 \\ 0 & \textcircled{5} & -2 & 1 \end{array}\right] \begin{array}{l} R_1/3 \\ R_2/5 \end{array} \sim \left[\begin{array}{ccc|c} \textcircled{1} & -\frac{1}{3} & 2 & \frac{1}{3} \\ 0 & \textcircled{1} & -\frac{2}{5} & \frac{1}{5} \end{array}\right] R_1+\tfrac{1}{3}R_2 \sim \tag{4.98}$$

$$\sim \left[\begin{array}{ccc|c} \textcircled{1} & 0 & \frac{28}{15} & \frac{2}{5} \\ 0 & \textcircled{1} & -\frac{2}{5} & \frac{1}{5} \end{array}\right]. \tag{4.99}$$

4.121 is the canonical form of system 4.121 and leads straightforwardly to the solution set 4.121.

♣

An important remark

Note that the columns corresponding to the basic variables of the linear system in canonical form in (4.121) form an identity matrix. However, this identity matrix does not necessarily have to appear in the first m columns, but it can also be scattered within the tableau. For instance, similarly to what we did in Example 19, if we swap the order of the variables from $(x_1,\ x_2,\ x_3)$ to $(x_1,\ x_3,\ x_2)$, then the original system becomes

$$\left\{\begin{array}{rcrcrcl} 3x_1 & + & 6x_3 & - & x_2 & = & 1 \\ 6x_1 & + & 10x_3 & + & 3x_2 & = & 3. \end{array}\right. \tag{4.100}$$

With the same operations performed so far, the canonical form with x_1 and x_2 in the basis becomes

$$\left[\begin{array}{ccc|c} \textcircled{1} & \frac{28}{15} & 0 & \frac{2}{5} \\ 0 & -\frac{2}{5} & \textcircled{1} & \frac{1}{5} \end{array}\right]. \tag{4.101}$$

Here, we want to generalize what we did in Example 22 and provide a particular interpretation of the coefficients of a system in canonical form.

$$\mathbf{Ax} = \mathbf{b} \tag{4.102}$$

Let us assume that system (4.102) has solutions. From the Rouché-Capelli theorem, this means that $rank(\mathbf{A}) = rank([\mathbf{A} \mid \mathbf{b}])$. Moreover, we assume that $rank(\mathbf{A}) = m$, i.e., all the rows of matrix $\mathbf{A}$ are linearly independent.

Without loss of generality, we assume that the first m variables associated to the first m columns of matrix $\mathbf{A}$ form a basis. We group these variables and these columns respectively into vector $\mathbf{x_B} = [x_1, \ldots, x_m]^T$ and into matrix $\mathbf{B} = [\mathbf{a}_1 \mid \ldots \mid \mathbf{a}_m]$. Similarly, we can group the non-basic variables into vector $\mathbf{x_D} =$

$[x_{m+1}, \ldots, x_n]^T$ and the corresponding columns of matrix $\mathbf{A}$ into matrix $\mathbf{D} = [\mathbf{a}_{m+1}, \ldots, \mathbf{a}_n]$. Thus, $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_D \end{bmatrix}$, $\mathbf{A} = [\mathbf{B} \,|\, \mathbf{D}]$, and (4.102) becomes

$$\mathbf{B}\mathbf{x}_\mathbf{B} + \mathbf{D}\mathbf{x}_\mathbf{D} = \mathbf{b}. \tag{4.103}$$

Since the m columns of matrix $\mathbf{B}$ form a basis, they are linearly independent and the inverse matrix $\mathbf{B}^{-1}$ exists. If we multiply each side of (4.103) by $\mathbf{B}^{-1}$, we get

$$\mathbf{I}\mathbf{x}_\mathbf{B} + \mathbf{B}^{-1}\mathbf{D}\mathbf{x}_\mathbf{D} = \mathbf{B}^{-1}\mathbf{b}. \tag{4.104}$$

Let $\mathbf{Y}$ be the matrix $\mathbf{B}^{-1}\mathbf{D}$ and $\mathbf{y}_0$ be the vector $\mathbf{B}^{-1}\mathbf{b}$, then we have

$$\mathbf{I}\mathbf{x}_\mathbf{B} + \mathbf{Y}\mathbf{x}_\mathbf{D} = \mathbf{y}_0, \tag{4.105}$$

which corresponds to the following *tableau*:

$$\begin{array}{cccccccc} 1 & 0 & \ldots & 0 & y_{1,\,m+1} & \cdots & y_{1n} & y_{10} \\ 0 & 1 & \ldots & 0 & y_{2,\,m+1} & \cdots & y_{2n} & y_{20} \\ \vdots & & \ddots & \vdots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 & y_{m,\,m+1} & \cdots & y_{mn} & y_{m0} \end{array} \tag{4.106}$$

The vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are linearly independent and span $\mathbb{R}^m$. This means that, for each vector $\mathbf{a}_q \in \mathbb{R}^m$, there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ such that $\mathbf{a}_q \in \mathbb{R}^m$ can be expressed as a linear combination of the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$:

$$\mathbf{a}_q = \sum_{i=1}^{m} \alpha_i \mathbf{a}_i. \tag{4.107}$$

The coefficients $\mathbf{a}_1, \ldots, \mathbf{a}_m$ can be determined as follows: we first write all the components making up the involved vectors in (4.107):

$$\begin{bmatrix} a_{1,\,q} \\ a_{2,\,q} \\ \vdots \\ a_{m,\,q} \end{bmatrix} = \begin{bmatrix} a_{1,\,1} \\ a_{2,\,1} \\ \vdots \\ a_{m,\,1} \end{bmatrix} \alpha_1 + \begin{bmatrix} a_{1,\,2} \\ a_{2,\,2} \\ \vdots \\ a_{m,\,2} \end{bmatrix} \alpha_2 + \ldots + \begin{bmatrix} a_{1,\,m} \\ a_{2,\,m} \\ \vdots \\ a_{m,\,m} \end{bmatrix} \alpha_m. \tag{4.108}$$

If we gather all the coefficients $\alpha_1, \ldots, \alpha_m$ into a single vector, (4.108) becomes

$$\begin{bmatrix} a_{1,\,q} \\ a_{2,\,q} \\ \vdots \\ a_{m,\,q} \end{bmatrix} = \begin{bmatrix} a_{1,\,1} & a_{1,\,2} & \ldots & a_{1,\,m} \\ a_{2,\,1} & a_{2,\,2} & \ldots & a_{2,\,m} \\ \vdots & \vdots & & \vdots \\ a_{m,\,1} & a_{m,\,2} & \ldots & a_{m,\,m} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}, \tag{4.109}$$

which, in matrix form, becomes

$$\mathbf{a}_q = [\mathbf{a}_1 \,|\, \mathbf{a}_2 \,|\, \ldots \,|\, \mathbf{a}_m]\, \boldsymbol{\alpha}. \tag{4.110}$$

But $[\mathbf{a}_1 \,|\, \ldots \,|\, \mathbf{a}_m]$ is the matrix $\mathbf{B}$, thus we have

$$\mathbf{a}_q = \mathbf{B}\boldsymbol{\alpha}. \tag{4.111}$$

Since the matrix $\mathbf{B}$ is invertible, the desired values for the coefficients $\alpha_1, \ldots, \alpha_m$ are given by

$$\boldsymbol{\alpha} = \mathbf{B}^{-1}\mathbf{a}_q. \tag{4.112}$$

But $\mathbf{B}^{-1}\mathbf{a}_q$ is the q-th column of matrix $\mathbf{B}^{-1}\mathbf{D}$, i.e., of matrix $\mathbf{Y}$. Therefore, we have the following fundamental property:

> Given a linear system $\mathbf{Ax} = \mathbf{b}$ with m equations, n unknowns, with non-trivial solutions, full rank m, basic variables $x_1, \ldots, x_m$ with corresponding matrix $\mathbf{B} = [\mathbf{a}_1 \ldots \mathbf{a}_m]$ and non-basic variables $x_{m+1}, \ldots, x_n$ with corresponding matrix $\mathbf{D} = [\mathbf{a}_{m+1} \ldots \mathbf{a}_n]$, then each non-basic vector $\mathbf{a}_q$ (with $q \in \{m+1, \ldots, n\}$) of matrix $\mathbf{D}$ can be expressed as a linear combination of the basic vectors in matrix $\mathbf{B}$. The coefficients of this linear combination are the same coefficients that appear in the q^{th} column of the system in canonical form.

4.7 Pivots

One advantage of the canonical form is that it allows us to move easily from a basic solution of the linear system $\mathbf{Ax} = \mathbf{b}$ to another one. By construction, we always have an identity matrix in a tableau of a linear system in canonical form. The variables associated to the columns making up the identity matrix are also those that are in the basis. All the remaining variables are non-basic variables.

The linear system in (4.121) is in canonical form. In this canonical form, x_1 and x_2 are the basic variables, while x_3 is a non-basic variable. To better emphasize this concept and for convenience when developing the simplex algorithm, let us rewrite the linear system (4.121) in canonical form as follows:

- Given any column vector but the coefficient vector of a tableau of a linear system in canonical form, we write above this vector its associated variable. Clearly, a variable associated to a column vector making up the identity matrix is a basic variable.
- We write at the beginning of each row the name of the basic variable associated to the pivot (of value 1) present in that row.

Therefore, we have an horizontal line of all the variables associated to each column of the tableau (but the coefficient vector) and a vertical line of variables revealing the basic variables in the current basis.

	x_1	x_2	x_3	
x_1	①	0	$\frac{28}{15}$	$\frac{2}{5}$
x_2	0	①	$-\frac{2}{5}$	$\frac{1}{5}$

(4.113)

Suppose that we want to move from the current basic solution (i.e., with x_1 and x_2 in the basis) to a new basic solution with x_3 in the basis. We first identify the column $\mathbf{a}_3$ associated to variable x_3 in the tableau:

	x_1	x_2	x_3	
x_1	(1)	0	$\frac{28}{15}$	$\frac{2}{5}$
x_2	0	(1)	$-\frac{2}{5}$	$\frac{1}{5}$

(4.114)

In order to find the new canonical form with x_3 in the basis, we have to transform through row operations one of the two elements in column $\mathbf{a}_3$ in a pivot. We can see from (4.121) that we have two possibilities: either the element $\frac{28}{15}$ or the element $-\frac{2}{5}$ in column $\mathbf{a}_3$ becomes the new pivot. Moreover, if x_3 enters into the new basis, then one variable between x_1 and x_2 leaves the basis. Once the new pivot is chosen, the variable leaving the basis is the one corresponding to the row where the new pivot is located. This variable is written in the left part of the tableau. For instance, if we decide that $\frac{28}{15}$ is the new pivot, then, as shown in the next tableau, the variable leaving the basis is x_1.

	x_1	x_2	x_3	
x_1	(1)	0	$\frac{28}{15}$	$\frac{2}{5}$
x_2	0	(1)	$-\frac{2}{5}$	$\frac{1}{5}$

(4.115)

However, the new basic solution of the linear system may have, in principle, negative components. As we will see in Section 4.121 this situation must be handled with particular care when using the canonical form to solve linear programming problems for which negative components are not allowed (because of constraints $\mathbf{x} \geq \mathbf{0}$).

This basic solution would be infeasible if we were about to solve a standard form linear programming problem. Therefore, we need to find a methodology able to preserve feasibility when moving from a basic feasible solution to another basic feasible solution.

$$
\begin{array}{cccccc|cccccc|c}
1 & 0 & \dots & 0 & 0 & 0 & y_{1,\,m+1} & \cdots & y_{1,\,q} & \cdots & y_{1n} & y_{10} & \\
0 & 1 & \dots & 0 & 0 & 0 & y_{2,\,m+1} & \cdots & y_{2,\,q} & \cdots & y_{2n} & y_{20} & \\
\vdots & & \ddots & & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \\
\vdots & & & 1 & & \vdots & y_{p,\,m+1} & \cdots & \textcircled{$y_{p,\,q}$} & \cdots & \cdots & y_{p0} & R_p/y_{p,\,q} \\
\vdots & & & & \ddots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \\
0 & 0 & \dots & 0 & 0 & 1 & y_{m,\,m+1} & \cdots & y_{m,\,q} & \cdots & y_{mn} & y_{m0} &
\end{array}
\tag{4.116}
$$

Assuming $y_{p,\,q}$ to be different than 0 and dividing row p by $y_{p,\,q}$, we get

$$\begin{array}{cccccccccccccc}
1 & 0 & \dots & 0 & 0 & 0 & y_{1,\,m+1} & \dots & y_{1,\,q} & \dots & y_{1n} & y_{10} & R_1 - y_{1,\,q}R_p \\
0 & 1 & \dots & 0 & 0 & 0 & y_{2,\,m+1} & \dots & y_{2,\,q} & \dots & y_{2n} & y_{20} & R_2 - y_{2,\,q}R_p \\
\vdots & & \ddots & & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
\vdots & & & \frac{1}{y_{p,\,q}} & & \vdots & \frac{y_{p,\,m+1}}{y_{p,\,q}} & \dots & \textcircled{1} & \dots & \dots & \frac{y_{p0}}{y_{p,\,q}} & \\
\vdots & & & & \ddots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & 0 & 0 & 1 & y_{m,\,m+1} & \dots & y_{m,\,q} & \dots & y_{mn} & y_{m0} & R_m - y_{m,\,q}R_p
\end{array} \tag{4.117}$$

$$\begin{array}{cccccccccccc}
1 & 0 & \dots & 0 & 0 & 0 & y_{1,\,m+1} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{1,\,q} & \dots & 0 & \dots & y_{1n} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{1,\,q} & y_{10} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{1,\,q} \\
0 & 1 & \dots & 0 & 0 & 0 & y_{2,\,m+1} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{2,\,q} & \dots & 0 & \dots & y_{2n} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{2,\,q} & y_{20} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{2,\,q} \\
\vdots & & \ddots & & \vdots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots \\
\vdots & & & \frac{1}{y_{p,\,q}} & & \vdots & \frac{y_{p,\,m+1}}{y_{p,\,q}} & \dots & \textcircled{1} & \dots & \dots & \frac{y_{p0}}{y_{p,\,q}} \\
\vdots & & & & \ddots & \vdots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & \dots & 0 & 0 & 1 & y_{m,\,m+1} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{m,\,q} & \dots & 0 & \dots & y_{mn} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{m,\,q} & y_{m0} - \frac{y_{p,\,m+1}}{y_{p,\,q}} y_{m,\,q}
\end{array} \tag{4.118}$$

$$\begin{aligned}
y'_{ij} &= y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, \quad i \neq p \\
y'_{pj} &= \frac{y_{pj}}{y_{pq}}
\end{aligned} \tag{4.119}$$

4.8 Moving from a basic feasible solution to another basic feasible solution

4.8.1 Another interpretation

4.9 The reduced costs

$$
\begin{aligned}
x_1 &= y_{10} - \sum_{j=m+1}^{n} y_{1j} x_j \\
x_2 &= y_{20} - \sum_{j=m+1}^{n} y_{2j} x_j \\
&\vdots \qquad (4.120) \\
x_k &= y_{k0} - \sum_{j=m+1}^{n} y_{kj} x_j \qquad (4.121) \\
&\vdots \qquad (4.122) \\
x_m &= y_{m0} - \sum_{j=m+1}^{n} y_{mj} x_j. \qquad (4.123)
\end{aligned}
$$

$$
z_0 = \mathbf{c}_B^T \mathbf{x}_B = \sum_{k=1}^{m} c_k x_{kB} = \sum_{k=1}^{m} c_k y_{k0}. \qquad (4.124)
$$

$$
\begin{aligned}
z &= \mathbf{c}^T\mathbf{x} = \\
&= \sum_{k=1}^{n} c_k x_k = \\
&= \sum_{k=1}^{m} c_k x_k + \sum_{k=m+1}^{n} c_k x_k = \\
&= \sum_{k=1}^{m} c_k x_k + \sum_{j=m+1}^{n} c_j x_j = \\
&= \sum_{k=1}^{m} c_k \left(y_{k0} - \sum_{j=m+1}^{n} y_{kj} x_j \right) + \sum_{j=m+1}^{n} c_j x_j = \\
&= \sum_{k=1}^{m} c_k y_{k0} - \sum_{j=m+1}^{n} \sum_{k=1}^{m} c_k y_{kj} x_j + \sum_{j=m+1}^{n} c_j x_j = \\
&= \sum_{k=1}^{m} c_k y_{k0} + \sum_{j=m+1}^{n} \left(c_j - \sum_{k=1}^{m} c_k y_{kj} \right) x_j
\end{aligned}
\tag{4.125}
$$

If we define

$$
z_j = \sum_{k=1}^{m} c_k y_{kj}, \tag{4.126}
$$

the relationship (4.121) becomes

$$
z = \mathbf{c}^T\mathbf{x} = z_0 + \sum_{j=m+1}^{n} (c_j - z_j)\, x_j. \tag{4.127}
$$

4.10 Putting all together: the simplex algorithm

4.11 The revised simplex algorithm

4.12 The simplex algorithm with upper bound constraints

5

Duality

6

Network flow problems and the simplex network algorithm

7

Introduction to computational complexity

8

The branch-and-bound algorithm

Bibliography

[1] C. T. Ragsdale, *Spreadsheet Modeling & Decision Analysis*, 6th ed., South-Western Cengage Learning, 2010.

[2] H. P. Williams, *Logic and Integer Programming*, Springer, 2009.

[3] H. P. Williams, *Model building in Mathematical Programming*, 4th ed., Wiley, 1999 (there is a newer edition of this item).

[4] W. Winston, *Introduction to Mathematical Programming: Applications and Algorithms*, Duxbury Press, 2nd ed., 1995 (there is a newer edition of this item).