1

Introduction

In this short chapter, we shall explain what is meant by linear programming and sketch a history of this subject.

A DIET PROBLEM

Polly wonders how much money she must spend on food in order to get all the energy (2,000 kcal), protein (55 g), and calcium (800 mg) that she needs every day. (For iron and vitamins, she will depend on pills. Nutritionists would disapprove, but the introductory example ought to be simple.) She chooses six foods that seem to be cheap sources of the nutrients; her data are collected in Table 1.1.

Table 1.1 Nutritive Value per Serving

Food	Serving size	Energy (kcal)	Protein (g)	Calcium (mg)	Price per serving (cents)
Oatmeal	28 g	110	4	2	3
Chicken	100 g	205	32	12	24
Eggs	2 large	160	13	54	13
Whole milk	237 cc	160	8	285	9
Cherry pie	170 g	420	4	22	20
Pork with beans	260 g	260	14	80	19

Then she begins to think about her menu. For example, 10 servings of pork with beans would take care of all her needs for only (?) \$1.90 per day. On the other hand, 10 servings of pork with beans is a lot of pork with beans—she would not be able stomach more than 2 servings a day. She decides to impose servings-per-day limits on all six foods:

Oatmeal	at most 4 servings per day
Chicken	at most 3 servings per day
Eggs	at most 2 servings per day
Milk	at most 8 servings per day
Cherry pie	at most 2 servings per day
Pork with beans	at most 2 servings per day.

Then, another look at the data shows Polly that 8 servings of milk and 2 servings of Cherry pie every day will satisfy the requirements nicely and at a cost of only \$1.12. She could cut down a little on the pie or the milk or perhaps try a different combination. But so many combinations seem promising that one could go on and on looking for the best one. Trial and error is not particularly helpful here. To be pragmatic, we may speculate about some as yet unspecified menu consisting of x_1 servings of oatmeal, x_2 servings of chicken, x_3 servings of eggs, and so on. In order to stay below the upper limits, that menu must satisfy

$$
\begin{aligned}
0 &\leq x_1 \leq 4 \\
0 &\leq x_2 \leq 3 \\
0 &\leq x_3 \leq 2 \\
0 &\leq x_4 \leq 8 \\
0 &\leq x_5 \leq 2 \\
0 &\leq x_6 \leq 2.
\end{aligned}
\tag{1.1}
$$

Of course, there are the requirements for energy, protein, and calcium; they lead to the inequalities

$$
\begin{aligned}
110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 &\geq 2{,}000 \\
4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 &\geq 55 \\
2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 &\geq 800.
\end{aligned}
\tag{1.2}
$$

If the numbers $x_1, x_2, \ldots, x_6$ satisfy inequalities (1.1) and (1.2), then they describe a satisfactory menu; such a menu will cost, in cents per day,

$$3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6. \tag{1.3}$$

In Designing the most economical menu, Polly wants to find numbers $x_1, x_2, \ldots, x_6$ which satisfy (1.1) and (1.2), and make (1.3) as small as possible. As a mathematician would put it, she wants to

$$
\begin{array}{ll}
\text{minimize} & 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 \\
\text{subject to} & 0 \le x_1 \le 4 \\
& 0 \le x_2 \le 3 \\
& 0 \le x_3 \le 2 \\
& 0 \le x_4 \le 8 \\
& 0 \le x_5 \le 2 \\
& 0 \le x_6 \le 2 \\
\multicolumn{2}{l}{110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 \ge 2000} \\
\multicolumn{2}{l}{\quad 4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 \ge 55} \\
\multicolumn{2}{l}{\quad 2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 \ge 800.}
\end{array}
\tag{1.4}
$$

Her problem is known as a *diet problem*.

LINEAR PROGRAMMING

Problems of this kind are called "linear programming problems," or "LP problems" for short; linear programming is the branch of applied mathematics concerned with these problems. Here are other examples:

$$
\begin{array}{ll}
\text{maximize} & 5x_1 + 4x_2 + 3x_3 \\
\text{subject to} & 2x_1 + 3x_2 + x_3 \le 5 \\
& 4x_1 + x_2 + 2x_3 \le 11 \\
& 3x_1 + 4x_2 + 2x_3 \le 8 \\
& x_1, x_2, x_3 \ge 0
\end{array}
\tag{1.5}
$$

(with "$x_1, x_2, x_3 \ge 0$" used as shorthand for "$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0$") or

$$
\begin{array}{ll}
\text{minimize} & 3x_1 - x_2 \\
\text{subject to} & -x_1 + 6x_2 - x_3 + x_4 \ge -3 \\
& 7x_2 + 2x_4 = 5 \\
& x_1 + x_2 + x_3 = 1 \\
& x_3 + x_4 \le 2 \\
& x_2, x_3 \ge 0.
\end{array}
\tag{1.6}
$$

In general, if $c_1, c_2, \ldots, c_n$ are real numbers, then the function f of real variables $x_1, x_2, \ldots, x_n$ defined by

$$f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \sum_{j=1}^{n} c_jx_j$$

is called a *linear function*. If f is a linear function and if b is a real number, then the equation

$$f(x_1, x_2, \ldots, x_n) = b$$

is called a *linear equation* and the inequalities

$$f(x_1, x_2, \ldots, x_n) \leq b$$
$$f(x_1, x_2, \ldots, x_n) \geq b$$

are called *linear inequalities*. Linear equations and linear inequalities are both referred to as *linear constraints*. Finally, a *linear programming problem* is the problem of maximizing (or minimizing) a linear function subject to a finite number of linear constraints. We shall usually attach different subscripts i to different constraints and different subscripts j to different variables. For simplicity of exposition, we shall restrict ourselves in Chapters 1–7 to LP problems of the following form:

$$\begin{aligned} &\text{maximize} && \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1, 2, \ldots, m) \\ &&& x_j \geq 0 \qquad (j = 1, 2, \ldots, n). \end{aligned} \tag{1.7}$$

These problems will be referred to as LP problems in the *standard form*. (The reader should be warned that the terminology is far from unified; several authors prefer the terms *canonical* or *symmetric* form, and others reserve these adjectives for altogether different problems.) For example, (1.5) is a problem in the standard form (with $n = 3$, $m = 3$, $a_{11} = 2$, $a_{12} = 3$, and so on). What distinguishes the problems in the standard form from the rest? First, all of their constraints are linear *inequalities*. Secondly, the last n of the $m + n$ constraints in (1.7) are very special: they simply stipulate that none of the n variables may assume negative values. Such constraints are called *nonnegativity constraints*. (Note that problem (1.6) differs from the standard form on both counts: two of its constraints are linear equations and the variables x_1 and x_4 may assume negative values.)

The linear function that is to be maximized or minimized in an LP problem is called the *objective function* of that problem. For example, the function z of variables $x_1, x_2, x_3, x_4, x_5, x_6$ defined by

$$z(x_1, x_2, \ldots, x_6) = 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6$$

is the objective function of Polly's diet problem (1.4). Numbers $x_1, x_2, \ldots, x_n$ that satisfy all the constraints of an LP problem are said to constitute a *feasible solution* to that problem. For instance, we have observed that

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 8, \quad x_5 = 2, \quad x_6 = 0$$

is a feasible solution of (1.4). Finally, a feasible solution that maximizes the objective function (or minimizes it, depending on the form of the problem) is called an *optimal solution*; the corresponding value of the objective function is called the *optimal value* of the problem. As it turns out, the unique optimal solution of (1.4) is

$$x_1 = 4, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 4.5, \quad x_5 = 2, \quad x_6 = 0$$

or simply (4, 0, 0, 4.5, 2, 0). Accordingly, the optimal value of (1.4) is 92.5. Not every LP problem has a unique optimal solution; some problems have many different optimal solutions and others have no optimal solutions at all. The latter may occur for one of two radically different reasons: either there are no feasible solutions at all or there are, in a sense, too many of them. The first case may be illustrated on the problem

$$\begin{array}{lrcl} \text{maximize} & 3x_1 - x_2 & & \\ \text{subject to} & x_1 + x_2 & \leq & 2 \\ & -2x_1 - 2x_2 & \leq & -10 \\ & x_1, x_2 & \geq & 0 \end{array} \tag{1.8}$$

which has no feasible solutions at all. Such problems are called *infeasible*. On the other hand, even though the problem

$$\begin{array}{lrcl} \text{maximize} & x_1 - x_2 & & \\ \text{subject to} & -2x_1 + x_2 & \leq & -1 \\ & -x_1 - 2x_2 & \leq & -2 \\ & x_1, x_2 & \geq & 0 \end{array} \tag{1.9}$$

does have feasible solutions, none of them is optimal: for every number M there is a feasible solution x_1, x_2 such that $x_1 - x_2 > M$. In a sense, (1.9) has such an abundance of feasible solutions that none of them can aspire to be the best. Problems with this property are called *unbounded*. As we shall prove later (Theorem 3.4), every linear programming problem belongs to one of the three categories noted here: it has an optimal solution, is infeasible, or is unbounded.

HISTORY OF LINEAR PROGRAMMING

As mathematical disciplines go, linear programming is quite young. It started in 1947 when G. B. Dantzig designed the "simplex method" for solving linear programming formulations of U.S. Air Force planning problems. What followed was an exciting period of rapid development in this new field. It soon became clear that a surprisingly wide range of apparently unrelated problems in production management could be stated in linear programming terms and, most importantly, solved by the simplex method. Such problems, if noticed at all, had traditionally been tackled by a hit-or-miss approach guided only by experience and intuition. The use of linear programming often brought about a considerable increase in the efficiency of the whole operation. (Until then, expansion of the efficiency frontier usually came from technological innovations. This new way to increase efficiency—*under existing technological conditions*—by improvements in organization and planning, made many managers appreciate the practical importance of mathematics. At least, it made them aware of the advantage of stating their decision problems in clear-cut and well-defined terms.) As the popularity of linear programming theory increased, applications in new areas occurred, many of them far from obvious. In turn, these applications stimulated further theoretical research by pointing out the need for solving problems that would have otherwise seemed uninteresting. In this fascinating interplay between theory and applications, a new branch of applied mathematics established itself.

As calculus developed from the seventeenth century's need to solve problems of mechanics, linear programming developed from the twentieth century's need to solve problems of management. Yet other profound influences stimulated the evolution of the new field from its very inception. Economics was one of them: as early as 1947, T. C. Koopmans began pointing out that linear programming provided an excellent framework for the analysis of classical economic theories, such as the renowned system proposed in 1874 by L. Walras. On the other hand, linear programming brought together previously known theorems of pure mathematics concerning such diverse topics as the geometry of convex sets, extremal problems of combinatorial nature, and the theory of two-person games. Finally, it was fortunate and perhaps even inevitable that linear programming developed concurrently with modern computer technology: without electronic computers, present-day large-scale linear programming would be unthinkable.

Scientific fields are rarely born overnight; with the advantage of hindsight, one can often track down the sources that paved the way for the decisive breakthrough. The field of linear programming is no exception. At the core of its mathematical theory is the study of systems of linear inequalities; such systems were investigated by Fourier as far back as 1826. Since then, quite a few other mathematicians have considered the subject, although none of them has devised an algorithm whose efficiency has come close to that of the simplex method. Nevertheless, some of them proved various special cases of a fundamental theorem that is now called the *duality theorem* of linear programming. On the applied side, L. V. Kantorovich pointed out the practical significance of a restricted class of LP problems, and proposed a rudimentary algorithm for their solution as early as 1939. Regrettably, this effort remained neglected in the U.S.S.R. and unknown elsewhere until long after linear programming became an elegant theory through the independent work of Dantzig and others.

In the 1970s, linear programming came twice to public attention. On October 14, 1975, the Royal Sweden Academy of Sciences awarded the Nobel Prize in economic science to L. V. Kantorovich and T. C. Koopmans "for their contributions to the theory of optimum allocation of resources." (As the reader may know, there is no Nobel Prize in mathematics. Apparently the Academy regarded the work of G. B. Dantzig, who is universally recognized as the father of linear programming, as being too mathematical.) The second event was even more dramatic. Ever since the invention of the simplex method, mathematicians had been looking for a *theoretically* satisfactory algorithm to solve LP problems. (A word of explanation is in order: theoretical criteria for judging the efficiency of algorithms are quite different from practical ones. Thus, an algorithm like the simplex method, which is eminently satisfactory in practical applications, may be found theoretically unsatisfactory. The converse is also true: theoretically satisfactory algorithms may be thoroughly useless in practice. We shall return to this distinction in

Chapter 4.) The breakthrough came in 1979 when L. G. Khachian published a description of such an algorithm (based on earlier works by Shor, and by Judin and Nemirovskii). Newspapers around the world published reports of this result, some of them full of hilarious misinterpretations. We shall present the algorithm in the appendix.

For a thorough survey of the history of linear programming, the reader is referred to Chapter 2 of Dantzig's monograph (1963). References to many applications of linear programming may be found in Riley and Gass (1958). Some of the more recent applications are referenced in Gass (1975).

PROBLEMS

Answers to problems marked with the symbol Δ are found at the back of the book.

1.1 Which of the problems below are in the standard form?

a.
$$\begin{array}{ll} \text{Maximize} & 3x_1 - 5x_2 \\ \text{subject to} & 4x_1 + 5x_2 \geq 3 \\ & 6x_1 - 6x_2 = 7 \\ & x_1 + 8x_2 \leq 20 \\ & x_1, x_2 \geq 0. \end{array}$$

b.
$$\begin{array}{ll} \text{Minimize} & 3x_1 + x_2 + 4x_3 + x_4 + 5x_5 \\ \text{subject to} & 9x_1 + 2x_2 + 6x_3 + 5x_4 + 3x_5 \leq 5 \\ & 8x_1 + 9x_2 + 7x_3 + 9x_4 + 3x_5 \leq 2 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

c.
$$\begin{array}{ll} \text{Maximize} & 8x_1 - 4x_2 \\ \text{subject to} & 3x_1 + x_2 \leq 7 \\ & 9x_1 + 5x_2 \leq -2 \\ & x_1, x_2 \geq 0. \end{array}$$

1.2 State in the standard form:

$$\begin{array}{ll} \text{minimize} & -8x_1 + 9x_2 + 2x_3 - 6x_4 - 5x_5 \\ \text{subject to} & 6x_1 + 6x_2 - 10x_3 + 2x_4 - 8x_5 \geq 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{array}$$

1.3 Prove that (1.8) is infeasible and (1.9) is unbounded.

Δ**1.4** Find necessary and sufficient conditions for the numbers s and t to make the LP problem

$$\begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & sx_1 + tx_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{array}$$

a. have an optimal solution,

b. be infeasible,

c. be unbounded.

.5 Prove or disprove: If problem (1.7) is unbounded, then there is a subscript k such that the problem

$$\begin{aligned} &\text{maximize} && x_k \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij}x_j \le b_i \qquad (i = 1, 2, \ldots, m) \\ &&& x_j \ge 0 \qquad (j = 1, 2, \ldots, n) \end{aligned}$$

is unbounded.

6 [Adapted from Greene et al. (1959).] A meat packing plant produces 480 hams, 400 pork bellies, and 230 picnic hams every day; each of these products can be sold either fresh or smoked. The total number of hams, bellies, and picnics that can be smoked during a normal working day is 420; in addition, up to 250 products can be smoked on overtime at a higher cost. The *net* profits are as follows:

	Fresh	Smoked on regular time	Smoked on overtime
Hams	\$8	\$14	\$11
Bellies	\$4	\$12	\$7
Picnics	\$4	\$13	\$9

For example, the following schedule yields a total net profit of \$9,965:

	Fresh	Smoked	Smoked (overtime)
Hams	165	280	35
Bellies	295	70	35
Picnics	55	70	105

The objective is to find the schedule that maximizes the total net profit. Formulate as an LP problem in the standard form.

7. [Adapted from Charnes et al. (1952).] An oil refinery produces four types of raw gasoline: alkylate, catalytic-cracked, straight-run, and isopentane. Two important characteristics of each gasoline are its performance number PN (indicating antiknock properties) and its vapor pressure RVP (indicating volatility). These two characteristics, together with the production levels in barrels per day, are as follows:

	PN	RVP	Barrels produced
Alkylate	107	5	3,814
Catalytic-cracked	93	8	2,666
Straight-run	87	4	4,016
Isopentane	108	21	1,300

These gasolines can be sold either raw, at \$4.83 per barrel, or blended into aviation gasolines (Avgas A and/or Avgas B). Quality standards impose certain requirements on the aviation gasolines; these requirements, together with the selling prices, are as follows:

	PN	RVP	Price per barrel
Avgas A	at least 100	at most 7	\$6.45
Avgas B	at least 91	at most 7	\$5.91

The PN and RVP of each mixture are simply weighted averages of the PNs and RVPs of its constituents. For example, the refinery could adopt the following strategy:

- Blend 2,666 barrels of alkylate and 2,666 barrels of catalytic into 5,332 barrels of Avgas A with

$$PN = \frac{(2{,}666 \times 107) + (2{,}666 \times 93)}{5{,}332} = 100$$

$$RVP = \frac{(2{,}666 \times 5) + (2{,}666 \times 8)}{5{,}332} = 6.5.$$

- Blend 1,148 barrels of alkylate, 4,016 barrels of straight-run, and 1,024 barrels of isopentane into 6,188 barrels of Avgas B with

$$PN = \frac{(1{,}148 \times 107) + (4{,}016 \times 87) + (1{,}024 \times 108)}{6{,}188} \doteq 94.2$$

$$RVP = \frac{(1{,}148 \times 5) + (4{,}016 \times 4) + (1{,}024 \times 21)}{6{,}188} \doteq 7.$$

Sell 276 barrels of isopentane raw.

This sample plan yields a total profit of

$$(5{,}332 \times 6.45) + (6{,}188 \times 5.91) + (276 \times 4.83) \doteq \$72{,}296.$$

The refinery aims for the plan that yields the largest possible profit. Formulate as an LP problem in the standard form.

1.8 An electronics company has a contract to deliver 20,000 radios within the next four weeks. The client is willing to pay \$20 for each radio delivered by the end of the first week, \$18 for those delivered by the end of the second week, \$16 by the end of the third week, and \$14 by the end of the fourth week. Since each worker can assemble only 50 radios per week, the company cannot meet the order with its present labor force of 40; hence it must hire and train temporary help. Any of the experienced workers can be taken off the assembly line to instruct a class of three trainees; after one week of instruction, each of the trainees can either proceed to the assembly line or instruct additional new classes.

At present, the company has no other contracts; hence some workers may become idle once the delivery is completed. All of them, whether permanent or temporary, must be kept on the payroll till the end of the fourth week. The weekly wages of a worker, whether assembling, instructing, or being idle, are \$200; the weekly wages of a trainee are \$100. The production costs, excluding the worker's wages, are \$5 per radio.

For example, the company could adopt the following program.

First week: 10 assemblers, 30 instructors, 90 trainees
Workers' wages: \$8,000

	Trainees' wages: \$9,000 Profit from 500 radios: \$7,500 Net loss: \$9,500
Second week:	120 assemblers, 10 instructors, 30 trainees Workers' wages: \$26,000 Trainees' wages: \$3,000 Profit from 6,000 radios: \$78,000 Net profit: \$49,000
Third week:	160 assemblers Workers' wages: \$32,000 Profit from 8,000 radios: \$88,000 Net profit: \$56,000
Fourth week:	110 assemblers, 50 idle Workers' wages: \$32,000 Profit from 5,500 radios: \$49,500 Net profit: \$17,500

This program, leading to a total net profit of \$113,000, is one of many possible programs. The company's aim is to maximize the total net profit. Formulate as an LP problem (not necessarily in the standard form).

1.9 [S. Masuda (1970); see also V. Chvátal (1983).] The *bicycle problem* involves n people who have to travel a distance of ten miles, and have one single-seat bicycle at their disposal. The data are specified by the walking speed w_j and the bicycling speed b_j of each person j $(j = 1, 2, \ldots, n)$; the task is to minimize the arrival time of the last person. (Can you solve the case of $n = 3$ and $w_1 = 4$, $w_2 = w_3 = 2$, $b_1 = 16$, $b_2 = b_3 = 12$?) Show that the optimal value of the LP problem

$$
\begin{array}{lrll}
\text{minimize} & t & & \\
\text{subject to} & t - x_j - x'_j - y_j - y'_j \geq & 0 & (j = 1, 2, \ldots, n) \\
& t \quad - \sum_{j=1}^{n} y_j - \sum_{j=1}^{n} y'_j \geq & 0 & \\
& w_j x_j - w_j x'_j + b_j y_j - b_j y'_j = & 10 & (j = 1, 2, \ldots, n) \\
& \sum_{j=1}^{n} b_j y_j - \sum_{j=1}^{n} b_j y'_j \leq & 10 & \\
& x_j, x'_j, y_j, y'_j \geq & 0 & (j = 1, 2, \ldots, n)
\end{array}
$$

provides a lower bound on the optimal value of the bicycle problem.

2

How the Simplex Method Works

In this chapter, we shall learn to solve LP problems in the standard form by the simplex method. A rigorous analysis of the details will be deferred to Chapter 3.

FIRST EXAMPLE

We shall illustrate the simplex method on the following example:

$$\begin{array}{ll} \text{maximize} & 5x_1 + 4x_2 + 3x_3 \\ \text{subject to} & 2x_1 + 3x_2 + x_3 \le 5 \\ & 4x_1 + x_2 + 2x_3 \le 11 \\ & 3x_1 + 4x_2 + 2x_3 \le 8 \\ & \qquad\quad x_1, x_2, x_3 \ge 0. \end{array} \tag{2.1}$$

A preliminary step of the method consists of introducing so-called slack variables.

In order to motivate this concept, let us consider the first of our constraints,

$$2x_1 + 3x_2 + x_3 \le 5. \tag{2.2}$$

For every feasible solution x_1, x_2, x_3, the value of the left-hand side of (2.1) is at most the value of the right-hand side; often, there may be a slack between the two values. We shall denote the slack by x_4. That is, we shall *define* $x_4 = 5 - 2x_1 - 3x_2 - x_3$; with this notation, inequality (2.2) may now be written as $x_4 \ge 0$. In an analogous way, the next two constraints give rise to variables x_5 and x_6. Finally, allowing a time-honored convention, we shall denote the objective function $5x_1 + 4x_2 + 3x_3$ by z. To summarize: for every *choice* of numbers x_1, x_2, and x_3, we shall *define* numbers x_4, x_5, x_6, and z by the formulas

$$\begin{aligned}
x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\
x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\
x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
z &= \phantom{11 -{}} 5x_1 + 4x_2 + 3x_3.
\end{aligned} \tag{2.3}$$

With this notation, our problem may be restated as

$$\text{maximize} \quad z \quad \text{subject to} \quad x_1, x_2, x_3, x_4, x_5, x_6 \ge 0. \tag{2.4}$$

The new variables x_4, x_5, x_6 defined by (2.3) are called *slack variables*; the old variables x_1, x_2, x_3 are usually referred to as the *decision variables*. It is crucial to note that the equations in (2.3) spell out an equivalence between (2.1) and (2.4). More precisely:

- Every feasible solution x_1, x_2, x_3 of (2.1) can be extended, in the unique way determined by (2.3), into a feasible solution $x_1, x_2, \ldots, x_6$ of (2.4).
- Every feasible solution $x_1, x_2, \ldots, x_6$ of (2.4) can be restricted, simply by deleting the slack variables, into a feasible solution x_1, x_2, x_3 of (2.1).
- This correspondence between feasible solutions of (2.1) and feasible solutions of (2.4) carries optimal solutions of (2.1) onto optimal solutions of (2.4), and vice versa.

The grand strategy of the simplex method is that of *successive improvements*: having found some feasible solution $x_1, x_2, \ldots, x_6$ of (2.4), we shall try to proceed to another feasible solution $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_6$, which is better in the sense that

$$5\bar{x}_1 + 4\bar{x}_2 + 3\bar{x}_3 > 5x_1 + 4x_2 + 3x_3.$$

Repeating this process a finite number of times, we shall eventually arrive at an optimal solution.

To begin with, we need some feasible solution $x_1, x_2, \ldots, x_6$. Finding one in our example presents no difficulty: setting the decision variables x_1, x_2, x_3 at zero, we evaluate the slack variables x_4, x_5, x_6 from (2.3). Hence our initial solution,

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 5, \quad x_5 = 11, \quad x_6 = 8 \tag{2.5}$$

yields $z = 0$.

In the spirit of the grand strategy sketched above, we should now look for a feasible solution that yields a higher value of z. Finding such a solution is not difficult. For example, if we keep $x_2 = x_3 = 0$ and increase the value of x_1, we obtain $z = 5x_1 > 0$. Thus, if we keep $x_2 = x_3 = 0$ and set $x_1 = 1$, we obtain $z = 5$ (and $x_4 = 3, x_5 = 7, x_6 = 5$). Better yet, if we keep $x_2 = x_3 = 0$ and set $x_1 = 2$, we obtain $z = 10$ (and $x_4 = 1$, $x_5 = 3$, $x_6 = 2$). However, if we keep $x_2 = x_3 = 0$ and set $x_1 = 3$, we obtain $z = 15$ and $x_4 = x_5 = x_6 = -1$; this won't do, since feasibility requires $x_i \geq 0$ for every i. The moral is that we cannot increase x_1 too much. The question is: *Just how much can we increase x_1 (keeping $x_2 = x_3 = 0$ at the same time) and still maintain feasibility* ($x_4, x_5, x_6 \geq 0$)?

The condition $x_4 = 5 - 2x_1 - 3x_2 - x_3 \geq 0$ implies $x_1 \leq \frac{5}{2}$; similarly, $x_5 \geq 0$ implies $x_1 \leq \frac{11}{4}$ and $x_6 \geq 0$ implies $x_1 \leq \frac{8}{3}$. Of these three bounds, the first is the most stringent. Increasing x_1 up to that bound we obtain our next solution,

$$x_1 = \frac{5}{2}, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 0, \quad x_5 = 1, \quad x_6 = \frac{1}{2}. \tag{2.6}$$

Note that this solution yields $z = \frac{25}{2}$, which is indeed an improvement over $z = 0$.

Next, we should look for a feasible solution that is even better than (2.6). However, this task seems a little more difficult. What made the first iteration so easy? We had at our disposal not only the feasible solution (2.5), but also the system of linear equations (2.3), which guided us in our quest for an improved feasible solution. If we wish to continue in a similar way, we should manufacture a new system of linear equations that relates to (2.6) much as system (2.3) relates to (2.5).

What properties should the new system have? Note that (2.3) expresses the variables that assume positive values in (2.5) in terms of the variables that assume zero values in (2.5). Similarly, the new system should express those variables that assume positive values in (2.6) in terms of the variables that assume zero values in (2.6): in short, it should express x_1, x_5, x_6 (as well as z) in terms of x_2, x_3, and x_4. In particular, the variable x_1, which just changed its value from zero to positive should change its position from the right-hand side to the left-hand side of the system of equations. Similarly, the variable x_4, which just changed its value from positive to zero, should move from the left-hand side to the right-hand side.

To construct the new system, we shall begin with the newcomer to the left-hand side, namely, the variable x_1. The desired formula for x_1 in terms of x_2, x_3, x_4 is obtained easily from the first equation in (2.3):

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4. \tag{2.7}$$

Next, in order to express x_5, x_6, and z in terms of x_2, x_3, x_4, we simply substitute from (2.7) into the corresponding rows of (2.3):

$$\begin{aligned}
x_5 &= 11 - 4\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - x_2 - 2x_3 \\
&= 1 + 5x_2 + 2x_4, \\
x_6 &= 8 - 3\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - 4x_2 - 2x_3 \\
&= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4, \\
z &= 5\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) + 4x_2 + 3x_3 \\
&= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4.
\end{aligned}$$

Hence our new system reads

$$\begin{aligned}
x_1 &= \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
x_6 &= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\
z &= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4.
\end{aligned} \tag{2.8}$$

As we did in the first iteration, we shall now try to increase the value of z by increasing the value of a suitably chosen right-hand side variable, while at the same time keeping the remaining right-hand side variables fixed at zero. Note that increases in the values of x_2 or x_4 would bring about *decreases* in the value of z, which is very much against our intentions. Thus, we have no choice: the right-hand side variable to increase its value is necessarily x_3. How much can we increase x_3? The answer can be read directly from system (2.8): with $x_2 = x_4 = 0$, the constraint $x_1 \geq 0$ implies $x_3 \leq 5$, the constraint $x_5 \geq 0$ imposes no restriction at all, and the constraint $x_6 \geq 0$ implies $x_3 \leq 1$. Hence, $x_3 = 1$ is the best we can do; our new solution is

$$x_1 = 2, \quad x_2 = 0, \quad x_3 = 1, \quad x_4 = 0, \quad x_5 = 1, \quad x_6 = 0. \tag{2.9}$$

Note that the value of z just increased from 12.5 to 13.)

As we have learned, getting just the improved solution isn't good enough; we also want a system of linear equations to go with (2.9). In this system, the positive-valued variables x_1, x_3, x_5 will appear on the left, whereas the zero-valued variables x_2, x_4, x_6

will appear on the right. To construct the system, we begin again with the newcomer to the left-hand side, namely, the variable x_3. From the third equation in (2.8), we have $x_3 = 1 + x_2 + 3x_4 - 2x_6$; substituting for x_3 into the remaining equations in (2.8), we obtain

$$\begin{aligned} x_3 &= 1 + x_2 + 3x_4 - 2x_6 \\ x_1 &= 2 - 2x_2 - 2x_4 + x_6 \\ x_5 &= 1 + 5x_2 + 2x_4 \\ z &= 13 - 3x_2 - x_4 - x_6. \end{aligned} \tag{2.10}$$

Now it's time for the third iteration. First of all, from the right-hand side of (2.10) we have to choose a variable whose increase brings about an increase of the objective function. However, there is no such variable: indeed, if we increase any of the right-hand side variables x_2, x_4, x_6, we will make the value of z *decrease*. Thus, it seems that we have come to a standstill. In fact, the very presence of this standstill indicates that we are done; we have solved our problem; the solution described by the last table is optimal. Why? The answer lies hidden in the last row of (2.10):

$$z = 13 - 3x_2 - x_4 - x_6. \tag{2.11}$$

Our last solution (2.9) yields $z = 13$; proving that this solution is optimal amounts to proving that every feasible solution satisfies the inequality $z \leq 13$. Since every feasible solution $x_1, x_2, \ldots, x_6$ satisfies, among other relations, the inequalities $x_2 \geq 0$, $x_4 \geq 0$, and $x_6 \geq 0$, the desired inequality $z \leq 13$ follows directly from (2.11).

DICTIONARIES

In general, given a problem

$$\begin{aligned} &\text{maximize} && \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1, 2, \ldots, m) \\ & && x_j \geq 0 \qquad (j = 1, 2, \ldots, n) \end{aligned} \tag{2.12}$$

we first introduce the slack variables $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ and denote the objective function by z. That is, we define

$$\begin{aligned} x_{n+i} &= b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad (i = 1, 2, \ldots, m) \\ z &= \sum_{j=1}^{n} c_j x_j. \end{aligned} \tag{2.13}$$

In the framework of the simplex method, each feasible solution $x_1, x_2, \ldots, x_n$ of (2.12) is represented by $n + m$ *nonnegative* numbers $x_1, x_2, \ldots, x_{n+m}$, with $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ defined by (2.13). In each iteration, the simplex method moves from some feasible solution $x_1, x_2, \ldots, x_{n+m}$ to another feasible solution $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{n+m}$, which is better than the previous one in the sense that

$$\sum_{j=1}^{n} c_j\bar{x}_j > \sum_{j=1}^{n} c_jx_j.$$

Actually, the last statement is not quite correct: the inequality is not always strict. This point and other subtleties will be discussed in Chapter 3.)

As we have seen, it is convenient to associate a system of linear equations with each of the feasible solutions: such systems make it easier to find the improved feasible solutions. They do so by translating any choice of values of the right-hand side variables into the corresponding values of the left-hand side variables and of the objective function. Following J. E. Strum (1972), we shall refer to these systems as *dictionaries*. Thus, every dictionary associated with (2.12) will be a system of linear equations in the variables $x_1, x_2, \ldots, x_{n+m}$ and z. However, not every system of linear equations in these variables constitutes a dictionary. To begin with, we have defined $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ and z in terms of $x_1, x_2, \ldots, x_n$, and so the $n + m + 1$ variables are heavily interdependent. This interdependence must be captured by every dictionary associated with (2.12): the translations must be correct. More precisely, we shall insist that:

Every solution of the set of equations comprising a dictionary must be also a solution of (2.13), and vice versa. (2.14)

For example, for every choice of numbers $x_1, x_2, \ldots, x_6$ and z, the following three statements are equivalent:

- $x_1, x_2, \ldots, x_6, z$ constitute a solution of (2.3),
- $x_1, x_2, \ldots, x_6, z$ constitute a solution of (2.8),
- $x_1, x_2, \ldots, x_6, z$ constitute a solution of (2.10).

In that sense, the three dictionaries (2.3), (2.8), and (2.10) contain the same information concerning the interdependence among the seven variables. Nevertheless, each of the three dictionaries presents this information in its very own way. The form of (2.13) suggests that we are free to choose the numerical values of x_1, x_2, and x_3 at will, whereupon the values of x_4, x_5, x_6, and z are determined: in this dictionary, the decision variables x_1, x_2, x_3 act as independent variables, while z and the slack variables x_4, x_5, x_6 are dependent on them. Dictionary (2.8) presents x_2, x_3, x_4 as independent and x_1, x_5, x_6, z as dependent. In dictionary (2.10), the independent variables are x_2, x_4, x_6 and the dependent ones are x_3, x_1, x_5, z. In general:

The equations of every dictionary must express m of the variables x_1,

$x_2, \ldots, x_{n+m}$ and the objective function z in terms of the remaining n variables. (2.15)

The properties (2.14) and (2.15) are the defining properties of dictionaries.

In addition to these two properties, dictionaries (2.3), (2.8), and (2.10) have the following property:

Setting the right-hand side variables at zero and evaluating the left-hand side variables, we arrive at a *feasible* solution.

Dictionaries with this additional property will be called *feasible dictionaries*. Hence, every feasible dictionary describes a feasible solution. However, not every feasible solution is described by a feasible dictionary; for instance, no dictionary describes the feasible solution $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 2, x_5 = 5, x_6 = 3$ of (2.1). Feasible solutions that can be described by dictionaries are called *basic*. The characteristic feature of the simplex method is the fact that it works exclusively with basic feasible solutions and ignores all other feasible solutions.

SECOND EXAMPLE

We shall complete our preview of the simplex method by applying it to another LP problem:

$$\begin{array}{ll} \text{maximize} & 5x_1 + 5x_2 + 3x_3 \\ \text{subject to} & x_1 + 3x_2 + x_3 \leq 3 \\ & -x_1 + 3x_3 \leq 2 \\ & 2x_1 - x_2 + 2x_3 \leq 4 \\ & 2x_1 + 3x_2 - x_3 \leq 2 \\ & x_1, x_2, x_3 \geq 0. \end{array}$$

In this case, the initial feasible dictionary reads

$$\begin{array}{l} x_4 = 3 - x_1 - 3x_2 - x_3 \\ x_5 = 2 + x_1 - 3x_3 \\ x_6 = 4 - 2x_1 + x_2 - 2x_3 \\ x_7 = 2 - 2x_1 - 3x_2 + x_3 \\ \hline z = 5x_1 + 5x_2 + 3x_3. \end{array} \tag{2.16}$$

(Even though the order of the equations in a dictionary is quite irrelevant, we shall make a habit of writing the formula for z last and separating it from the rest of the table by a solid line. Of course, that does *not* mean that the last equation is the sum of the previous ones.) This feasible dictionary describes the feasible solution

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 3, \quad x_5 = 2, \quad x_6 = 4, \quad x_7 = 2.$$

However, there is no need to write this solution down, as we just did: the solution is implicit in the dictionary.

In the first iteration, we shall attempt to increase the value of z by making one of the right-hand side variables positive. At this moment, any of the three variables x_1, x_2, x_3 would do. In small examples, it is common practice to choose the variable that, in the formula for z, has the largest coefficient: the increase in that variable will make z increase at the fastest rate (but not necessarily to the highest level). In our case, this rule leaves us a choice between x_1 and x_2; choosing arbitrarily, we decide to make x_1 positive. As the value of x_1 increases, so does the value of x_5. However, the values of x_4, x_6, and x_7 decrease, and none of them is allowed to become negative. Of the three constraints $x_4 \geq 0$, $x_6 \geq 0$, $x_7 \geq 0$ that impose upper bounds on the increment of x_1, the last constraint $x_7 \geq 0$ is the most stringent: it implies $x_1 \leq 1$. In the improved feasible solution, we shall have $x_1 = 1$ and $x_7 = 0$. Without writing the new solution down, we shall now construct the new dictionary. All we need to know is that x_1 just made its way from the right-hand side to the left, whereas x_7 went in the opposite direction. From the fourth equation in (2.16), we have

$$x_1 = 1 - \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_7. \tag{2.17}$$

Substituting from (2.17) into the remaining equations of (2.16), we arrive at the desired dictionary

$$\begin{array}{rcl} x_1 &=& 1 - \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_7 \\ x_4 &=& 2 - \frac{3}{2}x_2 - \frac{3}{2}x_3 + \frac{1}{2}x_7 \\ x_5 &=& 3 - \frac{3}{2}x_2 - \frac{5}{2}x_3 - \frac{1}{2}x_7 \\ x_6 &=& 2 + 4x_2 - 3x_3 + x_7 \\ \hline z &=& 5 - \frac{5}{2}x_2 + \frac{11}{2}x_3 - \frac{5}{2}x_7. \end{array} \tag{2.18}$$

The construction of (2.18) completes the first iteration of the simplex method.

Digression on Terminology

The variables x_j that appear on the left-hand side of a dictionary are called *basic*; the variables x_j that appear on the right-hand side are *nonbasic*. The basic variables are said to constitute a *basis*. Of course, the basis changes with each iteration: for example, in the first iteration, x_1 entered the basis whereas x_7 left it. In each iteration, we first choose the nonbasic variable that is to enter the basis and then we find out which basic variable must leave the basis. The choice of the *entering* variable is motivated by our desire to increase the value of z; the determination of the *leaving* variable is based on the requirement that all variables must assume nonnegative values. The leaving variable is that basic variable whose nonnegativity imposes the most stringent upper bound on the increment of the entering variable. The formula for the leaving variable appears in the *pivot row* of the dictionary; the computational process of constructing the new dictionary is referred to as *pivoting*.

Back to the Second Example

In our example, the variable to enter the basis during the second iteration is quite unequivocally x_3. This is the only nonbasic variable in (2.18) whose coefficient in the last row is positive. Of the four basic variables, x_6 imposes the most stringent upper bound on the increase of x_3, and, therefore, has to leave the basis. Pivoting, we arrive at our third dictionary,

$$\begin{array}{rcl}
x_3 &=& \frac{2}{3} + \frac{4}{3}x_2 + \frac{1}{3}x_7 - \frac{1}{3}x_6 \\[6pt]
x_1 &=& \frac{4}{3} - \frac{5}{6}x_2 - \frac{1}{3}x_7 - \frac{1}{6}x_6 \\[6pt]
x_4 &=& 1 - \frac{7}{2}x_2 \qquad\quad + \frac{1}{2}x_6 \\[6pt]
x_5 &=& \frac{4}{3} - \frac{29}{6}x_2 - \frac{4}{3}x_7 + \frac{5}{6}x_6 \\[6pt]
\hline
z &=& \frac{26}{3} + \frac{29}{6}x_2 - \frac{2}{3}x_7 - \frac{11}{6}x_6.
\end{array} \tag{2.19}$$

In the third iteration, the entering variable is x_2 and the leaving variable is x_5. Pivoting yields the dictionary

$$\begin{array}{rcl}
x_2 &=& \frac{8}{29} - \frac{8}{29}x_7 + \frac{5}{29}x_6 - \frac{6}{29}x_5 \\[6pt]
x_3 &=& \frac{30}{29} - \frac{1}{29}x_7 - \frac{3}{29}x_6 - \frac{8}{29}x_5 \\[6pt]
x_1 &=& \frac{32}{29} - \frac{3}{29}x_7 - \frac{9}{29}x_6 + \frac{5}{29}x_5 \\[6pt]
x_4 &=& \frac{1}{29} + \frac{28}{29}x_7 - \frac{3}{29}x_6 + \frac{21}{29}x_5 \\[6pt]
\hline
z &=& 10 - 2x_7 - x_6 - x_5.
\end{array} \tag{2.20}$$

At this point, no nonbasic variable can enter the basis without making the value of z decrease. Hence, the last dictionary describes an optimal solution of our example. That solution is

$$x_1 = \frac{32}{29}, \quad x_2 = \frac{8}{29}, \quad x_3 = \frac{30}{29}$$

and it yields $z = 10$.

URTHER REMARKS

The reader may have noticed that, having first carefully laid down the definition of a dictionary, we then proceeded to refer to (2.18), (2.19), and (2.20) as dictionaries, without bothering to verify that they do indeed have property (2.14). Such carelessness can be easily justified. Take, for example, system (2.18). Since (2.18) arises from (2.16) by arithmetical operations (namely, pivoting with x_1 entering and x_7 leaving), every solution of (2.16) must be also a solution of (2.18). The converse is also true, since (2.16) can be obtained from (2.18) by pivoting with x_7 entering and x_1 leaving. Hence, *every solution of* (2.18) *is a solution of* (2.16), *and vice versa*. Similar arguments show that *every solution of* (2.19) *is a solution of* (2.18), *and vice versa*; and that *every solution of* (2.20) *is a solution of* (2.19), *and vice versa*.

Another point of concern is the question of the *uniqueness*, as opposed to the *existence*, of optimal solutions. This question will be of no great interest to us; nevertheless, it is easy to deal with and so we will get it out of the way now. Note that in each of our two examples, we not only find an optimal solution, but we also collected the evidence to prove that there is only one optimal solution. For instance, the final dictionary for our first problem reads

$$\begin{array}{rcl}
x_3 &=& 1 + x_2 + 3x_4 - 2x_6 \\
x_1 &=& 2 - 2x_2 - 2x_4 + x_6 \\
x_5 &=& 1 + 5x_2 + 2x_4 \\
\hline
z &=& 13 - 3x_2 - x_4 - x_6.
\end{array}$$

The last row shows that every feasible solution with $z = 13$ satisfies $x_2 = x_4 = x_6 = 0$; the rest of the dictionary shows that every such solution satisfies $x_3 = 1, x_1 = 2, x_5 = 1$; therefore, there is just one optimal solution. A similar argument applies to the second problem.

Of course, there are LP problems with more than just one optimal solution; having solved such problems by the simplex method, we can effectively describe all the optimal solutions. For example, consider the following dictionary:

$$\begin{aligned}
x_4 &= 3 + x_2 - 2x_5 + 7x_3 \\
x_1 &= 1 - 5x_2 + 6x_5 - 8x_3 \\
x_6 &= 4 + 9x_2 + 2x_5 - x_3 \\
\hline
z &= 8 \qquad\qquad\qquad - x_3.
\end{aligned}$$

The last row shows that every optimal solution satisfies $x_3 = 0$ (but not necessarily $x_2 = 0$ or $x_5 = 0$). For such solutions, the rest of the dictionary implies

$$\begin{aligned}
x_4 &= 3 + x_2 - 2x_5 \\
x_1 &= 1 - 5x_2 + 6x_5 \\
x_6 &= 4 + 9x_2 + 2x_5.
\end{aligned} \tag{2.21}$$

We conclude that every optimal solution arises by the substitution formulas (2.21) from some x_2 and x_5 such that

$$\begin{aligned}
-x_2 + 2x_5 &\le 3 \\
5x_2 - 6x_5 &\le 1 \\
-9x_2 - 2x_5 &\le 4 \\
x_2, x_5 &\ge 0.
\end{aligned}$$

(In fact, the inequality $-9x_2 - 2x_5 \le 4$ is clearly redundant; its validity is forced by $x_2 \ge 0$ and $x_5 \ge 0$.)

There are a few other rough spots we deliberately failed to point out in our overview of the simplex method. We shall discuss them in Chapter 3.

TABLEAU FORMAT

The simplex method is often introduced in a format differing from ours. To outline the more popular *tableau format*, we shall return to the first example of this chapter. To begin, let us write down the equations of the first dictionary in a slightly modified form:

$$\begin{array}{rcl}
2x_1 + 3x_2 + x_3 + x_4 & & = 5 \\
4x_1 + x_2 + 2x_3 & + x_5 & = 11 \\
3x_1 + 4x_2 + 2x_3 & + x_6 & = 8 \\
\hline
-z + 5x_1 + 4x_2 + 3x_3 & & = 0.
\end{array}$$

Recording just the coefficients at the x_i's, together with the right-hand sides, we obtain our first *tableau*:

$$\begin{array}{cccccc c}
2 & 3 & 1 & 1 & 0 & 0 & 5 \\
4 & 1 & 2 & 0 & 1 & 0 & 11 \\
3 & 4 & 2 & 0 & 0 & 1 & 8 \\
\hline
5 & 4 & 3 & 0 & 0 & 0 & 0.
\end{array}$$

n a similar way, the equations of the second dictionary,

$$
\begin{array}{rcl}
x_1 + \frac{3}{2}x_2 + \frac{1}{2}x_3 + \frac{1}{2}x_4 & = & \frac{5}{2} \\
- 5x_2 \qquad - 2x_4 + x_5 & = & 1 \\
- \frac{1}{2}x_2 + \frac{1}{2}x_3 - \frac{3}{2}x_4 \qquad + x_6 & = & \frac{1}{2} \\
\hline
-z \quad - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 & = & -\frac{25}{2}
\end{array}
$$

ive rise to a second tableau:

$$
\begin{array}{cccccccc}
1 & \frac{3}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & & \frac{5}{2} \\
0 & -5 & 0 & -2 & 1 & 0 & & 1 \\
0 & -\frac{1}{2} & \frac{1}{2} & -\frac{3}{2} & 0 & 1 & & \frac{1}{2} \\
\hline
0 & -\frac{7}{2} & \frac{1}{2} & -\frac{5}{2} & 0 & 0 & & -\frac{25}{2}.
\end{array}
$$

is a routine matter to translate the pivoting rules, previously derived in terms of dictionaries, to the language of tableaus. The following steps describe the procedure; the reader should ıve no trouble verifying its correctness. (At any rate, the procedure is not important for our position since we do not use the tableau format.)

Step 1. Examine all numbers in the last row (except the one farthest right, which equals the current value of $-z$). If all of them are negative or zero, stop: the tableau describes an optimal solution. Otherwise find the largest of these numbers; the column in which it appears is called the *pivot column* and corresponds to the entering variable.

For example, the pivot column in our first tableau is the first one:

$$
\begin{array}{|c|cccccc}
2 & 3 & 1 & 1 & 0 & 0 & 5 \\
4 & 1 & 2 & 0 & 1 & 0 & 11 \\
3 & 4 & 2 & 0 & 0 & 1 & 8 \\
\hline
5 & 4 & 3 & 0 & 0 & 0 & 0
\end{array}
$$

Step 2. For each row whose entry r in the pivot column is positive, look up the entry s in the rightmost column. The row with the smallest ratio $\frac{s}{r}$ is called the *pivot row* and corresponds to the leaving variable. (If all the entries in the pivot column are negative or zero, then the problem is unbounded; more on that in Chapter 3.)

In our example, the pivot row is the first row $\left(\text{with } \frac{s}{r} = \frac{5}{2}\right)$:

2	3	1	1	0	0	5
4	1	2	0	1	0	11
3	4	2	0	0	1	8
5	4	3	0	0	0	0.

Step 3. Divide every entry in the pivot row by the *pivot number*, found in the intersection of the pivot row with the pivot column:

1	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{5}{2}$
4	1	2	0	1	0	11
3	4	2	0	0	1	8
5	4	3	0	0	0	0.

Step 4. From every remaining row, subtract a suitable multiple of the new pivot row. This operation is designed to make every entry in the pivot column (except for the pivot number) become zero; hence, the "suitable multiple" results when the new pivot row is multiplied by the entry appearing in the pivot column and in the row in question. (In our example, step 4 results in the second tableau.)

A tableau is nothing but a cryptic recording of a dictionary with all the variables collected on the left-hand side and the symbols for these variables omitted. We shall continue to use dictionaries instead, since they are more explicit. (Of course, nothing prevents the reader tired of writing the same symbols $x_1, x_2, \ldots$ over and over again from using the tableau shorthand.) □

A WARNING

There is often more than one way of describing a particular algorithm; descriptions aimed at clarifying underlying concepts are often quite different from those that suggest efficient computer implementations. The simplex method is no exception. Dictionaries may provide a convenient tool for explaining its basic principles. However, in implementing the method for computer solutions of large problems, considerations of computational efficiency and numerical accuracy overshadow such didactic niceties. We shall begin to study efficient implementations of the simplex method in Chapters 7 and 8.

PROBLEMS

2.1 Solve the following problems by the simplex method:

a.
$$\begin{array}{ll} \text{maximize} & 3x_1 + 2x_2 + 4x_3 \\ \text{subject to} & x_1 + x_2 + 2x_3 \leq 4 \\ & 2x_1 + 3x_3 \leq 5 \\ & 2x_1 + x_2 + 3x_3 \leq 7 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

b.
$$\begin{array}{ll} \text{maximize} & 5x_1 + 6x_2 + 9x_3 + 8x_4 \\ \text{subject to} & x_1 + 2x_2 + 3x_3 + x_4 \leq 5 \\ & x_1 + x_2 + 2x_3 + 3x_4 \leq 3 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{array}$$

c.
$$\begin{array}{ll} \text{maximize} & 2x_1 + x_2 \\ \text{subject to} & 2x_1 + 3x_2 \leq 3 \\ & x_1 + 5x_2 \leq 1 \\ & 2x_1 + x_2 \leq 4 \\ & 4x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0. \end{array}$$

2.2 Use the simplex method to describe *all* the optimal solutions of the following problem:

$$\begin{array}{ll} \text{maximize} & 2x_1 + 3x_2 + 5x_3 + 4x_4 \\ \text{subject to} & x_1 + 2x_2 + 3x_3 + x_4 \leq 5 \\ & x_1 + x_2 + 2x_3 + 3x_4 \leq 3 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{array}$$

3

Pitfalls and How to Avoid Them

The examples illustrating the simplex method in the preceding chapter were purposely smooth. They did not point out the dangers that can occur. The purpose of the present chapter, therefore, is to rigorously analyze the method by scrutinizing its every step.

THREE KINDS OF PITFALLS

Three kinds of pitfalls can occur in the simplex method.

(i) INITIALIZATION. We might not be able to start: How do we get hold of a feasible dictionary?

(ii) ITERATION. We might get stuck in some iteration: Can we always choose an entering variable, find the leaving variable, and construct the next feasible dictionary by pivoting?

(iii) TERMINATION. We might not be able to finish: Can the simplex method construct an endless sequence of dictionaries without ever reaching an optimal solution?

In the preceding chapter, INITIALIZATION never came up. Given a problem

$$\begin{aligned}
&\text{maximize} && \sum_{j=1}^{n} c_j x_j \\
&\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad (i = 1, 2, \dots, m) \\
& && x_j \ge 0 \qquad (j = 1, 2, \dots, n)
\end{aligned} \tag{3.1}$$

we constructed the initial feasible dictionary by simply writing down the formulas defining the slack variables and the objective function,

$$\begin{array}{l}
x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad (i = 1, 2, \dots, m) \\
\hline
z \quad = \quad \sum_{j=1}^{n} c_j x_j \, .
\end{array}$$

In general, this dictionary is feasible if and only if each right-hand side, b_i, in (3.1) is nonnegative. This is the case if and only if

$$x_1 = 0, \quad x_2 = 0, \dots, x_n = 0$$

is a feasible solution of (3.1). Since the set of zero values is sometimes called the "origin," problems (3.1) with each right-hand side b_i nonnegative are referred to as problems with a *feasible origin*. For the moment, we shall avoid the pitfalls of INITIALIZATION by default: we shall restrict ourselves to problems with a feasible origin. Problems with an infeasible origin are discussed on pages 39–42.

Iteration

Given some feasible dictionary, we have to select an entering variable, to find a leaving variable, and to construct the next feasible dictionary by pivoting.

Choosing an entering variable. The entering variable is *a nonbasic variable x_j with a positive coefficient $\bar{c}_j$ in the last row of the current dictionary.* This rule is ambiguous in the sense that it may provide more than one candidate for entering the basis, or no candidate at all. The latter alternative implies that the current dictionary describes an optimal solution, at which point the method may terminate. More precisely, consider the last row of our current dictionary,

$$z = z^* + \sum_{j \in N} \bar{c}_j x_j$$

with N standing for the set of subscripts j of nonbasic variables x_j. Our current solution, with $x_j = 0$ whenever $j \in N$, gives the objective function the numerical value of z^*. If $\bar{c}_j \le 0$ whenever $j \in N$, then every feasible solution, with $x_j \ge 0$

whenever $j \in N$, gives the objective function a numerical value of at most z^*; hence the current solution is optimal. On the other hand, if there is more than one candidate for entering the basis, then any of these candidates may serve. (In hand calculations involving small problems, it is customary to choose the candidate x_j that has the largest coefficient $\bar{c}_j$. In most computer implementations of the simplex method, however, this practice is abandoned. More on this subject in Chapter 7.)

Finding the leaving variable. The leaving variable is *that basic variable whose nonnegativity imposes the most stringent upper bound on the increase of the entering variable.* Again, this rule is ambiguous in the sense that it may provide more than one candidate for leaving the basis, or no candidate at all. The latter alternative is illustrated on the dictionary

$$\begin{array}{rcl}
x_2 &=& 5 + 2x_3 - x_4 - 3x_1 \\
x_5 &=& 7 \qquad\quad - 3x_4 - 4x_1 \\
\hline
z &=& 5 + x_3 - x_4 - x_1.
\end{array}$$

The entering variable is x_3, but neither of the two basic variables x_2, x_5 imposes an upper bound on its increase. Therefore, we can make x_3 as large as we wish (maintaining $x_1 = x_4 = 0$) and still retain feasibility: setting $x_3 = t$ for any positive t, we obtain a feasible solution with $x_1 = 0$, $x_2 = 5 + 2t$, $x_4 = 0$, $x_5 = 7$, and $z = 5 + t$. Since t can be made arbitrarily large, z can be made arbitrarily large. We conclude that the problem is *unbounded*: for every number M, there is a feasible solution $x_1, x_2, \ldots, x_5$ such that $x_3 - x_4 - x_1 > M$. The same conclusion can be reached in general: if there is no candidate for leaving the basis, then we can make the value of the entering variable, and therefore also the value of the objective function, as large as we wish. In that case, the problem is unbounded. On the other hand, if there is more than one candidate for leaving the basis, then any of these candidates may serve. Once the entering and leaving variables have been selected, pivoting is a straightforward matter.

Degeneracy. The presence of more than one candidate for leaving the basis has interesting consequences. For illustration, consider the dictionary

$$\begin{array}{rcl}
x_4 &=& 1 \qquad\qquad\qquad\quad - 2x_3 \\
x_5 &=& 3 - 2x_1 + 4x_2 - 6x_3 \\
x_6 &=& 2 + x_1 - 3x_2 - 4x_3 \\
\hline
z &=& \qquad 2x_1 - x_2 + 8x_3.
\end{array}$$

Having chosen x_3 to enter the basis, we find that each of the three basic variables x_4, x_5, x_6 limits the increase of x_3 to $\frac{1}{2}$. Hence each of these three variables is a candidate for leaving the basis. We arbitrarily choose x_4. Pivoting as usual, we obtain the dictionary

$$
\begin{array}{rcl}
x_3 &=& 0.5 \qquad\qquad\qquad\quad - 0.5x_4 \\
x_5 &=& \quad - 2x_1 + 4x_2 + \;\; 3x_4 \\
x_6 &=& \qquad\quad x_1 - 3x_2 + \;\; 2x_4 \\
\hline
z &=& 4 + 2x_1 - \;\; x_2 - \;\; 4x_4.
\end{array}
$$

This dictionary differs from all the dictionaries we have encountered so far in one important respect: along with the nonbasic variables, the basic variables x_5 and x_6 have value zero in the associated solution. Basic solutions with one or more basic variables at zero are called *degenerate*.

Although harmless in its own right, degeneracy may have annoying side effects. These are illustrated on the next iteration in our example. There, x_1 enters the basis and x_5 leaves; because of degeneracy, the constraint $x_5 \geq 0$ limits the increment of x_1 to *zero*. Hence the value of x_1 will remain unchanged, and so will the values of the remaining variables and the value of the objective function z. This is annoying, for the motivation behind the simplex method is a desire to increase the value of z in each iteration. In this particular iteration, that desire remains unfulfilled: pivoting changes the dictionary into

$$
\begin{array}{rcl}
x_1 &=& \qquad\quad 2x_2 + 1.5x_4 - 0.5x_5 \\
x_3 &=& 0.5 \qquad\quad - 0.5x_4 \\
x_6 &=& \quad - \;\; x_2 + 3.5x_4 - 0.5x_5 \\
\hline
z &=& 4 + 3x_2 - \qquad x_4 - \qquad x_5
\end{array}
$$

but it does not affect the associated solution at all. Simplex iterations that do not change the basic solution are called *degenerate*. (As the reader may verify, the next iteration is degenerate again, but the one after that turns out to be nondegenerate and brings us to the optimal solution.)

In a sense, degeneracy is something of an accident: a basic variable may vanish only if the results of successive pivot operations just happen to cancel each other out. And yet degeneracy abounds in LP problems arising from practical applications. It has been said that nearly all such problems yield degenerate basic feasible solutions at some stage of the simplex method. Whenever that happens, the simplex method may stall by going through a few (and sometimes quite a few) degenerate iterations in a row. Typically, such a block of degenerate iterations ends with a breakthrough represented by a nondegenerate iteration; an example of the atypical case is presented next.

Termination: Cycling

Can the simplex method go through an endless sequence of iterations without ever finding an optimal solution? Yes, it can. To justify this claim, let us consider the initial dictionary

$$\begin{array}{rcl}
x_5 &=& - 0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 \\
x_6 &=& - 0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 \\
x_7 &=& 1 - x_1 \\
\hline
z &=& 10x_1 - 57x_2 - 9x_3 - 24x_4
\end{array}$$

and let us agree on the following:

(i) The entering variable will always be the nonbasic variable that has the largest coefficient in the z-row of the dictionary.

(ii) If two or more basic variables compete for leaving the basis, then the candidate with the smallest subscript will be made to leave.

Now the sequence of dictionaries constructed in the first six iterations goes as follows.
After the first iteration:

$$\begin{array}{rcl}
x_1 &=& 11x_2 + 5x_3 - 18x_4 - 2x_5 \\
x_6 &=& - 4x_2 - 2x_3 + 8x_4 + x_5 \\
x_7 &=& 1 - 11x_2 - 5x_3 + 18x_4 + 2x_5 \\
\hline
z &=& 53x_2 + 41x_3 - 204x_4 - 20x_5.
\end{array}$$

After the second iteration:

$$\begin{array}{rcl}
x_2 &=& - 0.5x_3 + 2x_4 + 0.25x_5 - 0.25x_6 \\
x_1 &=& - 0.5x_3 + 4x_4 + 0.75x_5 - 2.75x_6 \\
x_7 &=& 1 + 0.5x_3 - 4x_4 - 0.75x_5 - 13.25x_6 \\
\hline
z &=& 14.5x_3 - 98x_4 - 6.75x_5 - 13.25x_6.
\end{array}$$

After the third iteration:

$$\begin{array}{rcl}
x_3 &=& 8x_4 + 1.5x_5 - 5.5x_6 - 2x_1 \\
x_2 &=& - 2x_4 - 0.5x_5 + 2.5x_6 + x_1 \\
x_7 &=& 1 - x_1 \\
\hline
z &=& 18x_4 + 15x_5 - 93x_6 - 29x_1.
\end{array}$$

After the fourth iteration:

$$\begin{array}{rcl}
x_4 &=& - 0.25x_5 + 1.25x_6 + 0.5x_1 - 0.5x_2 \\
x_3 &=& - 0.5x_5 + 4.5x_6 + 2x_1 - 4x_2 \\
x_7 &=& 1 - x_1 \\
\hline
z &=& 10.5x_5 - 70.5x_6 - 20x_1 - 9x_2.
\end{array}$$

After the fifth iteration:

$$\begin{array}{rl}
x_5 = & 9x_6 + 4x_1 - 8x_2 - 2x_3 \\
x_4 = & -\ x_6 - 0.5x_1 + 1.5x_2 + 0.5x_3 \\
x_7 = & 1 - x_1 \\
\hline
z = & 24x_6 + 22x_1 - 93x_2 - 21x_3.
\end{array}$$

After the sixth iteration:

$$\begin{array}{rl}
x_6 = & -\ 0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 \\
x_5 = & -\ 0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 \\
x_7 = & 1 - x_1 \\
\hline
z = & 10x_1 - 57x_2 - 9x_3 - 24x_4.
\end{array}$$

Since the dictionary constructed after the sixth iteration is identical with the initial dictionary, the method will go through the same six iterations again and again without ever finding the optimal solution (which, as we shall see later, has $z = 1$). This phenomenon is known as cycling. More precisely, we say that the simplex method *cycles* if one dictionary appears in two different iterations (and so the sequence of iterations leading from the dictionary to itself can be repeated over and over without end). Note that cycling can occur only in the presence of degeneracy: since the value of the objective function increases with each nondegenerate iteration and remains unchanged after each degenerate one, all the iterations in the sequence leading from a dictionary to itself must be degenerate. Cycling is one reason why the simplex method may fail to terminate; the following theorem shows that it is the only reason.

THEOREM 3.1. If the simplex method fails to terminate, then it must cycle.

PROOF. To begin, note that there are only finitely many ways of choosing m basic variables from all the $n + m$ variables. Thus, if the simplex method fails to terminate, then some basis must appear in two different iterations. Now it only remains to be proved that any two dictionaries with the same basis must be identical. This fact becomes trivial as soon as one describes dictionaries in terms of matrices, as we shall do in Chapter 7. Nevertheless, we can and shall present an easy proof from scratch right now.) Consider two dictionaries

$$\begin{array}{l} x_i = b_i - \sum_{j \notin B} a_{ij} x_j \qquad (i \in B) \\ \hline z \;= v + \sum_{j \notin B} c_j x_j \end{array} \tag{3.2}$$

and

$$\begin{array}{l} x_i = b_i^* - \sum_{j \notin B} a_{ij}^* x_j \qquad (i \in B) \\ \hline z \;= v^* + \sum_{j \notin B} c_j^* x_j \end{array} \tag{3.3}$$

with the same set of basic variables x_i ($i \in B$). It is a defining property of dictionaries that every solution $x_1, x_2, \ldots, x_{n+m}, z$ of (3.2) is a solution of (3.3) and vice versa. In particular, if x_k is a nonbasic variable and if t is a number, then the numbers

$$x_k = t, \quad x_j = 0\,(j \notin B \quad \text{and} \quad j \neq k), \quad x_i = b_i - a_{ik}t\ (i \in B), \quad z = v + c_k t,$$

constituting a solution of (3.2), must satisfy (3.3). Hence,

$$b_i - a_{ik}t = b_i^* - a_{ik}^* t \quad \text{for all} \quad i \in B, \quad \text{and} \quad v + c_k t = v^* + c_k^* t.$$

Since these identities must hold for all numbers t, we have

$$b_i = b_i^*, a_{ik} = a_{ik}^* \quad \text{for all} \quad i \in B, \quad \text{and} \quad v = v^*, c_k = c_k^*.$$

Since x_k was an arbitrary nonbasic variable, the two dictionaries are identical. ■

Cycling is a rare phenomenon. In fact, constructing an LP problem on which the simplex method may cycle is difficult. [Our example is adapted from K. T. Marshall and J. W. Suurballe (1969). The first example of this size was constructed by E. M. L. Beale (1955) and the first example ever was constructed by A. J. Hoffman (1953). Incidentally, Marshall and Suurballe (1969) proved that if the simplex method cycles off-optimum on a problem that has an optimal solution, then the dictionaries must involve at least six variables and at least three equations.] P. Wolfe (1963) and T. C. T. Kotiah and D. I. Steinberg (1978) reported having come across practical problems that cycled (in 25 and 18 iterations, respectively) but such reports are scarce. For this reason, the remote possibility of cycling is disregarded in most computer implementations of the simplex method.

There are ways of preventing the occurrence of cycling altogether. The classic *perturbation method and lexicographic method* avoid cycling by a judicious choice of the leaving variable in each simplex iteration; the more recent *smallest-subscript rule* does so by an easy choice of *both* the entering and the leaving variables. The former alternative maintains the freedom of choice among different candidates for entering the basis, but it requires extra computations to choose the leaving variable; the latter alternative requires no extra work at all, but it gives up the multitude of choices for the entering variable. We shall explain the details of both.

The perturbation method and the lexicographic method. The perturbation and the lexicographic methods are closely related. The perturbation method, suggested first by A. Orden and developed independently by A. Charnes (1952), provides an intuitive motivation for the lexicographic method of G. B. Dantzig, A. Orden, and P. Wolfe (1955). The lexicographic method can be seen as an implementation of the perturbation method.

The starting point relies on the observations that cycling can be stamped out by stamping out degeneracy and that degeneracy itself is something of an accident. To elaborate on the second observation, consider a degenerate dictionary. The basic variables currently at zero would most likely assume small nonzero values if the initial right-hand sides, b_i, were changed slightly; at the same time, if these changes were truly microscopic, then the problem could be considered unchanged for all practical purposes. One way of exploiting these observations is to add a small positive ε to each b_i, and then to apply the simplex method to the resulting problem. This trick (with $\varepsilon = 10^{-6}$ or so) is actually used in some computer implementations of the simplex method; it helps to reduce the number of degenerate iterations. Nevertheless, it does not constitute a reliable safeguard against cycling: for instance, if the simplex method is applied to the problem

$$
\begin{array}{llrrrrl}
\text{maximize} & 10x_1 & - \; 57x_2 & - \; 9x_3 & - \; 24x_4 & + \; 100x_5 & \\
\text{subject to} & & & & & x_5 & \leq 1 + \varepsilon \\
& 0.5x_1 & - \; 5.5x_2 & - \; 2.5x_3 & + \; 9x_4 & + \; x_5 & \leq 1 + \varepsilon \\
& 0.5x_1 & - \; 1.5x_2 & - \; 0.5x_3 & + \; x_4 & + \; x_5 & \leq 1 + \varepsilon \\
& x_1 & & & & + \; x_5 & \leq 2 + \varepsilon \\
& & & & & x_1, x_2, \ldots, x_5 & \geq 0
\end{array}
$$

Then the degenerate dictionary

$$
\begin{array}{rcl}
x_5 & = & 1 + \varepsilon \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad - \; x_6 \\
x_7 & = & -\,0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 + x_6 \\
x_8 & = & -\,0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 + x_6 \\
x_9 & = & 1 - x_1 + x_6 \\
\hline
z & = & 100 + 100\varepsilon + 10x_1 - 57x_2 - 9x_3 - 24x_4 - 100x_6
\end{array}
$$

is obtained after the first iteration and, as the reader may verify, the simplex method cycles in the next six iterations. (The cycle is essentially the same as that of the preceding example.)

What went wrong here was that the small amounts ε added to the right-hand sides cancelled each other out in the first iteration. To guarantee that such cancellations will never take place (and therefore all the dictionaries will remain nondegenerate), we shall perturb the different right-hand sides $b_1, b_2, \ldots, b_m$ by radically different amounts $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$. More precisely, we shall choose a very small ε_1 and then make each ε_{i+1} much smaller than the preceding ε_i: in symbols,

$$0 < \varepsilon_m \ll \varepsilon_{m-1} \ll \cdots \ll \varepsilon_2 \ll \varepsilon_1 \ll 1. \tag{3.4}$$

Then we shall apply the simplex method to the perturbed problem

$$\text{maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij}x_j \le b_i + \varepsilon_i \qquad (i = 1, 2, \ldots, m)$$

$$x_j \ge 0 \qquad (j = 1, 2, \ldots, n).$$

This is the *perturbation method.* (The perturbation method is usually presented with $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ equal to the powers $\varepsilon, \varepsilon^2, \ldots, \varepsilon^m$ of the same small ε. Our version makes the subsequent analysis a little more transparent.) For illustration, let us return to our first example on which the simplex method cycled. There, the initial dictionary reads

$$\begin{array}{rcl}
x_5 &=& \varepsilon_1 \qquad - 0.5x_1 + 5.5x_2 + 2.5x_3 - 9x_4 \\
x_6 &=& \varepsilon_2 \qquad - 0.5x_1 + 1.5x_2 + 0.5x_3 - x_4 \\
x_7 &=& 1 + \varepsilon_3 - x_1 \\
\hline
z &=& \qquad 10x_1 - 57x_2 - 9x_3 - 24x_4.
\end{array}$$

Again, the entering variable is x_1. The constraints $x_5 \ge 0$, $x_6 \ge 0$, and $x_7 \ge 0$ limit the increase of x_1 to $2\varepsilon_1$, $2\varepsilon_2$, and $1 + \varepsilon_3$, respectively. Since $2\varepsilon_2 < 2\varepsilon_1 < 1 + \varepsilon_3$, the leaving variable is x_6, and the next dictionary reads

$$\begin{array}{rcl}
x_1 &=& 2\varepsilon_2 \qquad + 3x_2 + x_3 - 2x_4 - 2x_6 \\
x_5 &=& \varepsilon_1 - \varepsilon_2 \qquad + 4x_2 + 2x_3 - 8x_4 + x_6 \\
x_7 &=& 1 - 2\varepsilon_2 + \varepsilon_3 - 3x_2 - x_3 + 2x_4 + 2x_6 \\
\hline
z &=& 20\varepsilon_2 \qquad - 27x_2 + x_3 - 44x_4 - 20x_6.
\end{array}$$

Now the only candidate for the entering variable is x_3 and the only candidate for the leaving variable is x_7. The resulting dictionary,

$$\begin{array}{rcl}
x_3 &=& 1 - 2\varepsilon_2 + \varepsilon_3 \qquad - 3x_2 + 2x_4 + 2x_6 - x_7 \\
x_1 &=& 1 + \varepsilon_3 \qquad - x_7 \\
x_5 &=& 2 + \varepsilon_1 - 5\varepsilon_2 + 2\varepsilon_3 - 2x_2 - 4x_4 + 5x_6 - 2x_7 \\
\hline
z &=& 1 + 18\varepsilon_2 + \varepsilon_3 \qquad - 30x_2 - 42x_4 - 18x_6 - x_7
\end{array}$$

is the optimal dictionary for the perturbed problem. It may be converted into the optimal dictionary for the original problem by simply disregarding all the terms involving $\varepsilon_1, \varepsilon_2, \varepsilon_3$.

How should we choose the numerical values of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$? The simplest answer is that we do not have to do that at all: rather than committing ourselves to definite values of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$, we may just think of these symbols as representing indefinite quantities, which satisfy (3.4). After several iterations of the simplex method, these symbols spread throughout the various rows of the dictionary, but they remain confined to the absolute terms in each of the $m + 1$ rows; the coefficients at the nonbasic variables in the dictionary are unaffected by the perturbation. Now when it comes to finding the leaving variable, each of the constraints $x_i \ge 0$ for a nonbasic x_i limits the increase of the entering x_j to a quantity such as $2\varepsilon_1$, $2\varepsilon_2$, $1 + \varepsilon_3$, or, more generally,

$$r = r_0 + r_1\varepsilon_1 + \cdots + r_m\varepsilon_m, \quad s = s_0 + s_1\varepsilon_1 + \cdots + s_m\varepsilon_m \tag{3.5}$$

and so on. As we are about to explain, assumption (3.4) allows us to compare the numerical values of such quantities without referring to the precise values of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$. If r and s in (3.5) are distinct, then there is the smallest subscript k such that $r_k \ne s_k$. It is customary to say that r is *lexicographically smaller* than s if $r_k < s_k$. (The choice of the term *lexicographically* is explained by observing that, for instance, $2 + 21\varepsilon_1 + 19\varepsilon_2 + 20\varepsilon_2$ is lexicographically smaller than $2 + 21\varepsilon_1 + 20\varepsilon_2 + 20\varepsilon_3 + 15\varepsilon_4 + 14\varepsilon_5$ for the same reason that "bust" comes before "button" in a dictionary.) It is easy to prove that r is lexicographically smaller than s if and only if r is numerically smaller than s for all values of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ that satisfy (3.4). This statement has to be made precise by specifying just what is meant by the symbol $\ll$ in (3.4); we leave the details for problem 3.7.

The *lexicographic method* is that implementation of the perturbation method in which $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ are treated as symbols, and quantities such as r and s in (3.5) are compared by the lexicographic rule. Note that it is always possible to choose the leaving variable by the lexicographic rule: in every finite set of expressions such as r and s in (3.5), there is always one that is lexicographically smaller than or equal to all the others. Even though this fact may be taken for granted intuitively, rigor requires that it be proved; we leave the details for problem 3.6. Another fine point concerns the behavior of the objective function z. The value of z, equal to some expression $v_0 + v_1\varepsilon_1 + \cdots + v_m\varepsilon_m$, remains unchanged in each degenerate iteration and increases, in the lexicographic sense, with each nondegenerate one. (In our example, the increase from 0 to $20\varepsilon_2$ in the first iteration was followed by the increase from $20\varepsilon_2$ to $1 + 18\varepsilon_2 + \varepsilon_3$ in the second iteration.) It is intuitively obvious that the total of two or more lexicographic increases is a lexicographic increase; a rigorous proof of this fact follows from the result of problem 3.5. Now it follows that, even in the generalized context of the lexicographic method, cycling is possible only in the presence of degeneracy. Finally, note that the only function of the terms involving $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ is to guide us toward the appropriate choice of a leaving variable whenever two or more candidates present themselves in the original problem. If, at any moment, these terms are deleted, then the dictionary for the perturbed problem reduces to a dictionary for the original problem.

THEOREM 3.2. The simplex method terminates as long as the leaving variable is selected by the lexicographic rule in each iteration.

PROOF. In view of the preceding remarks, we need merely prove that no degenerate dictionary will be constructed. (If all dictionaries are nondegenerate, then all iterations are nondegenerate. In that case, cycling cannot occur and the desired conclusion follows from Theorem 3.1.) Thus, we need only consider an arbitrary row

$$x_k = (r_0 + r_1\varepsilon_1 + \cdots + r_m\varepsilon_m) - \sum_{j \notin B} d_j x_j \tag{3.6}$$

of an arbitrary dictionary and to prove that at least one of the $m + 1$ numbers $r_0, r_1, \ldots, r_m$ is distinct from zero. (Actually, we shall prove that at least one of the m numbers $r_1, r_2, \ldots, r_m$ is distinct from zero.) Writing $d_k = 1$ and $d_i = 0$ for all basic variables x_i distinct from x_k, we record (3.6) as

$$\sum_{j=1}^{n+m} d_j x_j = r_0 + \sum_{i=1}^{m} r_i \varepsilon_i. \tag{3.7}$$

Since this equation has been obtained by algebraic manipulations from the definitions of the slack variables,

$$x_{n+i} = b_i + \varepsilon_i - \sum_{j=1}^{n} a_{ij} x_j \qquad (i = 1, 2, \ldots, m) \tag{3.8}$$

it must hold for all choices of numbers $x_1, x_2, \ldots, x_{n+m}$ and $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ that satisfy (3.8). Hence, the equation

$$\sum_{j=1}^{n} d_j x_j + \sum_{i=1}^{m} d_{n+i}(b_i + \varepsilon_i - \sum_{j=1}^{n} a_{ij}x_j) = r_0 + \sum_{i=1}^{m} r_i \varepsilon_i$$

which is obtained by substituting from (3.8) into (3.7), must hold for all choices of numbers x_1, $x_2, \ldots, x_n$ and $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$. Writing this identity as

$$\sum_{j=1}^{n} (d_j - \sum_{i=1}^{m} d_{n+i}a_{ij})x_j + \sum_{i=1}^{m} (d_{n+i} - r_i)\varepsilon_i = r_0 - \sum_{i=1}^{m} d_{n+i}b_i$$

we observe that the coefficient at each x_j, the coefficient at each ε_i, and the right-hand side must equal zero. Thus

$$\begin{aligned} d_{n+i} &= r_i && \text{for all} \quad i = 1, 2, \ldots, m \\ d_j &= \sum_{i=1}^{m} d_{n+i}a_{ij} && \text{for all} \quad j = 1, 2, \ldots, n. \end{aligned} \tag{3.9}$$

If all the numbers $r_1, r_2, \ldots, r_m$ were equal to zero, then (3.9) would imply $d_{n+i} = 0$ for all $i = 1, 2, \ldots, m$ and $d_j = 0$ for all $j = 1, 2, \ldots, n$, contradicting the fact that $d_k = 1$. ■

With the hindsight provided by Theorem 3.2, it becomes easy to prove that every LP problem in the standard form can be perturbed by adding suitable small *numbers* $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ to the right-hand sides $b_1, b_2, \ldots, b_m$ in such a way that the simplex method applied to the perturbed problem will terminate. In fact, the numbers $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ may be chosen as the powers $\varepsilon, \varepsilon^2, \ldots, \varepsilon^m$ of any sufficiently small positive ε. We leave the details for problem 3.8.

As we have observed, the terms involving $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ are needed only when a tie has to be broken between two or more candidates for leaving the basis. Thus we might just as well wait until such a need arises, and only then introduce an ad hoc perturbation. This idea was developed by P. Wolfe (1963); its lexicographic counterpart comes from G. B. Dantzig (1960).

Smallest-subscript rule. This term will refer to breaking ties in the choice of the entering and leaving variables by always choosing the candidate x_k that has the smallest subscript k. The motivation for this elegant concept is provided by the following result.

THEOREM 3.3. [R. G. Bland (1977).] The simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration.

PROOF. By virtue of Theorem 3.1, we need only show that cycling is impossible when the smallest-subscript rule is used. We shall do this by deriving a contradiction from the assumption that the smallest-subscript rule leads from some dictionary D_0 to itself in a sequence of degenerate iterations. For definiteness, let us say that this sequence of iterations produces dictionaries D_1, $D_2, \ldots, D_k$ such that $D_k = D_0$. A variable will be called *fickle* if it is nonbasic in some of these dictionaries and basic in others. Among all the fickle variables, let x_t have the largest subscript. In the sequence $D_0, D_1, \ldots, D_k$, there is a dictionary D with x_t leaving (basic in D but nonbasic in the next dictionary), and some other fickle variable x_s entering (nonbasic in D but basic in the next dictionary). Further along in the sequence $D_0, D_1, \ldots, D_k, D_1, D_2, \ldots, D_k$, there must be a dictionary D^* with x_t entering. Let us record D as

$$\begin{array}{ll} x_i = b_i - \sum_{j \notin B} a_{ij}x_j & (i \in B) \\ \hline z \;= v + \sum_{j \notin B} c_j x_j \, . & \end{array}$$

Since all the iterations leading from D to D^* are degenerate, the objective function z must have the same value v in both dictionaries. Thus, the last row of D^* may be recorded as

$$z = v + \sum_{j=1}^{n+m} c_j^* x_j$$

with $c_j^* = 0$ whenever x_j is basic in D^*. Since this equation has been obtained from D by algebraic manipulations, it must be satisfied by every solution of D. In particular, it must be satisfied by $x_s = y$, $x_j = 0$ ($j \notin B$ but $j \neq s$), $x_i = b_i - a_{is}y$ ($i \in B$) and $z = v + c_s y$ for every choice of y. Thus we have

$$v + c_s y = v + c_s^* y + \sum_{i \in B} c_i^*(b_i - a_{is}y)$$

and, after simplification,

$$\left(c_s - c_s^* + \sum_{i \in B} c_i^* a_{is}\right) y = \sum_{i \in B} c_i^* b_i$$

for every choice of y. Since the right-hand side of the last equation is a constant independent of y, we conclude that

$$c_s - c_s^* + \sum_{i \in B} c_i^* a_{is} = 0. \tag{3.10}$$

The rest is easy. Since x_s is entering in D, we have $c_s > 0$. Since x_s is not entering in D^* and yet $s < t$, we have $c_s^* \leq 0$. Hence (3.10) implies that

$$c_r^* a_{rs} < 0 \qquad \text{for some} \quad r \in B. \tag{3.11}$$

Since $r \in B$, the variable x_r is basic in D; since $c_r^* \neq 0$, the same variable is nonbasic in D^*. Hence, x_r is fickle and we have $r \leq t$. Actually, x_r is different from x_t: since x_t is leaving in D, we have $a_{ts} > 0$ and so $c_t^* a_{ts} > 0$. Now $r < t$ and yet x_r is not entering in D^*. Thus, we cannot have $c_r^* > 0$. From (3.11), we conclude that

$$a_{rs} > 0.$$

Since all the iterations leading from D to D^* are degenerate, the two dictionaries describe the same solution. In particular, the value of x_r is zero in both dictionaries (x_r is nonbasic in D^*) and so $b_r = 0$. Hence x_r was a candidate for leaving the basis of D—yet we picked x_t, even though $r < t$. This contradiction completes the proof. ■

One further point: termination of the simplex method can be guaranteed even without abiding by the smallest-subscript rule in every single iteration. We might resort to the smallest-subscript rule, for instance, only when the last fifty or so iterations were degenerate, and abandon it after the next nondegenerate iteration in favor of any other way of choosing the entering and leaving variables. Although cycling might conceivably take place in this case, each block of consecutive degenerate iterations would be followed by a nondegenerate iteration, and so each dictionary could be constructed only a finite number of times.

Initialization

The only remaining point that needs to be explained is getting hold of the initial feasible dictionary in a problem

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \le b_i & (i = 1, 2, \dots, m) \\ & x_j \ge 0 & (j = 1, 2, \dots, n) \end{array}$$

with an infeasible origin. The trouble with an infeasible origin is twofold. First, it may not be clear that our problem has any feasible solutions at all. Second, even if a feasible solution is apparent, a feasible dictionary may not be. One way of getting around both obstacles uses a so-called *auxiliary problem,*

$$\begin{array}{lll} \text{minimize} & x_0 & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j - x_0 \le b_i & (i = 1, 2, \dots, m) \\ & x_j \ge 0 & (j = 0, 1, \dots, n). \end{array}$$

A feasible solution of the auxiliary problem is readily available: it suffices to set the value of each x_j with $1 \le j \le n$ at zero and make the value of x_0 sufficiently large. Furthermore, it is easy to see that the original problem has a feasible solution *if and only if* the auxiliary problem has a feasible solution with $x_0 = 0$. To put it differently, the original problem has a feasible solution if and only if the optimum value of the auxiliary problem is zero. Hence our plan is to solve the auxiliary problem first; the technical details are illustrated on the problem

$$\begin{array}{lrcrcrcr} \text{maximize} & x_1 & - & x_2 & + & x_3 & & \\ \text{subject to} & 2x_1 & - & x_2 & + & 2x_3 & \le & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & \le & -5 \\ & -x_1 & + & x_2 & - & 2x_3 & \le & -1 \\ & & & \multicolumn{3}{r}{x_1, x_2, x_3} & \ge & 0. \end{array}$$

To avoid unnecessary confusion, we write the auxiliary problem in its maximization form:

$$\begin{array}{lrcrcrcrcr} \text{maximize} & -x_0 & & & & & & & & \\ \text{subject to} & 2x_1 & - & x_2 & + & 2x_3 & - & x_0 & \le & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & - & x_0 & \le & -5 \\ & -x_1 & + & x_2 & - & 2x_3 & - & x_0 & \le & -1 \\ & & & & & \multicolumn{3}{r}{x_0, x_1, x_2, x_3} & \ge & 0. \end{array}$$

Writing down the formulas defining the slack variables x_4, x_5, x_6 and the objective function w, we obtain the dictionary

$$\begin{array}{rcl}
x_4 &=& 4 - 2x_1 + x_2 - 2x_3 + x_0 \\
x_5 &=& -5 - 2x_1 + 3x_2 - x_3 + x_0 \\
x_6 &=& -1 + x_1 - x_2 + 2x_3 + x_0 \\
\hline
w &=& - x_0
\end{array}$$

which is infeasible. Nevertheless, this infeasible dictionary can be transformed into a feasible one by a single pivot, with x_0 entering and x_5 leaving the basis:

$$\begin{array}{rcl}
x_0 &=& 5 + 2x_1 - 3x_2 + x_3 + x_5 \\
x_4 &=& 9 - 2x_2 - x_3 + x_5 \\
x_6 &=& 4 + 3x_1 - 4x_2 + 3x_3 + x_5 \\
\hline
w &=& -5 - 2x_1 + 3x_2 - x_3 - x_5.
\end{array}$$

In general, the auxiliary problem may be written as

$$\begin{array}{lll}
\text{maximize} & -x_0 & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij}x_j - x_0 \leq b_i & (i = 1, 2, \ldots, m) \\
& x_j \geq 0 & (j = 0, 1, \ldots, n).
\end{array}$$

Writing down the formulas defining the slack variables $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ and the objective function w gives us the dictionary

$$\begin{array}{rcll}
x_{n+i} &=& b_i - \sum_{j=1}^{n} a_{ij}x_j + x_0 & (i = 1, 2, \ldots, m) \\
\hline
w &=& \phantom{b_i - \sum a_{ij}x_j} - x_0 &
\end{array}$$

which is infeasible. Nevertheless, this infeasible dictionary can be transformed into a feasible one by a single pivot, with x_0 entering and the "most infeasible" x_{n+i} leaving the basis. More precisely, the leaving variable is that x_{n+k} whose negative value, b_k, has the largest magnitude among all the negative numbers b_i. After pivoting, the variable x_0 assumes the positive value of $-b_k$, whereas each basic x_{n+i} assumes the nonnegative value of $b_i - b_k$. Now we are set to solve the auxiliary problem by the simplex method. In our illustrative example, the computations go as follows.

After the first iteration, with x_2 entering and x_6 leaving:

$$\begin{array}{rcl}
x_2 &=& 1 + 0.75x_1 + 0.75x_3 + 0.25x_5 - 0.25x_6 \\
x_0 &=& 2 - 0.25x_1 - 1.25x_3 + 0.25x_5 + 0.75x_6 \\
x_4 &=& 7 - 1.5x_1 - 2.5x_3 + 0.5x_5 + 0.5x_6 \\
\hline
w &=& -2 + 0.25x_1 + 1.25x_3 - 0.25x_5 - 0.75x_6.
\end{array}$$

After the second iteration, with x_3 entering and x_0 leaving:

$$\begin{array}{rcl} x_3 &=& 1.6 - 0.2x_1 + 0.2x_5 + 0.6x_6 - 0.8x_0 \\ x_2 &=& 2.2 + 0.6x_1 + 0.4x_5 + 0.2x_6 - 0.6x_0 \\ x_4 &=& 3 - x_1 - x_6 + 2x_0 \\ \hline w &=& - x_0. \end{array} \tag{3.12}$$

The last dictionary (3.12) is optimal. Since the optimal value of the auxiliary problem is zero, dictionary (3.12) points out a feasible solution of the original problem: $x_1 = 0, x_2 = 2.2, x_3 = 1.6$. Furthermore, (3.12) can be easily converted into the desired feasible dictionary of the original problem. To obtain the first three rows of the desired dictionary, we simply copy down the first three rows of (3.12), omitting all the terms involving x_0:

$$\begin{array}{rcl} x_3 &=& 1.6 - 0.2x_1 + 0.2x_5 + 0.6x_6 \\ x_2 &=& 2.2 + 0.6x_1 + 0.4x_5 + 0.2x_6 \\ x_4 &=& 3 - x_1 - x_6. \end{array} \tag{3.13}$$

To obtain the last row, we have to express the original objective function

$$z = x_1 - x_2 + x_3 \tag{3.14}$$

in terms of the nonbasic variables x_1, x_5, x_6. For this purpose, we simply substitute from (3.13) into (3.14), obtaining

$$\begin{aligned} z &= x_1 - (2.2 + 0.6x_1 + 0.4x_5 + 0.2x_6) + (1.6 - 0.2x_1 + 0.2x_5 + 0.6x_6) \\ &= -0.6 + 0.2x_1 - 0.2x_5 + 0.4x_6. \end{aligned}$$

In short, the desired dictionary reads

$$\begin{array}{rcl} x_3 &=& 1.6 - 0.2x_1 + 0.2x_5 + 0.6x_6 \\ x_2 &=& 2.2 + 0.6x_1 + 0.4x_5 + 0.2x_6 \\ x_4 &=& 3 - x_1 - x_6 \\ \hline z &=& -0.6 + 0.2x_1 - 0.2x_5 + 0.4x_6. \end{array}$$

Clearly, the same procedure will transform an optimal dictionary of the auxiliary problem into a feasible dictionary of the original problem whenever x_0 is nonbasic in the former.

Now, let us review the general situation. We have learned how to construct the auxiliary problem and its first feasible dictionary. In the process of solving the auxiliary problem, we may encounter a dictionary where x_0 competes with other variables for leaving the basis. If and when that happens, it is only natural to choose x_0 as the actual leaving variable; immediately after pivoting, we obtain a dictionary where

x_0 is nonbasic, and so the value of w is zero. (3.15)

Clearly, a feasible dictionary with this property is optimal. However, we may also reach the optimum of the auxiliary problem while x_0 is still basic. Thus, we may obtain an optimal dictionary where

$$x_0 \text{ is basic and the value of } w \text{ is nonzero} \tag{3.16}$$

or, conceivably, an optimal dictionary where

$$x_0 \text{ is basic and the value of } w \text{ is zero.} \tag{3.17}$$

Let us examine case (3.17). Since the next-to-last dictionary was not yet optimal, the value of $w = -x_0$ must have changed from some negative level to zero in the last iteration. To put it differently, the value of the basic variable x_0 must have dropped from some positive level to zero in the last iteration. But then x_0 was a candidate for leaving the basis; yet, contrary to our policy, we did not pick it. This contradiction shows that (3.17) cannot occur. Hence the optimal dictionary of the auxiliary problem has either property (3.15) or property (3.16). In the former case, we construct a feasible dictionary of the original problem as illustrated previously and proceed to solve the original problem by the simplex method; in the latter case, we simply conclude that the original problem is infeasible.

This strategy is known as the *two-phase simplex method*. In the *first phase*, we set up and solve the auxiliary problem; if the optimal dictionary turns out to have property (3.15) then we proceed to the *second phase*, solving the original problem itself. We shall return to the two-phase simplex method in Chapter 8.

THE FUNDAMENTAL THEOREM OF LINEAR PROGRAMMING

This name is given to the following result.

THEOREM 3.4. Every LP problem in the standard form has the following three properties:

(i) If it has no optimal solution, then it is either infeasible or unbounded.

(ii) If it has a feasible solution, then it has a basic feasible solution.

(iii) If it has an optimal solution, then it has a basic optimal solution.

PROOF. The first phase of the two-phase simplex method either discovers that the problem is infeasible or else it delivers a basic feasible solution. The second phase of the two-phase simplex method either discovers that the problem is unbounded or else it delivers a basic optimal solution. ■

Note that the first property is not shared by problems whose constraints may include *strict* linear inequalities $\sum a_j x_j < b$. To take a trivial example, the problem

$$\text{maximize } x \quad \text{subject to } x < 0$$

is neither infeasible nor unbounded and yet it has no optimal solution. The remaining two properties (ii) and (iii) tell us that, when looking for feasible or optimal solutions of an LP problem in the standard form, we may confine our search to a *finite* set. These two properties, easy to establish from scratch, are often used to motivate the simplex method. Our exposition has followed the reverse pattern, with an emphasis placed on actually solving the problem—and the fundamental theorem of linear programming obtained as an effortless afterthought. □

PROBLEMS

△ **3.1** Maximize $x_1 + 3x_2 - x_3$

subject to

$$\begin{aligned} 2x_1 + 2x_2 - x_3 &\le 10 \\ 3x_1 - 2x_2 + x_3 &\le 10 \\ x_1 - 3x_2 + x_3 &\le 10 \\ x_1, x_2, x_3 &\ge 0. \end{aligned}$$

3.2 In the tableau format, a natural tie-breaking rule for the choice of the pivot row favors the rows that appear higher up in the tableau. Show that in the following example (constructed by H. W. Kuhn), this tie-breaking rule leads to cycling:

maximize $2x_1 + 3x_2 - x_3 - 12x_4$

subject to

$$\begin{aligned} -2x_1 - 9x_2 + x_3 + 9x_4 &\le 0 \\ \frac{1}{3}x_1 + x_2 - \frac{1}{3}x_3 - 2x_4 &\le 0 \\ x_1, x_2, x_3, x_4 &\ge 0. \end{aligned}$$

3.3 Solve problem 3.2 by the perturbation technique.

3.4 Arrange the following expressions in a sequence from lexicographically smallest to lexicographically largest:

$$\begin{aligned} &3 - \varepsilon_1 \\ &3 \\ &2 + 10\varepsilon_1 \\ &3 - 4\varepsilon_1 + \varepsilon_2 \\ &\varepsilon_2 + 3\varepsilon_3 \\ &3 + 4\varepsilon_1 + \varepsilon_3 \\ &3 - 4\varepsilon_1 + \varepsilon_2 + \varepsilon_3. \end{aligned}$$

3.5 Prove: If $r = r_0 + r_1\varepsilon_1 + \cdots + r_m\varepsilon_m$ is lexicographically smaller than $s = s_0 + s_1\varepsilon_1 + \cdots + s_m\varepsilon_m$ and if s is lexicographically smaller than $t = t_0 + t_1\varepsilon_1 + \cdots + t_m\varepsilon_m$, then r is lexicographically smaller than t.

3.6 Use the result of problem 3.5 to prove that, in every finite set of distinct expressions, such as r and s in (3.5), there is an expression that is lexicographically smaller than all the others.

3.7 Prove that for every pair of expressions in (3.5) there is a positive number δ such that the following two statements are equivalent: (i) r is lexicographically smaller than s; (ii) for every choice of numbers $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ such that

$$0 < \varepsilon_1 < \delta \quad \text{and} \quad 0 < \varepsilon_i < \delta\varepsilon_{i-1} \qquad \text{for all } i = 2, 3, \ldots, m$$

r is numerically smaller than s.

3.8 Use Theorem 3.2 and the result of problem 3.7 to prove the following. For every LP problem

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \le b_i & (i = 1, 2, \ldots, m) \\ & x_j \ge 0 & (j = 1, 2, \ldots, n) \end{array}$$

there is a positive number δ such that the simplex method used to

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \le b_i + \varepsilon^i & (i = 1, 2, \ldots, m) \\ & x_j \ge 0 & (j = 1, 2, \ldots, n) \end{array}$$

terminates whenever $0 < \varepsilon < \delta$.

3.9 Solve the following problems by the two-phase simplex method:

a. $$\begin{array}{lrcrcr} \text{maximize} & 3x_1 & + & x_2 & & \\ \text{subject to} & x_1 & - & x_2 & \le & -1 \\ & -x_1 & - & x_2 & \le & -3 \\ & 2x_1 & + & x_2 & \le & 4 \\ & & & x_1, x_2 & \ge & 0 \end{array}$$

b. $$\begin{array}{lrcrcr} \text{maximize} & 3x_1 & + & x_2 & & \\ \text{subject to} & x_1 & - & x_2 & \le & -1 \\ & -x_1 & - & x_2 & \le & -3 \\ & 2x_1 & + & x_2 & \le & 2 \\ & & & x_1, x_2 & \ge & 0 \end{array}$$

c. $$\begin{array}{lrcrcr} \text{maximize} & 3x_1 & + & x_2 & & \\ \text{subject to} & x_1 & - & x_2 & \le & -1 \\ & -x_1 & - & x_2 & \le & -3 \\ & 2x_1 & - & x_2 & \le & 2 \\ & & & x_1, x_2 & \ge & 0. \end{array}$$

3.10 Prove or disprove: A feasible dictionary whose last row reads $z = z^* + \sum \bar{c}_j x_j$ describes an optimal solution if and only if $\bar{c}_j \le 0$ for all j.

4

Chapter 4 is not required reading.

How Fast Is the Simplex Method?

The subject of this chapter is the number of iterations in the simplex method. We shall also comment on the distinction between theoretically satisfactory and practically satisfactory algorithms, with a particular regard to linear programming.

TYPICAL NUMBER OF ITERATIONS

For *practical* problems of the form

$$\begin{aligned}
&\text{maximize} && \sum_{j=1}^{n} c_j x_j \\
&\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad (i = 1, 2, \ldots, m) \\
& && x_j \ge 0 \qquad (j = 1, 2, \ldots, n)
\end{aligned} \tag{4.1}$$

with $m < 50$ and $m + n < 200$, Dantzig (1963, p. 160) reported the number of iterations as being usually less than $3m/2$ and only rarely going to $3m$. This observation agrees with empirical findings obtained more recently for much larger problems: the typical number of iterations increases proportionally to m (with the proportionality constant in the range suggested by Dantzig) and only very slowly with n. (It is sometimes said that, for a fixed m, the typical number of iterations is proportional to the logarithm of n.) Theoretical explanations of this phenomenon were proposed by G. B. Dantzig (1980), K.-H. Borgwardt (1982) and S. Smale (1982). It is this remarkable efficiency of the simplex method that accounts for its staggering success. At the current level of computer technology, typical problems with about 100 constraints and variables are solved in a few seconds; even problems with several thousands of constraints can be handled successfully. (To attain this level of efficiency, the simplex method has to be implemented properly, so that the time *per iteration* is reduced as much as possible. Consequently, the format of dictionaries has to be abandoned in favor of less time-consuming ways of organizing the necessary computations. We shall begin to study this matter in Chapter 7.) For problems with some particular structure amenable to specialized versions of the simplex method (such as the network simplex method of Chapter 19 or generalized upper bounding of Chapter 25), this limit can be pushed even further.

Monte Carlo simulation studies of the number of iterations were pioneered by H. W. Kuhn and R. E. Quandt (1963), who solved a number of problems (4.1) with $c_j = 1$ for all j, $b_i = 10{,}000$ for all i, and each a_{ij} selected at random from the set of positive integers between 1 and 1,000. A small part of these experiments has been reproduced, on a slightly larger scale, with the results exhibited in Table 4.1. (Each entry in the table represents the average number of iterations over 100 problems.) In each simplex iteration, the entering variable was that nonbasic variable that had the largest coefficient in the z-row of the dictionary. We shall refer to this selection rule as the *largest-coefficient* rule.

TABLE 4.1 Average Number of Iterations Required by the Largest-Coefficient Rule

m \ n	10	20	30	40	50
10	9.40	14.2	17.4	19.4	20.2
20		25.2	30.7	38.0	41.5
30			44.4	52.7	62.9
40				67.6	78.7
50					95.2

Source: D. Avis and V. Chvátal (1978).

The production management problems solved in practice are very much different from such randomly generated examples. Typically, most of their coefficients a_{ij} are zeros, the remaining nonzero coefficients occur in clusters that are very far from random, and the range of distinct numerical values of the coefficients is often very small. In spite of these differences, the Monte Carlo simulation results are in striking agreement with the empirical observations quoted above: for instance, if $n = 50$, then the average number of iterations is about $2m$.

PROBLEMS REQUIRING AN UNUSUALLY LARGE NUMBER OF ITERATIONS

From a purist point of view, it would be even more reassuring to have a proof that, for *every* problem (4.1), the simplex method would require no more than, say, $10mn$ iterations to find an optimal solution. However, there is no such proof. Worse than that, there are examples of LP problems that make the simplex method go through an enormous number of iterations. V. Klee and G. J. Minty (1972) have shown that in the process of solving the problem

$$\begin{array}{lll} \text{maximize} & \sum_{j=1}^{n} 10^{n-j} x_j & \\ \text{subject to} & \left(2 \sum_{j=1}^{i-1} 10^{i-j} x_j\right) + x_i \le 100^{i-1} & (i = 1, 2, \ldots, n) \\ & x_j \ge 0 & (j = 1, 2, \ldots, n) \end{array} \tag{4.2}$$

the simplex method goes through $2^n - 1$ iterations. (A proof is outlined in problems 4.2 and 4.3.) This number is quite frightening. For example, at the rate of 100 iterations per second (a reasonably generous estimate), problem (4.2) with $n = 50$ would take more than 300,000 years to solve! (The empirical and simulation results just quoted do *not* contradict this result. They simply suggest that problems requiring large numbers of iterations must be rare. For this reason, the Klee–Minty examples (4.2) and other similar examples are sometimes referred to as "pathological.")

As our starting point for further discussion, we choose the Klee–Minty problem with $n = 3$,

$$\begin{array}{lrcl} \text{maximize} & 100x_1 + 10x_2 + x_3 & & \\ \text{subject to} & x_1 & \le & 1 \\ & 20x_1 + x_2 & \le & 100 \\ & 200x_1 + 20x_2 + x_3 & \le & 10{,}000 \\ & x_1, x_2, x_3 & \ge & 0. \end{array} \tag{4.3}$$

Using the largest-coefficient rule, we construct the following sequence of dictionaries The initial dictionary:

$$\begin{array}{rcrcrcrcl} x_4 & = & 1 & - & x_1 & & & & \\ x_5 & = & 100 & - & 20x_1 & - & x_2 & & \\ x_6 & = & 10{,}000 & - & 200x_1 & - & 20x_2 & - & x_3 \\ \hline z & = & & & 100x_1 & + & 10x_2 & + & x_3. \end{array}$$

After the first iteration:

$$\begin{array}{rcrcrcrcl} x_1 & = & 1 & - & x_4 & & & & \\ x_5 & = & 80 & + & 20x_4 & - & x_2 & & \\ x_6 & = & 9{,}800 & + & 200x_4 & - & 20x_2 & - & x_3 \\ \hline z & = & 100 & - & 100x_4 & + & 10x_2 & + & x_3. \end{array}$$

After the second iteration:

$$\begin{array}{rcrcrcrcl} x_1 & = & 1 & - & x_4 & & & & \\ x_2 & = & 80 & + & 20x_4 & - & x_5 & & \\ x_6 & = & 8{,}200 & - & 200x_4 & + & 20x_5 & - & x_3 \\ \hline z & = & 900 & + & 100x_4 & - & 10x_5 & + & x_3. \end{array}$$

After the third iteration:

$$\begin{array}{rcrcrcrcl} x_4 & = & 1 & - & x_1 & & & & \\ x_2 & = & 100 & - & 20x_1 & - & x_5 & & \\ x_6 & = & 8{,}000 & + & 200x_1 & + & 20x_5 & - & x_3 \\ \hline z & = & 1{,}000 & - & 100x_1 & - & 10x_5 & + & x_3. \end{array}$$

After the fourth iteration:

$$\begin{array}{rcrcrcrcl} x_4 & = & 1 & - & x_1 & & & & \\ x_2 & = & 100 & - & 20x_1 & - & x_5 & & \\ x_3 & = & 8{,}000 & + & 200x_1 & + & 20x_5 & - & x_6 \\ \hline z & = & 9{,}000 & + & 100x_1 & + & 10x_5 & - & x_6. \end{array}$$

After the fifth iteration:

$$\begin{array}{rcrcrcrcl} x_1 & = & 1 & - & x_4 & & & & \\ x_2 & = & 80 & + & 20x_4 & - & x_5 & & \\ x_3 & = & 8{,}200 & - & 200x_4 & + & 20x_5 & - & x_6 \\ \hline z & = & 9{,}100 & - & 100x_4 & + & 10x_5 & - & x_6. \end{array}$$

After the sixth iteration:

$$\begin{array}{rcl}
x_1 &=& 1 - x_4 \\
x_5 &=& 80 + 20x_4 - x_2 \\
x_3 &=& 9{,}800 + 200x_4 - 20x_2 - x_6 \\
\hline
z &=& 9{,}900 + 100x_4 - 10x_2 - x_6.
\end{array}$$

After the seventh iteration:

$$\begin{array}{rcl}
x_4 &=& 1 - x_1 \\
x_5 &=& 100 - 20x_1 - x_2 \\
x_3 &=& 10{,}000 - 200x_1 - 20x_2 - x_6 \\
\hline
z &=& 10{,}000 - 100x_1 - 10x_2 - x_6.
\end{array}$$

In the first iteration, we were led to an unfortunate choice of the entering variable: had we made x_3 rather than x_1 enter the basis, we would have pivoted directly to the final dictionary. In view of this blunder, it is natural to question the expediency of the largest-coefficient rule: perhaps the simplex method would *always* go through only a small number of iterations if it were directed by some other rule. In fact, the largest-coefficient rule is not quite natural. More specifically, it ranks the potential candidates for entering the basis according to their coefficients in the last row of the dictionary: variables with larger coefficients appear to be more promising. But appearances are misleading and the ranking order is easily upset by changes in the scale on which each candidate is measured. For instance, the substitution

$$\bar{x}_1 = x_1, \quad \bar{x}_2 = 0.01x_2, \quad \bar{x}_3 = 0.0001x_3$$

converts the Klee–Minty problem (4.3) into the form

$$\begin{array}{ll}
\text{maximize} & 100\bar{x}_1 + 1{,}000\bar{x}_2 + 10{,}000\bar{x}_3 \\
\text{subject to} & \bar{x}_1 \leq 1 \\
& 20\bar{x}_1 + 100\bar{x}_2 \leq 100 \\
& 200\bar{x}_1 + 2{,}000\bar{x}_2 + 10{,}000\bar{x}_3 \leq 10{,}000 \\
& \bar{x}_1, \bar{x}_2, \bar{x}_3 \geq 0.
\end{array}$$

In the first dictionary associated with this new version of (4.3), the nonbasic variable $\bar{x}_3$ appears most attractive, and so the simplex method reaches the optimal solution in only one iteration.

ALTERNATIVE PIVOTING RULES

Thus we are led to ranking the candidates x_j for entering the basis according to criteria that are independent of changes of scale. One criterion of this kind is the increase in the objective function obtained when x_j actually enters the basis. The resulting rule (always choose that candidate whose entrance into the basis brings about the largest increase in the objective function) is referred to as the *largest-increase* rule. On the Klee–Minty examples (4.2), the largest-increase rule leads the simplex method to the optimal solution in only one iteration, as opposed to the $2^n - 1$ iterations required by the previously used largest-coefficient rule. However, the new rule does not always lead to a small number of iterations: R. G. Jeroslow (1973) constructed LP problems that are to the largest-increase rule what the Klee–Minty problems are to the largest-coefficient rule. (More precisely, the number of iterations required by the largest-increase rule grows exponentially with m and n.) Again, these examples exploit the myopia inherent in the simplex method. It is conceivable that every easily implemented rule for choosing the entering variable can be tricked in a similar way into requiring very large numbers of iterations.

Which of the two rules is better? On problems arising from applications, the number of iterations required by the largest increase is *usually* smaller than the number of iterations required by the largest coefficient. Simulation experiments lead to a similar outcome (see Table 4.2).

Table 4.2 Average Numbers of Iterations Required by the Largest-Increase Rule

m \ n	10	20	30	40	50
10	7.02	9.17	10.8	12.1	12.6
20		16.2	20.2	24.2	27.3
30			28.7	34.5	39.4
40				43.3	39.9
50					58.9

Source: D. Avis and V. Chvátal (1978).

Nevertheless, as the largest-coefficient rule takes less time to execute than the largest increase, it is the former that usually wins in terms of total computing time. More generally, the number of iterations is a poor criterion for assessing the efficiency of a rule for choosing the entering variable. It is the total computing time that counts, and rules that tend to reduce the number of iterations often take too much time to execute. In this light, even the largest-coefficient rule is found too time-consuming and therefore rarely, if ever, used in practice. The choice of entering variables in efficient implementations of the simplex method is influenced by the logistics of handling large problems on a computer; this matter will be studied in Chapter 7.

A systematic rule that always leads to an unambiguous choice of the entering variable, and to an unambiguous choice of the leaving variable in case of a tie, is called a *pivoting rule*. The largest-coefficient rule and the largest-increase rule, amended by unambiguous instructions for tie-breaking, are two examples of pivoting rules; the smallest-subscript rule of Chapter 3 is another.

EFFICIENCY OF ALGORITHMS IN THEORY AND PRACTICE

As noted in Chapter 1, the theoretical and the practical criteria for judging the efficiency of algorithms are radically different. From the theoretical point of view, an algorithm is satisfactory if its running time increases only slowly with the size of the problem. This is a vague definition as it stands; we are going to make it precise.

Let us consider a fixed class of problems (such as linear programming problems in the standard form) and a fixed algorithm (such as the simplex method) for solving problems in this class. A fair interpretation of the "size of the problem" is the time required to transmit the data. To put it differently, the size of a problem is the number of times you have to hit the keyboard of your typewriter in order to write down the data. For instance, the size of the Klee–Minty problems (4.2) is roughly $n^3/3$ when n gets very large (each of the $i - j + 1$ digits in each coefficient $2 \cdot 10^{i-j}$ has to be written down). A fair interpretation of "running time" is the total number of elementary steps (such as adding up, multiplying, or comparing two one-digit numbers; executing a "go to" instruction in a computer program; and so on) that have to be executed. (Thus it is implicitly assumed that each elementary step requires one unit of time.) Now for each s, there may be many (but only finitely many) different problems of size s in our class, and our algorithm may require different amounts of time $t_1, t_2, \ldots, t_M$ for different problems $P_1, P_2, \ldots, P_M$ of this size. Only the largest of these numbers t_i matters in the theoretical context. Of course, this largest t_i depends on s; we shall denote it by $t(s)$. Thus, our algorithm solves every problem of size s within $t(s)$ units of time and actually uses up these $t(s)$ units of time in the worst case. Finally, the algorithm is considered satisfactory if $t(s)$ grows only slowly with s. More precisely, the algorithm is satisfactory if there is a polynomial p such that $t(s) \leq p(s)$ for all s.

This definition, proposed by J. Edmonds (1965), is one of the most fruitful and stimulating concepts in theoretical computer science. [Those wishing for more information on this subject are referred to Garey and Johnson (1979).] Nevertheless, even though this concept does reflect to some extent the reasons why practitioners are satisfied by some algorithms and unsatisfied by others, it fails to capture these reasons fully. Two of the features that make it unrealistic from a practical point of view are:

(i) The worst-case criterion.
(ii) The asymptotic point of view.

The inadequacy of the worst-case criterion is demonstrated most dramatically on the case of the simplex method itself: even eminently useful algorithms may be labeled unsatisfactory on the basis of a few isolated examples of a kind that might never come up in practice. The average running time ($\sum t_i/M$) might provide a more realistic criterion than the worst running time (max t_i); unfortunately, a rigorous analysis of the average performance is often much more difficult than an analysis of the worst performance. The inadequacy of the second feature may manifest itself even when the running time depends only on the size of the problem, so that the average performance and the worst performance coincide. The point is that the actual values $t(s)$, with s restricted to a finite range, do not matter at all; the only thing that counts is the rate of growth of $t(s)$ as s increases beyond every bound. Thus, a hypothetical algorithm with a running time $t = 10^{s/1{,}000{,}000{,}000}$ (rounded up to the nearest integer) would be found theoretically unsatisfactory even though $t(s) \leq 10$ whenever $s \leq 10^9$; on the other hand, an algorithm with a running time $t(s) = 10^{1{,}000}s$ would be found theoretically satisfactory even though $t(s) \geq 10^{1{,}000}$ for all s. Theorists judge algorithms by their worst performance on problems of sizes outside the range of practical interest, whereas practitioners judge algorithms by their typical performance on problems whose sizes are limited to a finite range. (In all fairness, it should be admitted that the theoretical definition is not all that bad. As it turns out, the polynomials bounding the running time of theoretically satisfactory algorithms often assume reasonably small values for reasonably small values of s and, on the other hand, even the typical running time of theoretically unsatisfactory algorithms will often get out of hand already for small values of s.)

For many years, while practitioners were trying to reduce the typical running time of the simplex method by yet another 10% or 20%, theorists were trying to answer a fundamental question: Is there a theoretically satisfactory algorithm for solving linear programming problems? Eventually, L. G. Khachian (1979) provided an affirmative answer by presenting such an algorithm. This "ellipsoid method" is surprisingly simple and elegant; we shall describe its details in the appendix. Will this beautiful gem of pure mathematics ever become a serious challenger of the simplex method's supremacy in solving practical LP problems? That remains to be seen; at the time of this writing, it seems very likely that the answer is no.

PROBLEMS

4.1 Compare the performance of the three pivoting rules discussed in this chapter on the following examples:

a. maximize $4x_1 + 5x_2$

subject to
$$\begin{aligned} 2x_1 + x_2 &\leq 9 \\ x_1 &\leq 4 \\ x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

b. maximize $2x_1 + x_2$

subject to
$$\begin{aligned} 3x_1 + x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

c. maximize $3x_1 + 5x_2$

subject to
$$\begin{aligned} x_1 + 2x_2 &\leq 5 \\ x_1 &\leq 3 \\ x_2 &\leq 2 \\ x_1, x_2 &\geq 0. \end{aligned}$$

4.2 In the Klee–Minty problem (4.2), denote the slack variables by $s_1, s_2, \ldots, s_n$ rather than by $x_{n+1}, x_{n+2}, \ldots, x_{2n}$. Prove that in every feasible dictionary, precisely one of the two variables x_i, s_i is basic.

4.3 Use the result of problem 4.2 and induction on n to prove that, when the simplex method with the largest coefficient rule is applied to (4.2), the resulting dictionaries have the following properties:

(i) After $2^{n-1} - 1$ iterations, the last row reads

$$z = 10\left(100^{n-2} - \sum_{j=1}^{n-2} 10^{n-1-j}x_j - s_{n-1}\right) + x_n.$$

(ii) After 2^{n-1} iterations, the last row reads

$$z = 90 \cdot 100^{n-2} + 10\left(\sum_{j=1}^{n-2} 10^{n-1-j}x_j + s_{n-1}\right) - s_n.$$

(iii) After $2^n - 1$ iterations, the last row reads

$$z = 100^{n-1} - \sum_{j=1}^{n-1} 10^{n-j}x_j - s_n.$$

(iv) After each iteration, all the coefficients in the last row are integers.

5

The Duality Theorem

Every maximization LP problem in the standard form gives rise to a minimization LP problem called the dual problem. The two problems are linked in an interesting way. Every feasible solution in one yields a bound on the optimal value of the other. In fact, if one of the two problems has an optimal solution, then so does the other, and the two optimal values coincide. This fact, known as the Duality Theorem, is the subject of the present chapter. We shall also note that, in managerial applications, the variables featured in the dual problem can be interpreted in a very useful way.

MOTIVATION: FINDING UPPER BOUNDS ON THE OPTIMAL VALUE

We shall begin this chapter with the following LP problem:

$$
\begin{array}{lrcrcrcrcr}
\text{maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{subject to} & x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
 & 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
 & -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
 & & & & & & x_1, x_2, & x_3, x_4 & \geq & 0.
\end{array}
$$

Rather than *solving* it, we shall try to get a quick *estimate* of the optimal value z^* of its objective function. To get a reasonably good lower bound on z^*, we need only come up with a reasonably good feasible solution. For example, the bound $z^* \geq 5$ comes from considering the feasible solution $(0, 0, 1, 0)$. The feasible solution $(2, 1, 1, \frac{1}{3})$ shows that $z^* \geq 15$. Better yet, the feasible solution $(3, 0, 2, 0)$ yields $z^* \geq 22$. Needless to say, such guesswork is vastly inferior to the systematic attack by the simplex method: even if we were lucky enough to hit on the optimal solution, our guess would provide no *proof* that the solution is indeed optimal.

We shall not pursue this line any further: the subject of this chapter stems from a similar quest for *upper* bounds on z^*. For example, a glance at the data suggests that $z^* \leq \frac{275}{3}$. Indeed, multiplying the second constraint by $\frac{5}{3}$ we obtain the inequality

$$\frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \leq \frac{275}{3}.$$

Hence every feasible solution (x_1, x_2, x_3, x_4) satisfies the inequality

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \leq \frac{275}{3}.$$

In particular, this inequality holds for the optimal solution and so $z^* \leq \frac{275}{3}$. With a little inspiration, we can improve this bound considerably. For instance, the sum of the second and third constraints reads

$$4x_1 + 3x_2 + 6x_3 + 3x_4 \leq 58.$$

Therefore, $z^* \leq 58$. Rather than searching for further improvements in a haphazard way, we shall now describe the strategy in precise and general terms.

We construct *linear combinations* of the constraints. That is, we multiply the first constraint by some number y_1, the second by y_2, the third by y_3, and then we add them up. (In the first case, we had $y_1 = 0$, $y_2 = \frac{5}{3}$, $y_3 = 0$; in the second case, we had $y_1 = 0$, $y_2 = y_3 = 1$.) The resulting inequality reads

$$\begin{aligned}(y_1 + 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3 + (3y_1 + 8y_2 - 5y_3)x_4 \\ \leq y_1 + 55y_2 + 3y_3.\end{aligned} \tag{5.1}$$

Of course, each of the three multipliers y_i must be nonnegative: otherwise the corresponding inequality would reverse its direction. Next, we want to use the left-hand side of (5.1) as an upper bound on $z = 4x_1 + x_2 + 5x_3 + 3x_4$. This can be justified only if in (5.1), the coefficient at each x_j is at least as big as the corresponding coefficient in z. More explicitly, we want

$$\begin{aligned} y_1 + 5y_2 - y_3 &\geq 4 \\ -y_1 + y_2 + 2y_3 &\geq 1 \\ -y_1 + 3y_2 + 3y_3 &\geq 5 \\ 3y_1 + 8y_2 - 5y_3 &\geq 3. \end{aligned}$$

If the multipliers y_i are nonnegative and if they satisfy these four inequalities, then we may safely conclude that every feasible solution (x_1, x_2, x_3, x_4) satisfies the inequality

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq y_1 + 55y_2 + 3y_3.$$

In particular, this inequality is satisfied by the optimal solution; therefore

$$z^* \leq y_1 + 55y_2 + 3y_3.$$

Of course, we want as small an upper bound on z^* as we can possibly get. Thus, we are led to the following LP problem:

$$\begin{array}{lrcl}
\text{minimize} & y_1 + 55y_2 + 3y_3 & & \\
\text{subject to} & y_1 + 5y_2 - y_3 & \geq & 4 \\
 & -y_1 + y_2 + 2y_3 & \geq & 1 \\
 & -y_1 + 3y_2 + 3y_3 & \geq & 5 \\
 & 3y_1 + 8y_2 - 5y_3 & \geq & 3 \\
 & y_1, y_2, y_3 & \geq & 0.
\end{array}$$

THE DUAL PROBLEM

This problem is called the *dual* of the original one; the original problem is called the *primal* problem. In general, the dual of the problem

$$\begin{array}{lll}
\text{maximize} & \sum_{j=1}^{n} c_j x_j & \\
\text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & (i = 1, 2, \ldots, m) \\
 & x_j \geq 0 & (j = 1, 2, \ldots, n)
\end{array} \tag{5.2}$$

is defined to be the problem

$$\begin{array}{lll}
\text{minimize} & \sum_{i=1}^{m} b_i y_i & \\
\text{subject to} & \sum_{i=1}^{m} a_{ij} y_i \geq c_j & (j = 1, 2, \ldots, n) \\
 & y_i \geq 0 & (i = 1, 2, \ldots, m).
\end{array} \tag{5.3}$$

(Note that the dual of a maximization problem is a minimization problem. Furthermore, the m primal constraints $\sum a_{ij}x_j \leq b_i$ are in a one-to-one correspondence with the m dual variables y_i; conversely, the n dual constraints $\sum a_{ij}y_i \geq c_j$ are in a one-to-one correspondence with the n primal variables x_j. The coefficient at each variable in the objective function, primal or dual, appears in the other problem as the right-hand side of the corresponding constraint.)

As in our example, every feasible solution of the dual yields an upper bound on the optimal value of the primal. More explicitly, for every primal feasible solution $(x_1, x_2, \ldots, x_n)$ and for every dual feasible solution $(y_1, y_2, \ldots, y_m)$ we have

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i. \tag{5.4}$$

The proof of (5.4), which was illustrated at the beginning of this section, can be written down succinctly as

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \sum_{i=1}^{m} b_i y_i.$$

Inequality (5.4) is extremely useful: if we happen to stumble across a primal feasible solution $(x_1^*, x_2^*, \ldots, x_n^*)$ and a dual feasible solution $(y_1^*, y_2^*, \ldots, y_m^*)$ such that

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*$$

then we may conclude that both of these solutions are optimal. Indeed, (5.4) implies that every primal feasible solution $(x_1, x_2, \ldots, x_n)$ satisfies

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i^* = \sum_{j=1}^{n} c_j x_j^*$$

and that every dual feasible solution $(y_1, y_2, \ldots, y_m)$ satisfies

$$\sum_{i=1}^{m} b_i y_i \geq \sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*.$$

For instance, we have an easy way of showing that the primal feasible solution $x_1 = 0$, $x_2 = 14$, $x_3 = 0$, $x_4 = 5$ of our original example is optimal: just consider the dual feasible solution $y_1 = 11$, $y_2 = 0$, $y_3 = 6$. It is not at all obvious, however, that an analogous proof of optimality can be given for *every* LP problem that has an optimal solution; this fact is the central theorem of linear programming.

THE DUALITY THEOREM AND ITS PROOF

The explicit version of the theorem comes from D. Gale, H. W. Kuhn, and A. W. Tucker (1951); its notions originated in conversations between G. B. Dantzig and J. von Neumann in the fall of 1947.

THEOREM 5.1 (The Duality Theorem). If the primal (5.2) has an optimal solution $(x_1^*, x_2^*, \ldots, x_n^*)$, then the dual (5.3) has an optimal solution $(y_1^*, y_2^*, \ldots, y_m^*)$ such that

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*. \tag{5.5}$$

Before presenting the proof, let us briefly illustrate its crucial point: the optimal solution of the *dual* problem can be read off the z-row of the final dictionary for the *primal* problem. In the example that we used to motivate the concept of the dual problem, the final dictionary reads

$$\begin{array}{rcl} x_2 &=& 14 - 2x_1 - 4x_3 - 5x_5 - 3x_7 \\ x_4 &=& 5 - x_1 - x_3 - 2x_5 - x_7 \\ x_6 &=& 1 + 5x_1 + 9x_3 + 21x_5 + 11x_7 \\ \hline z &=& 29 - x_1 - 2x_3 - 11x_5 - 6x_7. \end{array}$$

Note that the slack variables x_5, x_6, x_7 can be matched up with the dual variables y_1, y_2, y_3 in a natural way: for instance, x_5 is the slack variable in the first constraint, whereas y_1 represents the multiplier for the same constraint. By the same logic, x_6 goes with y_2 and x_7 goes with y_3. In the *z-row* of the dictionary, the coefficients at the slack variables are

$$-11 \text{ at } x_5, \quad 0 \text{ at } x_6, \quad -6 \text{ at } x_7.$$

Assigning these values with reversed signs to the corresponding dual variables, we obtain the desired optimal solution of the dual:

$$y_1 = 11, \quad y_2 = 0, \quad y_3 = 6.$$

At first, this may seem like pulling a rabbit out of a hat; however, the following general argument explains the magic.

PROOF OF THEOREM 5.1. We need only find a *feasible* solution $(y_1^*, y_2^*, \ldots, y_m^*)$ satisfying (5.5); indeed, such a solution will be *optimal* by virtue of the remarks following (5.4). In order to find that solution, we solve the primal problem by the simplex method; having introduced the slack variables

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad (i = 1, 2, \ldots, m) \tag{5.6}$$

we eventually arrive at the final dictionary. For the sake of definiteness, let us say that the last row of that dictionary reads

$$z = z^* + \sum_{k=1}^{n+m} \bar{c}_k x_k. \tag{5.7}$$

In (5.7), each $\bar{c}_k$ is a nonpositive number (in fact, $\bar{c}_k = 0$ whenever x_k is a basic variable). In addition, z^* is the optimal value of the objective function, and so

$$z^* = \sum_{j=1}^{n} c_j x_j^*. \tag{5.8}$$

Defining

$$y_i^* = -\bar{c}_{n+i} \qquad (i = 1, 2, \ldots, m) \tag{5.9}$$

we claim that $(y_1^*, y_2^*, \ldots, y_m^*)$ is a dual feasible solution satisfying (5.5); the rest of the proof consists of a straightforward verification of our claim. Substituting $\sum c_j x_j$ for z and substituting from (5.6) for the slack variables in (5.7) we obtain the identity

$$\sum_{j=1}^{n} c_j x_j = z^* + \sum_{j=1}^{n} \bar{c}_j x_j - \sum_{i=1}^{m} y_i^* \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right)$$

which may be written as

$$\sum_{j=1}^{n} c_j x_j = \left(z^* - \sum_{i=1}^{m} b_i y_i^* \right) + \sum_{j=1}^{n} \left(\bar{c}_j + \sum_{i=1}^{m} a_{ij} y_i^* \right) x_j.$$

This identity, having been obtained by algebraic manipulations from the definitions of the slack variables and the objective function, must hold for every choice of values of $x_1, x_2, \ldots, x_n$. Hence we have

$$z^* = \sum_{i=1}^{m} b_i y_i^* \tag{5.10}$$

and

$$c_j = \bar{c}_j + \sum_{i=1}^{m} a_{ij} y_i^* \qquad (j = 1, 2, \ldots, n). \tag{5.11}$$

Since $\bar{c}_k \leq 0$ for every $k = 1, 2, \ldots, n + m$, (5.11) and (5.9) imply

$$\sum_{i=1}^{m} a_{ij} y_i^* \geq c_j \qquad (j = 1, 2, \ldots, n)$$

$$y_i^* \geq 0 \qquad (i = 1, 2, \ldots, m).$$

Finally, (5.10) and (5.8) imply (5.5). ∎

RELATIONSHIP BETWEEN THE PRIMAL AND DUAL PROBLEMS

Next, let us point out that the dual of the dual is always the primal problem. Indeed, the dual problem may be written as

$$\begin{array}{lll} \text{maximize} & \sum_{i=1}^{m} (-b_i)y_i & \\ \text{subject to} & \sum_{i=1}^{m} (-a_{ij})y_i \le -c_j & (j = 1, 2, \ldots, n) \\ & y_i \ge 0 & (i = 1, 2, \ldots, m). \end{array}$$

The dual of this problem is

$$\begin{array}{lll} \text{minimize} & \sum_{j=1}^{n} (-c_j)x_j & \\ \text{subject to} & \sum_{j=1}^{n} (-a_{ij})x_j \ge -b_i & (i = 1, 2, \ldots, m) \\ & x_j \ge 0 & (j = 1, 2, \ldots, n) \end{array}$$

which is clearly equivalent to the original problem. A nice corollary to this observation and to the duality theorem is that the primal problem has an optimal solution *if and only if* the dual problem has an optimal solution. Note also that if the primal is unbounded, then the dual must be infeasible [this follows directly from (5.4)]. By the same argument, if the dual is unbounded then the primal must be infeasible. However, both primal and dual may be infeasible at the same time. For example, both the problem

$$\begin{array}{ll} \text{maximize} & 2x_1 - x_2 \\ \text{subject to} & x_1 - x_2 \le 1 \\ & -x_1 + x_2 \le -2 \\ & x_1, x_2 \ge 0 \end{array}$$

and its dual are infeasible. These conclusions are summarized in Table 5.1.

Table 5.1
Primal–Dual Combinations

		Dual		
		Optimal	Infeasible	Unbounded
Primal	Optimal	Possible	Impossible	Impossible
	Infeasible	Impossible	Possible	Possible
	Unbounded	Impossible	Possible	Impossible

In particular, if the primal problem has a feasible solution *and* if the dual problem has a feasible solution, then both problems have optimal solutions.

Duality has important practical implications. In certain cases, we may find it advantageous to apply the simplex method to the dual of the problem that we are really interested in. (Of course, the optimal solution of the primal problem can then be read directly off the final dictionary for the dual.) For example, if $m = 99$ and $n = 9$, then dictionaries will have 100 rows in the primal problem but only 10 rows in the dual. Since the typical number of simplex iterations is proportional to the number of rows in a dictionary and relatively insensitive to the number of variables, we shall most likely be better off solving the dual problem.

From a theoretical point of view, duality is important because it points out an elegant and succinct way of proving optimality of solutions of LP problems: as we have observed, an optimal solution of the dual problem provides a "certificate of optimality" for an optimal solution of the primal problem, and vice versa. Furthermore, the duality theorem asserts that for *every* optimal solution there is a certificate of optimality. To appreciate the impact of this fact, consider a student who is supposed to solve the problem

$$\begin{aligned} &\text{maximize} && \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad (i = 1, 2, \ldots, m) \\ &&& x_j \ge 0 \qquad (j = 1, 2, \ldots, n). \end{aligned} \tag{5.12}$$

Applying the simplex method to (5.12), the student finds simultaneously an optimal solution $x_1^*, x_2^*, \ldots, x_n^*$ of (5.12) and an optimal solution $y_1^*, y_2^*, \ldots, y_m^*$ of the dual problem

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{m} b_i y_i \\ &\text{subject to} && \sum_{i=1}^{m} a_{ij} y_i \ge c_j \qquad (j = 1, 2, \ldots, n) \\ &&& y_i \ge 0 \qquad (i = 1, 2, \ldots, m). \end{aligned} \tag{5.13}$$

Then he shows *both* solutions to his supervisor. The supervisor has an easy way of checking the correctness of the answer. To check the *feasibility* of the allegedly optimal solution, she has to verify the inequalities

$$\begin{aligned} &\sum_{j=1}^{n} a_{ij} x_j^* \le b_i \qquad (i = 1, 2, \ldots, m) \\ &x_j^* \ge 0 \qquad (j = 1, 2, \ldots, n). \end{aligned} \tag{5.14}$$

To check its *optimality*, she has to verify the inequalities

$$\begin{aligned} \sum_{i=1}^{m} a_{ij} y_i^* &\geq c_j \qquad (j = 1, 2, \ldots, n) \\ y_i^* &\geq 0 \qquad (i = 1, 2, \ldots, m) \end{aligned} \tag{5.15}$$

and the equation

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*. \tag{5.16}$$

Of course, the computational effort involved in these *verifications* is much smaller than the computational effort required to *solve* (5.12) from scratch by the simplex method.

COMPLEMENTARY SLACKNESS

Now we shall show how the supervisor can often recover the certificate of optimality $y_1^*, y_2^*, \ldots, y_m^*$ from the optimal solution $x_1^*, x_2^*, \ldots, x_n^*$ alone. The key to the procedure is a convenient way of breaking down equation (5.16) into simple constituents.

THEOREM 5.2. Let $x_1^*, x_2^*, \ldots, x_n^*$ be a feasible solution of (5.12) and let $y_1^*, y_2^*, \ldots, y_m^*$ be a feasible solution of (5.13). Necessary and sufficient conditions for simultaneous optimality of $x_1^*, x_2^*, \ldots, x_n^*$ and $y_1^*, y_2^*, \ldots, y_m^*$ are

$$\sum_{i=1}^{m} a_{ij} y_i^* = c_j \quad \text{or} \quad x_j^* = 0 \quad \text{(or both)} \quad \text{for every} \quad j = 1, 2, \ldots, n \tag{5.17}$$

and

$$\sum_{j=1}^{n} a_{ij} x_j^* = b_i \quad \text{or} \quad y_i^* = 0 \quad \text{(or both)} \quad \text{for every} \quad i = 1, 2, \ldots, m. \tag{5.18}$$

PROOF. Assumptions (5.14) and (5.15) imply

$$c_j x_j^* \leq \left(\sum_{i=1}^{m} a_{ij} y_i^* \right) x_j^* \qquad (j = 1, 2, \ldots, n) \tag{5.19}$$

$$\left(\sum_{j=1}^{n} a_{ij} x_j^* \right) y_i^* \leq b_i y_i^* \qquad (i = 1, 2, \ldots, m) \tag{5.20}$$

and so

$$\sum_{j=1}^{n} c_j x_j^* \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i^* \right) x_j^* = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j^* \right) y_i^* \leq \sum_{i=1}^{m} b_i y_i^*. \tag{5.21}$$