Chapter 2

The Simplex Algorithm

In Chapter 1, we used a graphical method for solving l.p. problems with just two decision variables. Although, in simple cases, this method can be extended to solving problems with three decision variables, it will be clear to you that it is of very limited use in practice, since most real life problems have many decision variables and functional constraints. In this chapter, we introduce an iterative algebraic method, called the **simplex algorithm**, which was first proposed by G.B. Dantzig in 1947. Modern computer implementations of this algorithm are now routinely used for solving problems with literally thousands of variables and functional constraints.

The graphical solution of an l.p. problem with two or three variables is very helpful in understanding the workings of the simplex algorithm however, and we shall use the graphical solution of the Boatbuilder's problem (Example 1.3) to illustrate how the algorithm works.

2.1 Formulating l.p. problems using vectors and matrices

As you will have seen from the exercises and examples in Chapter 1, the objective of an l.p. problem may be either to maximize or to minimize the objective function and constraints may be expressed using either "$\leq$" or "$\geq$" signs. Some problems have constraints of both types, and equality constraints may also occur. In order to describe the simplex algorithm, it is therefore useful to define a *standard* form of an l.p. problem with n decision variables and m functional constraints, for any positive integers m and n. We shall show later that any general l.p. problem may be reformulated as a standard l.p. problem.

Definition 2.1 *A* **standard linear programming problem** *is an optimisation problem in the following form.*

Find $x_1, x_2, ..., x_n$ *to maximise* $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, *subject to*

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \leq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{1n}x_n & \leq & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \leq & b_m \end{array}$$

$$x_1 \geq 0, x_2 \geq 0, ..., x_n \geq 0,$$

where c_j, a_{ij} *and* b_i *are given real numbers, for* $1 \leq i \leq m$, $1 \leq j \leq n$.

Thus, the formulation of the Boatyard problem in Example 1.1 is an example of a *standard* linear programming problem, whereas the formulation of the Cattle Feed problem in Example 1.2 is a *general* linear programming problem but is not a standard linear programming problem.

In order to represent an l.p. problem more succinctly, it is useful to use vector and matrix notation. Let $\mathbf{R}^n$ denote the set of all n-dimensional column vectors (that is, column vectors with n entries), where the entries are any real numbers. In Chapter 1, we represented the solution of l.p. problems with two variables by a pair of coordinates (x_1, x_2). Using a vector in $\mathbf{R}^n$, we can represent a solution of an l.p. problem with n decision variables, for *any* finite positive number n.

In order to express the standard linear programming problem of Definition 2.1 using vector and matrix notation, we need to define what is meant by an *inequality* between two vectors.

Definition 2.2 *Let* $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ *and* $\mathbf{y} = [y_1, y_2, ..., y_n]^T$. *We say that* $\mathbf{x} \leq \mathbf{y}$ *if* $x_i \leq y_i$, *for ALL* i, $1 \leq i \leq n$. *Similarly,* $\mathbf{x} \geq \mathbf{y}$ *if* $x_i \geq y_i$, *for ALL* i, $1 \leq i \leq n$. *If neither of these alternatives hold, then* $\mathbf{x}$ *and* $\mathbf{y}$ *are said to be* **incomparable.**

Definition 2.3 *The n-dimensional vector* $[0, 0, ..., 0]^T$ *is called the* **zero vector** *and is represented by the symbol* $\mathbf{0}$, *printed in bold face type.*

Example 2.1 Let

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 3 \\ 0 \\ 5 \end{bmatrix}.$$

Then we have $\mathbf{u} \geq \mathbf{0}$, $\mathbf{v} \geq \mathbf{0}$ and $\mathbf{w} \geq \mathbf{0}$; also, $\mathbf{u} \geq \mathbf{v}$, $\mathbf{v} \leq \mathbf{w}$, but $\mathbf{u}$ and $\mathbf{w}$ are incomparable. □

We can now express the standard l.p. problem of Definition 2.1 more concisely as follows:

> Find $\mathbf{x} \in \mathbf{R}^n$ to maximise $z = \mathbf{c}^T\mathbf{x}$, subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$,
> where $\mathbf{A}$ is a given $m \times n$ real matrix, and $\mathbf{c} \in \mathbf{R}^n$, $\mathbf{b} \in \mathbf{R}^m$ are given vectors.

We can also restate in vector and matrix notation the definitions of terms introduced in Chapter 1.

Definition 2.4 *A vector* $\mathbf{x} \in \mathbf{R}^n$ *is a* **feasible solution** *of an l.p. problem with n decision variables if* $\mathbf{x}$ *satisfies ALL the contraints. The* **optimum** *value of an l.p. problem is the maximum (or minimum) value achieved by the objective function* $z = \mathbf{c}^T\mathbf{x}$, *taken over all feasible solutions* $\mathbf{x}$. *An* **optimal** *solution is any feasible solution at which z achieves its optimum value.*

We also defined in Chapter 1 the **feasible region** of an l.p. problem as the set of all feasible solutions. Thus the **feasible region** for the standard l.p. problem of Definition 2.1 is the set

$$S = \{\mathbf{x} \in \mathbf{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Example 2.2 Consider the Boatbuilder's problem of Example 1.1. We want to find $x_1, x_2 \in \mathbf{R}$ to maximize $z = 6x_1 + 5x_2$, subject to

$$\begin{aligned} 24x_1 + 6x_2 &\leq 2400 && (2.1) \\ 6x_1 + 4x_2 &\leq 720 && (2.2) \\ 2x_1 + 8x_2 &\leq 1000 && (2.3) \\ x_1, x_2 &\geq 0. && (2.4) \end{aligned}$$

We can reformulate this in vector notation as follows.

Find $\mathbf{x} \in \mathbf{R}^2$ to maximise $z = \mathbf{c}^T\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$,

where

$$\mathbf{c} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 2400 \\ 720 \\ 1000 \end{bmatrix} \text{ and } \mathbf{A} = \begin{bmatrix} 24 & 6 \\ 6 & 4 \\ 2 & 8 \end{bmatrix}. \square$$

2.2 Corner-point solutions of an l.p. problem

In the graphical solution of the examples and exercises in Chapter 1, you will have observed that when these l.p. problems had an optimal solution, then at least one such solution *always* occurred at a *corner-point* of the feasible region. In this section, we investigate the geometry of the feasible region for l.p. problems in 2-dimensions to establish some special properties of the corner-point solutions that can be extended to higher dimensions.

Consider the feasible region for the Boatbuilder's problem shown in Figure 1.2. This figure contains five lines, each corresponding to one of the constraints. We shall call a point of intersection of any two constraint lines a **corner-point**. There are altogether 10 corner-points in the solution for the Boatbuilder's problem, since each pair of the five constraint lines intersects in a distinct point. The coordinates (x_1, x_2) of a corner-point give a solution to the l.p. problem. This is called a **feasible corner-point solution** when the corner-point is on the boundary of (and therefore belongs to) the feasible region and an **infeasible corner-point solution** when the corner-point is outside the feasible region.

Example 2.3 For the Boatbuilder's problem, the *feasible* corner-point solutions are:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 \\ 0 \end{bmatrix}, \begin{bmatrix} 88 \\ 48 \end{bmatrix}, \begin{bmatrix} 44 \\ 114 \end{bmatrix}, \begin{bmatrix} 0 \\ 125 \end{bmatrix}.$$

The corner-point determined by the intersection of the constraint lines $x_2 = 0$ and $6x_1 + 4x_2 = 720$ gives the following example of an *infeasible* corner-point solution.

$$\begin{bmatrix} 120 \\ 0 \end{bmatrix}. \square$$

In 2-dimensions, the boundary of the feasible region is made up of a sequence of line segments, each one part of a constraint line. Such a boundary is called **polygonal**. If, when we trace the boundary starting from any point and travelling clockwise we return to the starting point, then the feasible region is said to be a **closed polygon**. The feasible region for the Boatbuilder's problem is a closed polygon, while the feasible region for the Cattle Feed problem (see Exercise 1.4) has an open polygonal boundary. Two corner-points of the feasible region are said to be **adjacent** if they are endpoints of the same line segment on the boundary.

In 3-dimensions, a very similar situation holds. There is a *plane* corresponding to each constraint. This plane divides 3-dimensional space into two regions, such that the points of the plane and those lying on one side of it satisfy the constraint, while the points lying on the other side of the plane do not satisfy the constraint. In particular, the equations $x_i = 0$, $i = 1, 2, 3$, each give one of the coordinate planes, so that the non-negativity constraints $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$ define the positive quadrant of the 3-dimensional coordinate diagram. Thus, when an l.p. problem with three decision variables has feasible solutions, the boundary of the feasible region is composed of plane surfaces.

Some sets of three or more of the constraint planes may intersect in a unique point. As in the 2-dimensional case, we call such a point of intersection a **corner-point**, and its coordinates give a **corner-point solution**. Also, two corner-points are called **adjacent** if they are the endpoints of a line of intersection of two of the constraint planes.

Example 2.4 Consider the following l.p. problem.

Find $x_1, x_2, x_3 \in \mathbf{R}$ to maximize $z = 3x_1 + 2x_2 + 5x_3$ subject to

$$\begin{aligned} x_1 + 2x_2 + 10x_3 &\leq 60 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

The feasible region is bounded by the four planes $x_1 + 2x_2 + 10x_3 = 60$, $x_1 = 0$, $x_2 = 0$, $x_3 = 0$. Testing a trial point, we see that the point $(0,0,0)$ satisfies all the constraints so that the feasible region is the set of points on the surface and in the interior of the tetrahedron bounded by these four planes. There are just four corner-points, found as follows:

Planes	*Corner-point*
$x_1 = 0$, $x_2 = 0$, $x_3 = 0$	$(0,0,0)$
$x_1 = 0$, $x_2 = 0$, $x_1 + 2x_2 + 10x_3 = 60$	$(0,0,6)$
$x_1 = 0$, $x_3 = 0$, $x_1 + 2x_2 + 10x_3 = 60$	$(0,30,0)$
$x_2 = 0$, $x_3 = 0$, $x_1 + 2x_2 + 10x_3 = 60$	$(60,0,0)$.

In this case, all the corner-point solutions are feasible. Also, it happens that each pair of corner-points is adjacent. □

The feasible region of every l.p. problem has an important property. Mark any two points of the feasible region for the Boatbuilder's problem (or for any of the other l.p. problems you solved graphically in Exercises 1.4). You will see that the line joining these points *lies entirely in the feasible region.* This is also a property of the feasible region for the l.p. problem in Example 2.4. A region of the plane or of 3-dimensional space with this property is said to be **convex**.

Example 2.5 All triangles, rectangles and circles are examples of convex figures in 2-dimensions, while all tetrahedrons, cuboids and spheres are examples of convex solids. For comparison, Figure 2.1 shows an example of a quadrialeral and a pentagon that are *not* convex. □

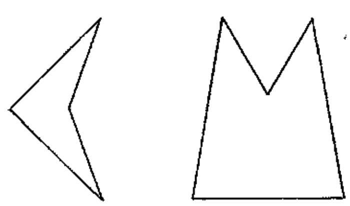

Figure 2.1.

We state without proof that if an l.p. problem with two or three decision variables has a non-empty feasible region, then this region is always convex. In particular, this means that when we draw a line joining two *non-adjacent* corner-points of the feasible region, then the interior points of the line are all interior points of the feasible region. This has the important consequences summarized in the following result.

Result 2.5 *Suppose that an l.p. problem has a finite optimal solution. Then*

(a) *one optimal solution* always *occurs at a corner-point of the feasible region;*

(b) *if there is* more than one *optimal solution, then two such solutions occur at* adjacent *corner-points of the feasible region;*

(c) *if we can find a feasible corner-point solution such that no adjacent feasible corner-point solution gives a* better *value of z, than this corner-point solution is optimal.*

Not every l.p. problem has a finite optimal solution (see, for example, Exercise 1.5). In particular, since an optimal solution must be feasible, no optimal solution will exist if the feasible region is empty. If the problem has at least one feasible solution, then it has a finite maximum value if z is bounded above (or a finite minimum value if z is bounded below). In this case, a finite optimal solution exists and we can use Result 2.5 to locate it.

Example 2.6 We use this method to solve the Boatbuilder's problem with the objective functions of Example 1.3 and Example 1.4, by tabulating their values for each feasible corner-point solution as shown below.

Feasible corner-point solutions	$[0,0]^T$	$[100,0]^T$	$[88,48]^T$	$[44,114]^T$	$[0,125]^T$
$z = 6x_1 + 5x_2$	0	600	768	834	625
$z = 6x_1 + 4x_2$	0	600	720	720	500

Since the value of z is bounded above by the constraints, a finite optimal solution exists. When $z = 6x_1 + 5x_2$, Result 2.5 tells us that the optimum value of z is 834 and the *unique* optimal solution is $x_1 = 44$, $x_2 = 114$.

When $z = 6x_1 + 4x_2$, Result 2.5 tells us that the optimum value of z is 720 and optimal solutions occur at the adjacent corner-points (88, 48) and (44, 114). From Example 1.4, we know this implies that every point of the line segment joining these two corner-points is also an optimal solution.

2.3 Introduction to the simplex algorithm

Our solution of Example 2.6 still depended on our ability to graph the feasible region. In this section, we shall show how to move to a completely algebraic solution for problems in standard form with $\mathbf{b} \geq 0$. In the next chapter, we shall show how the method can be extended to solve general l.p. problems.

Our first step is to turn the inequalities representing the functional constraints into *equations*. In each constraint, we add a new variable called a *slack* variable to the *left* side of the inequality and replace the "$\leq$" by an "$=$" sign. Thus for any particular solution $\mathbf{x} \in \mathbf{R}^n$, the slack variable in constraint i represents the difference between the value of the right side of the constraint (the total amount of resource i available) and the value of the left side (the amount of resource i used at that solution). Thus a solution satisfies the constraint if and only if it makes the slack variable non-negative.

Example 2.7 The first constraint in the Boatbuilder's problem (Example 2.2) expresses the fact that the total number of kilograms of aluminium used in making x_1 dinghies and x_2 canoes must not exceed 2400. Adding a slack variable x_3 to the left side of (2.1) in Example 2.2 gives

$$24x_1 + 6x_2 + x_3 = 2400.$$

Thus $x_3 = 2400 - (24x_1 + 6x_2)$ will be non-negative for any solution that uses *at most* $2400kg$ of aluminium and will be negative for any solution that uses too much aluminium. Thus this constraint is satisfied if and only if we have a solution that makes $x_3 \geq 0$. $\square$

Adding slack variables to each of the functional constraints in Example 2.2, we obtain

$$\begin{array}{rcrcrcrcrcl} 24x_1 & + & 6x_2 & + & x_3 & & & & & = & 2400 \\ 6x_1 & + & 4x_2 & & & + & x_4 & & & = & 720 \\ 2x_1 & + & 8x_2 & & & & & + & x_5 & = & 1000 \end{array} \tag{2.5}$$

However, the set of solutions to this system of equations that give *feasible* solutions to the original problem are just those with $x_3 \geq 0$, $x_4 \geq 0$ and $x_5 \geq 0$. Thus we must add these three new constraints to the system, so that the non-negativity constraint (2.4) becomes

$$x_1, x_2, x_3, x_4, x_5 \geq 0. \tag{2.6}$$

We stress that this system is equivalent to the model in Example 2.2. The reason for changing the algebraic form of the model is that equations are simpler to handle than inequalities. We call the new model the **augmented form** of the problem.

Example 2.8 The matrix represention of the augmented form of the Boatbuilder's problem is

Find $\mathbf{x} \in \mathbf{R}^5$ to maximise $z = \mathbf{c}^T\mathbf{x}$ subject to $[\mathbf{A} : \mathbf{I}]\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$, where

$$\mathbf{c} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 2400 \\ 720 \\ 1000 \end{bmatrix} \text{ and } [\mathbf{A} : \mathbf{I}] = \begin{bmatrix} 24 & 6 & 1 & 0 & 0 \\ 6 & 4 & 0 & 1 & 0 \\ 2 & 8 & 0 & 0 & 1 \end{bmatrix}. \ \square$$

Notice that the number of variables has increased by the number of functional constraints, from two to five. Any solution $[x_1, x_2, x_3, x_4, x_5]^T$ of the augmented form is called an **augmented solution** of this l.p. problem.

Example 2.9 Substituting $x_1 = 50$ and $x_2 = 60$ in the augmented form of the Boatbuilder's problem yields $x_3 = 840$, $x_4 = 180$ and $x_5 = 420$. Thus the augmented solution corresponding to the solution $[50, 60]^T$ is $[50, 60, 840, 180, 420]^T$. Since the values of x_3, x_4 and x_5 are all *non-negative*, this solution is *feasible*. □

The augmented form of the standard l.p. problem defined in Definition 2.1 is as follows:

Find $\mathbf{x} \in \mathbf{R}^{n+m}$ to maximise $z = \mathbf{c}^T\mathbf{x}$, subject to $[\mathbf{A} : \mathbf{I}]\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$,
where $\mathbf{A}$ is a given $m \times n$ real matrix, and $\mathbf{c} \in \mathbf{R}^n$, $\mathbf{b} \in \mathbf{R}^m$ are given vectors.

The constraint lines corresponding to the functional constraints in the graphical solution of the Boatbuilder's problem have a simple algebraic interpretation. They each correspond to one of the equations

$$x_3 = 0, \ x_4 = 0, \ x_5 = 0.$$

Thus the ten corner-points in Figure 1.2 each correspond to an augmented solution of the problem in which two of the variables are put equal to zero.

Example 2.10 The constraint line $6x_1 + 4x_2 = 720$ is given by the equation $x_4 = 0$. This constraint line intersects the constraint line $x_1 = 0$ in the corner-point $(0, 180)$. The augmented solution corresponding to this corner-point is then $[0, 180, 1320, 0, -440]^T$. Note that this augmented solution shows that the corresponding corner-point solution is *infeasible* because x_5 is negative. We can see from Figure 1.2 that this corner-point is indeed outside the feasible region.

The constraint line $24x_1 + 6x_2 = 2400$ corresponds to $x_3 = 0$. The corner-point at which this line intersects the constraint line $x_4 = 0$ is $(88, 48)$. The augmented solution corresponding to this corner-point is $[88, 48, 0, 0, 440]^T$. Thus the corresponding corner-point solution is feasible. It corresponds to the point labelled B in Figure 1.2. □

It happens in the Boatbuilder's problem that every pair of constraint lines intersect in a distinct point, so that in Figure 1.2 we have ten corner-points altogether. Thus, putting any two of the five variables in the augmented form equal to zero and solving the three constraint equations for the remaining three variables gives an augmented corner-point solution. Extending this idea to the general case where we have n decision variables and m functional constraints, we obtain the following algebraic interpretation of the constraint lines and the corner-points.

Definition 2.6 *The augmented form of the standard problem has m equations in $m+n$ variables. Thus if we select n of these variables and put them equal to zero, we are left with a system of m linear equations in m variables. If this system has a* unique *solution, we call the corresponding augmented solution a* **basic solution** *to the l.p problem. If it is also feasible, it is called a* **basic feasible solution**. *The m variables that we solve for to obtain a basic solution are called the* **basic** *variables, and the variables that are put equal to zero are called* **non-basic**. *The set of basic variables is called the* **basis** *for the solution.*

Example 2.11 Consider again the five corner-points, O,A,B,C,D, of the feasible region for the Boatbuilder's problem. The corner-point O lies at the intersection of the constraint lines $x_1 = 0$ and $x_2 = 0$. Thus to find the basic feasible solution at O, we put $x_1 = x_2 = 0$ and solve the following three equations for x_3, x_4 and x_5.

$$\begin{array}{rcrcrcrcrcr}
24x_1 & + & 6x_2 & + & x_3 & & & & & = & 2400 \\
6x_1 & + & 4x_2 & & & + & x_4 & & & = & 720 \\
2x_1 & + & 8x_2 & & & & & + & x_5 & = & 1000
\end{array}$$

This gives the basic feasible solution $[0, 0, 2400, 720, 1000]^T$. At this solution, the variables x_3, x_4, x_5 are *basic*, and the variables x_1, x_2 are *non-basic*. □

Now consider how the basis changes as we move round the boundary of the feasible region for the Boatbuilder's problem, shown in Figure 2.2.

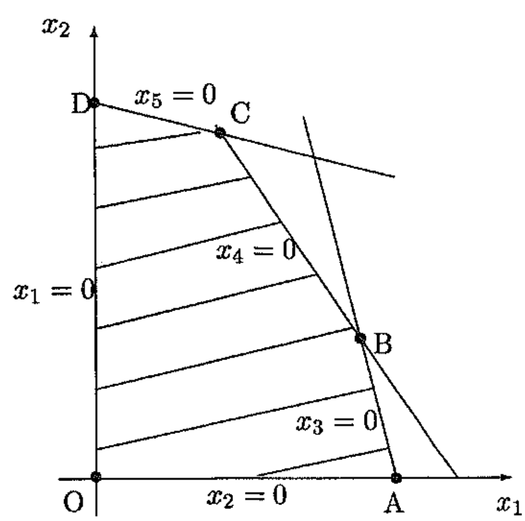

Figure 2.2.

We tabulate below the non-basic variables that determine each corner-point of the feasible region and the basis for the corresponding basic feasible solution.

Corner-point	*Non-basic variables*	*Basis*
O	x_1, x_2	x_3, x_4, x_5
A	x_2, x_3	x_1, x_4, x_5
B	x_3, x_4	x_1, x_2, x_5
C	x_4, x_5	x_1, x_2, x_3
D	x_1, x_5	x_4, x_2, x_3

You will see that as we move from any corner-point of the feasible region to an *adjacent* corner-point, the basis changes by *just one* variable. Thus as we move from point O to point A, for example, the variable x_3 leaves the basis and is replaced by the variable x_1; as we move from A to B, x_4 leaves the basis and is replaced by x_2, and so on. On the other hand, the bases at two *non-adjacent* corner-points of the feasible region differ in *at least two* variables. These considerations motivate the following definition.

Definition 2.7 *Two basic feasible solutions are said to be* **adjacent** *if they differ in exactly one variable.*

Suppose that we are given a standard l.p. problem with n decision variables, m functional constraints and a bounded objective function. In theory, we could find the optimal solution by determining all the corner-points of the feasible region, testing to see which give feasible corner-point solutions and then evaluating the objective function at each of them, as in Example 2.6. However, the number of basic solutions of the system $[\mathbf{A} : \mathbf{I}]\mathbf{x} = \mathbf{b}$, where $\mathbf{x} \in \mathbf{R}^{n+m}$, increases exponentially with $min\{m, n\}$. For example, for a problem with $m = n = 10$, we would need to consider $C(m+n, n) = C(20, 10) = 184,756$ different sets of 10 equations in 10 variables to find all the corner-points. Thus, when m and n are both large, this method is impractical.

Instead we use the simplex algorithm. It finds an optimum solution by calculating what is normally only a small proportion of the basic feasible solutions. Before considering the algebraic details of the method, we give an outline of its structure:

1. *Initialization step* Find a basic feasible solution.
2. *Optimality test* We ask whether we can get a *better* value of z by moving to an adjacent basic feasible solution; if not, the current solution is optimal and we stop.
3. *Iterative step* Move to an adjacent basic feasible solution that gives a *better* value of z. Return to Step 2.

Note that since there are at most $C(m+n, n)$ basic solutions, this process is guaranteed to terminate after a finite number of steps.

2.4 The steps in the simplex algorithm

In this section we describe the steps in the simplex algorithm and illustrate the method by using it to solve the Boatyard problem, starting from the augmented form of the problem. Thus, we are required to find $x_1, x_2, x_3, x_4, x_5 \in \mathbf{R}$ to maximize $z = 6x_1 + 5x_2$, subject to

$$\begin{array}{rcrcrcrcrcr} 24x_1 & + & 6x_2 & + & x_3 & & & & & = & 2400 \\ 6x_1 & + & 4x_2 & & & + & x_4 & & & = & 720 \\ 2x_1 & + & 8x_2 & & & & & + & x_5 & = & 1000 \end{array} \tag{2.7}$$

and

$$x_1, x_2, x_3, x_4, x_5 \geq 0. \tag{2.8}$$

Initialization step

We have already seen that we can get a basic feasible solution by putting each of the decision variables equal to 0 and solving the system (2.7) for the slack variables (this solution corresponds to the corner-point O). Thus our initial basic feasible solution is $[0, 0, 2400, 720, 1000]^T$; the variables x_1, x_2 are non-basic and the basis at this solution is $\{x_3, x_4, x_5\}$.

Note carefully that the columns of the matrix $[\mathbf{A} : \mathbf{I}]$ corresponding to the *basic* variables are columns of the 3×3 identity matrix. This makes it very easy to solve the system for the values of the basic variables. Further, the objective function is expressed in terms of non-basic variables only. When the system is in this form, we shall say it is in **proper form** for the next iteration.

Test for optimality

The current value of the objective function is $z = 6x_1 + 5x_2 = 0$. Can we increase the value of z by moving to an adjacent basic feasible solution? The answer is yes; since both variables have positive coefficients in this expression for z, we can increase z by increasing either x_1 or x_2 from zero. That is, we allow one of these variables to enter the basis and thus to take a positive value.

Iterative step

The move to an adjacent basic feasible solution which gives a better value of z consists of a sequence of three operations, summarized below.

1. Choose one variable that is currently non-basic to enter the basis. This variable is called the **entering variable.**

2. Determine which variable in the current basis should leave the basis to make room for the entering variable (recollect that the basis always contains precisely m variables, where m is the number of equations in the augmented system). This variable is called the **leaving variable.**

3. Convert the constraint equations and the objective function to proper form for the next iteration.

We now consider in detail how each of these operations should be performed.

1. *Choose an entering variable.* We have already decided in the test for optimality that we can increase z by increasing either x_1 or x_2 from zero. *Any* variable with a positive coefficient in the objective function may be chosen as the entering variable. However, when there is more than one candidate for entering variable, we usually choose the one that has the *highest positive coefficient* in the objective function. By making this choice, we hope to increase the value of z by as much as possible at each iteration, and thus minimize the total number of iterations needed to reach the optimum. Using this rule, we select x_1 as the entering variable.

2. *Choose the leaving variable.* The candidates for leaving variable are those in the current basis. We want to increase the entering variable x_1 as much as possible in order to make the maximum increase in z, *subject to the condition that the solution remains feasible.*

   At this first iteration, the basic variables are just the slack variables. To understand the general rule for choosing the leaving variable, it is helpful first to consider the physical interpretation of what happens to these variables as the entering variable increases. Recollect that the slack variables represent the difference between the total amount of each resource available and the amount of each resource used at the current solution. As we *increase* x_1 from 0 through positive values, the value of each of the slack variables *decreases*. This represents the fact that as we increase the production of dinghies, we use up an increasing amount of the available resources. Eventually, one of the resources will be exhausted and at this point the corresponding slack variable becomes zero. We cannot increase x_1 further, because then this slack variable would become negative and the corresponding solution would be infeasible. Thus we increase x_1 just until one of the slack variable becomes zero; the first slack variable to become zero can then be made non-basic, so that it leaves the basis.

Thus the general rule is this: *the leaving variable is the first basic variable to become zero as the entering variable increases.*

To determine which basic variable is the first to become zero as x_1 increases, we consider each equation in turn. We can disregard the terms in x_2, because x_2 remains zero while we increase x_1. Then for each basic variable, we calculate the maximum amount by which x_1 can be increased before that basic variable becomes zero. This calculation is displayed in the table below.

Basic variable	*Equation*	*Upper bound on x_1*
x_3	$24x_1 + x_3 = 2400$	$x_1 \leq 2400/24 = 100$
x_4	$6x_1 + x_4 = 720$	$x_1 \leq 720/6 = 120$
x_5	$2x_1 + x_5 = 1000$	$x_1 \leq 1000/2 = 500$

The table shows that the basic variables remain positive while x_1 increases from 0 to 100, at which point $x_3 = 0$. Thus x_3 is the first variable to become zero as x_1 increases and so x_3 is the leaving variable.

3. *Convert the constraint equations and objective function to proper form.* We shall do this by Gaussian elimination.

   The leaving variable occurs in just one of the constraint equations. This equation is called the **pivot equation** for the current iteration. In this case, the leaving variable is x_3 and hence the pivot equation is

$$24x_1 + 6x_2 + x_3 = 2400. \tag{2.9}$$

At this iteration, our aim is to interchange the roles of x_3 and x_1. We start by dividing through the pivot equation by the coefficient of x_1, so that in the revised form of the pivot equation, the coefficient of the entering variable is 1. This gives

$$x_1 + 1/4x_2 + 1/24x_3 = 100. \tag{2.10}$$

We now use (2.10) to eliminate x_1 from the objective function and the other constraint equations.

The first step is to rewrite the objective function in the same form as the constraint equations with all variables on the left side of the equation. Thus we have

$$z - 6x_1 - 5x_2 = 0 \tag{2.11}$$

To eliminate x_1 from equation (2.11), we add 6 times (2.10) to (2.11).

$$\begin{array}{rcrcrcrcr}
z & - & 6x_1 & - & 5x_2 & & & = & 0 \\
 & & 6x_1 & + & 3/2x_2 & + & 1/4x_3 & = & 600 \\
\hline
z & & & - & 7/2x_2 & + & 1/4x_3 & = & 600
\end{array}$$

Next, we eliminate x_1 from the other two constraint equations. Subtracting 6 times (2.10) from the second equation of the system (2.7), we obtain

$$\begin{array}{rcrcrcrcr} 6x_1 & + & 4x_2 & & & + & x_4 & = & 720 \\ 6x_1 & + & 3/2x_2 & + & 1/4x_3 & & & = & 600 \\ \hline & & 5/2x_2 & - & 1/4x_3 & + & x_4 & = & 120 \end{array}$$

and subtracting 2 times (2.10) from the third equation in the system (2.7) gives

$$\begin{array}{rcrcrcrcr} 2x_1 & + & 8x_2 & & & + & x_5 & = & 1000 \\ 2x_1 & + & 1/2x_2 & + & 1/12x_3 & & & = & 200 \\ \hline & & 15/2x_2 & - & 1/12x_3 & + & x_5 & = & 800 \end{array}$$

We have now converted the system (2.7) to the following equivalent form.

$$\begin{array}{rrcrcrcrcrcr} z & & - & 7/2x_2 & + & 1/4x_3 & & & & & = & 600 \\ & x_1 & + & 1/4x_2 & + & 1/24x_3 & & & & & = & 100 \\ & & & 5/2x_2 & - & 1/4x_3 & + & x_4 & & & = & 120 \\ & & & 15/2x_2 & - & 1/12x_3 & & & + & x_5 & = & 800 \end{array} \qquad (2.12)$$

The system is now in proper form with the new basis $\{x_1, x_4, x_5\}$. Thus we have completed the iterative step.

Putting the non-basic variables x_2 and x_3 equal to 0 and solving (2.12) for the basic variables and z, we obtain the new basic feasible solution $\mathbf{x} = [100, 0, 0, 120, 800]^T$ and $z = 600$. This is the feasible corner-point solution at A in Figure 2.2.

We have now completed the first iteration of all the steps in the algorithm and have returned to the test for optimality.

Test for optimality. The objective equation in the system (2.12), gives

$$z = 600 + 7/2x_2 - 1/4x_3.$$

Since x_2 has a positive coefficient in this expression, we can increase z by increasing x_2 from zero, while keeping $x_3 = 0$. Thus the current solution is not optimal.

Iterative step

1. We have identified the entering variable as x_2 in the previous step.

2. We use the system (2.12) with $x_3 = 0$ to determine the corresponding leaving variable.

Basic variable	*Equation*	*Upper bound on* x_2
x_1	$x_1 + 1/4x_2 = 100$	$x_2 \leq 100/0.25 = 400$
x_4	$5/2x_2 + x_4 = 120$	$x_2 \leq 120/2.5 = 48$
x_5	$15/2x_2 + x_5 = 800$	$x_2 \leq 800/7.5 = 106.67$

The table shows that the basic variables remain positive while x_2 increases from 0 to 48, at which point $x_4 = 0$. Thus x_4 is the leaving variable.

3. We interchange the roles of x_2 and x_4 and convert the system to proper form with the new basis $\{x_1, x_2, x_5\}$. The pivot equation is the equation in the system (2.12) in which the leaving variable x_4 appears. This is the equation

$$5/2x_2 - 1/4x_3 + x_4 = 120.$$

Dividing through by the coefficient of the entering variable x_2, we have

$$x_2 - 1/10x_3 + 2/5x_4 = 48. \qquad (2.13)$$

We now eliminate the entering variable x_2 from the objective equation and the first and third constraint equations in the system (2.12).

Adding 7/2 times equation (2.13) to the objective equation in (2.12), we obtain

$$z - 1/10x_3 + 7/5x_4 = 768;$$

subtracting 1/4 times equation (2.13) from the first constraint equation in (2.12), we obtain

$$x_1 + 1/15x_3 - 1/10x_4 = 88;$$

and finally, subtracting 15/2 times equation (2.13) from the third constraint equation in (2.12), gives

$$2/3x_3 - 3x_4 + x_5 = 440.$$

Thus we have converted the system (2.12) to the following equivalent form.

$$\begin{array}{rrrrrrrrrr} z & & & - & 1/10x_3 & + & 7/5x_4 & & & = & 768 \\ & x_1 & & + & 1/15x_3 & - & 1/10x_4 & & & = & 88 \\ & & x_2 & - & 1/10x_3 & + & 2/5x_4 & & & = & 48 \\ & & & & 2/3x_3 & - & 3x_4 & + & x_5 & = & 440 \end{array} \tag{2.14}$$

The new basis is $\{x_1, x_2, x_5\}$ and the system (2.14) is therefore in proper form. This completes the iterative step.

Putting the two non-basic variables x_3 and x_4 equal to 0 and solving (2.14) for the basic variables and z, gives the new basic feasible solution $\mathbf{x} = [88, 48, 0, 0, 440]^T$ and $z = 768$. This is the feasible corner-point solution at B in Figure 2.2.

This completes the second iteration of the steps in the algorithm, and we return to the test for optimality.

Test for optimality. The objective equation in the system (2.14), gives

$$z = 768 + 1/10x_3 - 7/5x_4.$$

Thus we can increase z by increasing x_3 from zero, while keeping $x_4 = 0$, so the current solution is not optimal.

Iterative step

1. The entering variable is x_3.

2. We use the system (2.14) to determine the corresponding leaving variable.

Basic variable	*Equation*	*Upper bound on x_3*
x_1	$x_1 + 1/15x_3 = 88$	$x_3 \leq 88 \times 15 = 1320$
x_2	$x_2 - 1/10x_3 = 48$	No limit on x_3
x_5	$2/3x_3 + x_5 = 440$	$x_3 \leq 440 \times 3/2 = 660$

The table shows that x_5 is the first variable to become zero as x_3 increases, so x_5 is the leaving variable.

3. We interchange the roles of x_5 and x_3 and convert the system to proper form with the new basis $\{x_1, x_2, x_3\}$. The pivot equation is the equation in the system (2.14) in which the leaving variable x_5 is basic. This is the equation

$$2/3x_3 - 3x_4 + x_5 = 440.$$

Dividing through by the coefficient of the entering variable x_3, gives

$$x_3 - 9/2x_4 + 3/2x_5 = 660. \tag{2.15}$$

We now eliminate the entering variable x_3 from the objective equation and the first and second constraint equations in the system (2.14).

Adding 1/10 times equation (2.15) to the objective equation in (2.14), we obtain

$$z + 19/20x_4 + 3/20x_5 = 834;$$

subtracting 1/15 times equation (2.15) from the first constraint equation in (2.14), we obtain

$$x_1 + 1/5x_4 - 1/10x_5 = 44;$$

and finally, adding 1/10 times equation (2.15) from the second constraint equation in (2.14), gives

$$x_2 - 1/20x_4 + 3/20x_5 = 114.$$

Thus we have converted the system (2.14) to the following equivalent form.

$$\begin{array}{cccccccccc} z & & & & + & 19/20x_4 & + & 3/20x_5 & = & 834 \\ & x_1 & & & + & 1/5x_4 & - & 1/10x_5 & = & 44 \\ & & x_2 & & - & 1/20x_4 & + & 3/20x_5 & = & 114 \\ & & & x_3 & - & 9/2x_4 & + & 3/2x_5 & = & 660 \end{array} \tag{2.16}$$

The new basis is $\{x_1, x_2, x_3\}$ and the system (2.16) is therefore in proper form. Thus we have completed the iterative step.

Putting the two non-basic variables x_4 and x_5 equal to 0 and solving (2.16) for the basic variables and z, gives the new basic feasible solution $\mathbf{x} = [44, 114, 660, 0, 0]^T$ and $z = 834$. This is the feasible corner-point solution at C in Figure 2.2.

This completes the third iteration of the steps in the algorithm and we return again to the test for optimality.

Test for optimality. The objective equation in the system (2.14), gives

$$z = 834 - 19/20x_4 - 3/20x_5.$$

This equation, together with the constraints $x_4 \geq 0$ and $x_5 \geq 0$, implies that the maximum value of z is 834. Further, to achieve $z = 834$, we must have $x_4 = 0$ and $x_5 = 0$. Hence, the current basic feasible solution is optimal, agreeing with the graphical solution obtained in Example 1.3.

2.5 Tableau method for the simplex algorithm

Just as the steps in the solution of a system of simultaneous equations by Gaussian elimination can be recorded by operating on the augmented matrix of coefficients, we can do a similar thing for the iterations of the simplex algorithm. In this case, we also record the basis at each iteration and the calculation for determining the leaving variable, but the principle behind it is the same: we can work faster and in a more organised way by operating just on the matrix of coefficients.

We start as before with a problem in standard form, with $\mathbf{b} \geq \mathbf{0}$, and add slack variables to obtain the augmented form. Then putting the n decision variables equal to 0 will give an initial basic feasible solution, with the m slack variables forming the first basis. We test this solution for optimality, just as described in the previous section. If this solution is not optimal, we rewrite the objective function with all variables on the left side of the equation. We are then ready to form the intitial tableau.

As an illustration of the method, we describe how to use tableaux to represent the calculations of the previous section for finding the optimal solution of the Boatbuilder's problem by the simplex algorithm. We construct the intitial tableau as shown below, where the heading "RS" denotes the *right side* of the equations, and "EV" denotes the *entering variable.* The coefficients in the objective equation (2.11) are recorded in the first row as equation (0); the coefficients in the constraint equations (2.7) are recorded in the next three rows, as equations (1), (2) and (3) respectively.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-6	-5	0	0	0	0	
x_3	1	0	24	6	1	0	0	2400	
x_4	2	0	6	4	0	1	0	720	
x_5	3	0	2	8	0	0	1	1000	

Note that we can easily read off the current basic feasible solution and corresponding value of z from this array. Putting the non-basic variables x_1, x_2 equal to zero, the corresponding equations become:

$$\begin{aligned} z &= 0 \\ x_3 &= 2400 \\ x_4 &= 720 \\ x_5 &= 1000. \end{aligned}$$

This gives the initial basic feasible solution $[0, 0, 2400, 720, 1000]^T$.

1. *Decide on the entering variable.* Since all terms in the variables have been moved to the *left* side of the objective equation, the coefficient of the entering variable appears as the *negative* number with the largest absolute value in the objective row. In this example, the first entering variable is x_1.

   Since the coefficients of the entering variable in the constraint equations play a key role in the determination of the leaving variable and in the subsequent Gaussian elimination step, it is helpful to indicate the column belonging to the entering variable by drawing a box round it.

2. *Determine the leaving variable.* The calculation of the maximum amount by which the entering variable can be increased before each basic variable becomes negative is performed exactly as described in the previous section and displayed in the final column of the tableau. The leaving variable is the first basic variable to become zero as the entering variable increases through positive values. This is the variable that gives the *lowest* upper bound on the entering variable (in this case, x_1). We indicate this lowest value with an arrow, to record how we made our choice of leaving variable. We put a box round the coefficients in the constraint equation belonging to this variable, as this is going to be the pivot row for Gaussian elimination.

   In this case, the leaving variable is x_3 and so the pivot row for constructing the second tableau is equation (1). The table below shows the initial tableau after these two operations.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-6	-5	0	0	0	0	-
x_3	1	0	24	6	1	0	0	2400	$x_1 \leq 2400/24 = 100 \leftarrow$
x_4	2	0	6	4	0	1	0	720	$x_1 \leq 720/6 = 120$
x_5	3	0	2	8	0	0	1	1000	$x_1 \leq 1000/2 = 500$

3. *Constructing the tableau to represent the second basic feasible solution.* The second tableau is placed under the first tableau as shown below. Fill in the equation numbers 0,1,2,3, as in the first tableau, and also the entries 1,0,0,0 for the coefficients of z, as these remain unchanged during the subsequent Gaussian elimination.

   The next step is to divide through the pivot row by the coefficient of the entering variable in this row. This coefficient is easily identified because it is at the intersection of the horizontal and vertical boxes. Enter the revised coefficients of the pivot row in the corresponding row in the second tableau. At the same time, exchange the leaving variable in the basis (first column) for the entering variable.

   In this case, the pivot row is equation (1) and the coefficient of the entering variable x_1 is 24. We divide through row (1) by 24 and exchange x_3 for x_1 in the basis, recording the result in row (1) of the second tableau as shown below.

   Note that the entries in the tableaux may be recorded either in fractions or in decimal from. The latter will often introduce rounding errors, but is the only practical way to work with larger systems. In this example, we will stick to fractions, so that you can more easily compare the entries in the tableaux with the equations we obtained in the previous section.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-6	-5	0	0	0	0	-
x_3	1	0	24	6	1	0	0	2400	$x_1 \le 2400/24 = 100 \leftarrow$
x_4	2	0	6	4	0	1	0	720	$x_1 \le 720/6 = 120$
x_5	3	0	2	8	0	0	1	1000	$x_1 \le 1000/2 = 500$
-	0	1							
x_1	1	0	1	1/4	1/24	0	0	100	
	2	0							
	3	0							

   The next step is to use Gaussian elimination to convert the column of the second tableau belonging to the entering variable (in this case, x_1) to a column of the identity matrix. Thus we add or subtract suitable multiples of the pivot row to the other rows so that all other entries in this column are zero.

   Hence, in this example, we perform the following row operations, where "new row (1)" refers to row (1) in the second tableau.

   (a) Add 6 times new row (1) to row (0) and enter the result in row (0) of the second tableau;

   (b) subtract 6 times new row (1) from row (2) and enter the result in row (2) of the second tableau; the basic variable in equation (2) remains x_4;

   (c) subtract 2 times new row (1) from row (3) and enter the result in row (3) of the second tableau; the basic variable in equation (3) remains x_5.

The second tableau is now complete and we have the following array.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-6	-5	0	0	0	0	-
x_3	1	0	24	6	1	0	0	2400	$x_1 \le 2400/24 = 100 \leftarrow$
x_4	2	0	6	4	0	1	0	720	$x_1 \le 720/6 = 120$
x_5	3	0	2	8	0	0	1	1000	$x_1 \le 1000/2 = 500$
-	0	1	0	$-7/2$	1/4	0	0	600	
x_1	1	0	1	1/4	1/24	0	0	100	
x_4	2	0	0	5/2	$-1/4$	1	0	120	
x_5	3	0	0	15/2	$-1/12$	0	1	800	

The second tableau corresponds to the system (2.12). We can easily read off the current basic feasible solution and corresponding value of z from the second tableau. Putting the non-basic variables x_2, x_3 equal to zero, the equations corresponding to the rows of the array become:

$$\begin{aligned} z &= 600 \\ x_1 &= 100 \\ x_4 &= 120 \\ x_5 &= 800. \end{aligned}$$

This gives the current basic feasible solution as $[100, 0, 0, 120, 800]^T$.

We now test this solution for optimality. Equation (0) gives

$$z = 600 + 7/2x_2 - 1/4x_3.$$

Since x_2 has a positive coefficient in this expression, we can increase the current value of z by increasing x_2 from zero, while keeping $x_3 = 0$. Thus the current solution is not optimal and x_2 is the entering variable.

Once you understand the principle behind the decision on optimality and the choice of entering variable, you can make these decisions just by observing the entries in the objective row of the tableau, without rewriting the objective equation. Thus, in general, the current solution is *not* optimal if the objective row contains a *negative* entry; and in this case, the variable with the negative coefficient of largest absolute value in the objective row is chosen as the next entering variable.

The calculations to determine the corresponding leaving variable are set out in the final column of the second tableau, as shown below.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	$-7/2$	$1/4$	0	0	600	-
x_1	1	0	1	$1/4$	$1/24$	0	0	100	$x_2 \leq 100/0.25 = 400$
x_4	2	0	0	$5/2$	$-1/4$	1	0	120	$x_2 \leq 120/2.5 = 48 \leftarrow$
x_5	3	0	0	$15/2$	$-1/12$	0	1	800	$x_2 \leq 800/7.5 = 106.67$

Thus the leaving variable is x_4 and equation (2) becomes the pivot row for the next iteration. The coefficient of the entering variable in the pivot row is $5/2$. We divide through the coefficients in equation (2) by $5/2$ and write down the revised pivot row in row (2) of the third tableau. We also exchange x_2 for x_4 as the basic variable in this row. This gives the following array.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	$-7/2$	$1/4$	0	0	600	-
x_1	1	0	1	$1/4$	$1/24$	0	0	100	$x_2 \leq 100/0.25 = 400$
x_4	2	0	0	$5/2$	$-1/4$	1	0	120	$x_2 \leq 120/2.5 = 48 \leftarrow$
x_5	3	0	0	$15/2$	$-1/12$	0	1	800	$x_2 \leq 800/7.5 = 106.67$
	0	1							
	1	0							
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	
	3	0							

We now use Gaussian elimination, with the new row (2) as pivot row, to convert the column belonging to the new entering variable x_2 to a column of the identity matrix. Thus we perform the following row operations.

1. Add 7/2 times new row (2) to row (0) and enter the result in row (0) of the new tableau;

2. subtract 1/4 times new row (2) from row (1) and enter the result in row (1) of the second tableau; the basic variable in equation (1) remains x_1;

3. subtract 15/2 times new row (2) from row (3) and enter the result in row (3) of the new tableau; the basic variable in equation (3) remains x_5.

We show below the second and third tableaux after these operations. The third tableau represents the system (2.14).

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	$-7/2$	$1/4$	0	0	600	-
x_1	1	0	1	$1/4$	$1/24$	0	0	100	$x_2 \leq 100/0.25 = 400$
x_4	2	0	0	$5/2$	$-1/4$	1	0	120	$x_2 \leq 120/2.5 = 48 \leftarrow$
x_5	3	0	0	$15/2$	$-1/12$	0	1	800	$x_2 \leq 800/7.5 = 106.67$
-	0	1	0	0	$-1/10$	$7/5$	0	768	
x_1	1	0	1	0	$1/15$	$-1/10$	0	88	
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	
x_5	3	0	0	0	$2/3$	-3	1	440	

Putting the non-basic variables x_3, x_4 equal to zero, the third tableau reduces to the system

$$\begin{aligned} z &= 768 \\ x_1 &= 88 \\ x_2 &= 48 \\ x_5 &= 440, \end{aligned}$$

giving the current basic feasible solution as $[88, 48, 0, 0, 440]^T$.

We return to the test for optimality. Since x_3 has a negative coefficient in the objective row, the value of z can be increased by increasing x_3 from zero. Thus the current solution is not optimal and x_3 is the next entering variable.

The calculations for the corresponding leaving variable are entered in the last column of the array. Note that x_3 has a *negative* coefficient in equation (2). With $x_4 = 0$, this equation becomes $x_2 - 1/10x_3 = 48$, or

$$x_2 = 48 + 1/10x_3.$$

Hence as x_3 increases from zero through positive values, x_2 increases also. Thus x_2 will not be made negative by increasing x_3.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	0	$-1/10$	$7/5$	0	768	-
x_1	1	0	1	0	$1/15$	$-1/10$	0	88	$x_3 \leq 88 \times 15 = 1320$
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	No limit on x_3
x_5	3	0	0	0	$2/3$	-3	1	440	$x_3 \leq 440 \times 3/2 = 660 \leftarrow$

Thus the leaving variable is x_5 and the pivot row for the next iteration is row (3). The coefficient of the entering variable x_3 in the pivot row is 2/3. We divide through the coefficients in equation (3) by 2/3 and write down the revised pivot row in row (3) of the fourth tableau. We also exchange x_5 for x_3 as the basic variable in this row. This gives the following array.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	0	$-1/10$	$7/5$	0	768	-
x_1	1	0	1	0	$1/15$	$-1/10$	0	88	$x_3 \le 88 \times 15 = 1320$
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	No limit on x_3
x_5	3	0	0	0	$\boxed{2/3}$	-3	1	440	$x_3 \le 440 \times 3/2 = 660 \leftarrow$
	0	1							
	1	0							
	2	0							
x_3	3	0	0	0	1	$-9/2$	$3/2$	660	

We now use Gaussian elimination, with the new row (3) as pivot row, to convert the column belonging to the new entering variable x_3 to a column of the identity matrix. Thus we perform the following row operations.

1. Add 1/10 times new row (3) to row (0) and enter the result in row (0) of the new tableau;
2. subtract 1/15 times new row (3) from row (1) and enter the result in row (1) of the new tableau; the basic variable in equation (1) remains x_1;
3. add 1/10 times new row (3) to row (2) and enter the result in row (2) of the new tableau; the basic variable in equation (2) remains x_2.

The third and fourth tableaux after these operations are shown below.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	0	$-1/10$	$7/5$	0	768	-
x_1	1	0	1	0	$1/15$	$-1/10$	0	88	$x_3 \le 88 \times 15 = 1320$
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	No limit on x_3
x_5	3	0	0	0	$2/3$	-3	1	440	$x_3 \le 440 \times 3/2 = 660 \leftarrow$
-	0	1	0	0	0	$19/20$	$3/20$	834	
x_1	1	0	1	0	0	$1/5$	$-1/10$	44	
x_2	2	0	0	1	0	$-1/20$	$3/20$	114	
x_3	3	0	0	0	1	$-9/2$	$3/2$	660	

Since all entries in the objective row of the new tableau are positive, increasing either of the non-basic variables x_4 or x_5 from zero would *decrease* the value of z. Hence z has achieved its maximum value at the current iteration and further, z only achieves this maximum value when $x_4 = 0$ and $x_5 = 0$.

Putting $x_4 = 0 = x_5$, the equations represented by this tableau give

$$z = 834,\ x_1 = 44,\ x_2 = 114,\ x_3 = 660.$$

This gives the optimal solution as $x_1 = 44$, $x_2 = 114$ and the maximum value of z as 834, as we obtained before.

When using the tableau method, we normally display the tableaux one after the other without leaving a space in between. The column boxes indicate the choice of entering variable and the working in the final column explains the choice of leaving variable, so that no explanation is required until an optimal basic feasible solution is reached. For this problem, the finished tableaux would appear as shown below.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-6	-5	0	0	0	0	-
x_3	1	0	24	6	1	0	0	2400	$x_1 \leq 2400/24 = 100 \leftarrow$
x_4	2	0	6	4	0	1	0	720	$x_1 \leq 720/6 = 120$
x_5	3	0	2	8	0	0	1	1000	$x_1 \leq 1000/2 = 500$
-	0	1	0	$-7/2$	$1/4$	0	0	600	-
x_1	1	0	1	$1/4$	$1/24$	0	0	100	$x_2 \leq 100/0.25 = 400$
x_4	2	0	0	$5/2$	$-1/4$	1	0	120	$x_2 \leq 120/2.5 = 48 \leftarrow$
x_5	3	0	0	$15/2$	$-1/12$	0	1	800	$x_2 \leq 800/7.5 = 106.67$
-	0	1	0	0	$-1/10$	$7/5$	0	768	-
x_1	1	0	1	0	$1/15$	$-1/10$	0	88	$x_3 \leq 88 \times 15 = 1320$
x_2	2	0	0	1	$-1/10$	$2/5$	0	48	No limit on x_3
x_5	3	0	0	0	$2/3$	-3	1	440	$x_3 \leq 440 \times 3/2 = 660 \leftarrow$
-	0	1	0	0	0	$19/20$	$3/20$	834	
x_1	1	0	1	0	0	$1/5$	$-1/10$	44	
x_2	2	0	0	1	0	$-1/20$	$3/20$	114	
x_3	3	0	0	0	1	$-9/2$	$3/2$	660	

Once you are sure you understand the reason for each of the steps in the simplex algorithm, you will need to practice this process until you can use the tableau method confidently. There are some suitable exercises at the end of this chapter.

A useful point to note is this: the entries in the column RS representing the right side of the equations give the current values of the basic variables. Thus *no entry in this column should ever be negative*, because this would give an *infeasible* solution. If you should calculate a negative entry in this column, then stop working and check your arithmetic for that entry. If that seems to be correct, then go back to the *previous tableau* and check your calculations for the choice of leaving variable. It is very likely that you chose the wrong variable, so that you have strayed to a corner-point outside the feasible region.

2.6 Special situations

In this section we consider what we should do when there is a tie for entering or leaving variable, and how to recognise from the simplex algorithm if the problem has multiple optimal solutions or if the objective function has no finite optimal value. The possibility that there is no feasible solution is discussed in the next chapter, as this cannot arise with l.p. problems in standard form with $\mathbf{b} \geq \mathbf{0}$. This is because we can always find a feasible solution by putting all the decision variables equal to 0.

Tie for entering variable

Suppose, for example, that we are maximizing $z = 5x_1 + 2x_2 + 5x_3$, where x_1, x_2, x_3 are non-basic variables. Our rule is to choose as entering variable the non-basic variable with the largest positive coefficient in the objective function (or, equivalently, the variable with the negative coefficient with largest absolute value in the objective row of the tableau). Thus in this example there is a tie for entering variable between x_1 and x_3. Does it matter which we choose? The answer to this question is, no; we can ALWAYS settle ties for entering variable arbitrarily.

Tie for leaving variable

A tie for leaving variable occurs if, as we increase the entering variable from zero through positive values, there are two or more of the current basic variables that are simultaneously first to become zero. We consider first an example where this occurs.

Example 2.12 Find $x_1, x_2 \in \mathbf{R}$ to maximize $z = 3x_1 - x_2$, subject to

$$\begin{aligned} 2x_1 - x_2 &\leq 4 \\ x_1 - 2x_2 &\leq 2 \\ x_1 + x_2 &\leq 5 \\ x_1, x_2 &\geq 0. \end{aligned}$$

Adding slack variables, we obtain the following system.

$$\begin{array}{rcrcrcrcrcl} 2x_1 & - & x_2 & + & x_3 & & & & & = & 4 \\ x_1 & - & 2x_2 & & & + & x_4 & & & = & 2 \\ x_1 & + & x_2 & & & & & + & x_5 & = & 5 \end{array} \tag{2.17}$$

where

$$x_1, x_2, x_3, x_4, x_5 \geq 0.$$

Putting the decision variables x_1, x_2 equal to zero, we get the initial basic feasible solution $\mathbf{x} = [0, 0, 4, 2, 5]^T$, giving $z = 0$.

Testing for optimality, we can increase z by increasing x_1 from zero, while keeping $x_2 = 0$. Hence the current solution is not optimal and we must choose x_1 as the entering variable. Rewriting the objective function with all terms in variables on the left side of the equation gives

$$z - 3x_1 + x_2 = 0.$$

The intitial tableau is shown below, with the calculation for leaving variable shown in the last column.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-3	1	0	0	0	0	-
x_3	1	0	2	-1	1	0	0	4	$x_1 \leq 4/2 = 2 \leftarrow$
x_4	2	0	1	-2	0	1	0	2	$x_1 \leq 2/1 = 2 \leftarrow$
x_5	3	0	1	1	0	0	1	5	$x_1 \leq 5/1 = 5$

Thus there is a tie for leaving variable between x_3 and x_4. Suppose we choose x_4 as the next leaving variable. Then row (2) becomes the pivot row , x_1 replaces x_4 in the basis, and we obtain the following tableaux.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	-3	1	0	0	0	0	-
x_3	1	0	2	-1	1	0	0	4	$x_1 \leq 4/2 = 2 \leftarrow$
x_4	2	0	1	-2	0	1	0	2	$x_1 \leq 2/1 = 2 \leftarrow$
x_5	3	0	1	1	0	0	1	5	$x_1 \leq 5/1 = 5$
-	0	1	0	-5	0	3	0	6	
x_3	1	0	0	2	1	-2	0	0	
x_1	2	0	1	-2	0	1	0	2	
x_5	3	0	0	3	0	-1	1	3	

Putting the non-basic variables x_2, x_4 equal to zero, the second tableau reduces to the system

$$z = 6,\ x_3 = 0,\ x_1 = 2,\ x_5 = 3,$$

and the current basic feasible solution is $\mathbf{x} = [2, 0, 0, 0, 3]^T$.

Thus, as a result of the tie between x_3 and x_4 for leaving variable in the first basis, the variable that was *not chosen* as leaving variable has the value 0 in the next basis.

Definition 2.8 *A basis containing a variable which has value 0 is called* **degenerate** *and the corresponding basic feasible solution is also called* **degenerate**.

Since the objective row contains a negative entry, we investigate what happens at the next iteration. The entering variable is x_2. Below we show the calculation for the corresponding leaving variable and the third tableau.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	-5	0	3	0	6	-
x_3	1	0	0	2	1	-2	0	0	$x_2 \le 0/2 = 0 \leftarrow$
x_1	2	0	1	-2	0	1	0	2	no limit on x_2
x_5	3	0	0	3	0	-1	1	3	$x_2 \le 3/3 = 1$
-	0	1	0	0	$5/2$	-2	0	6	
x_2	1	0	0	1	$1/2$	-1	0	0	
x_1	2	0	1	0	1	-1	0	2	
x_5	3	0	0	0	$-3/2$	2	1	3	

Putting the non-basic variables x_3 and x_4 equal to 0, the third tableau reduces to the system

$$z = 6,\ x_2 = 0,\ x_1 = 2,\ x_5 = 3,$$

and the current basic feasible solution is still $\mathbf{x} = [2, 0, 0, 0, 3]^T$.

Thus after this iteration, the basic feasible solution is exactly the same as it was at the end of the previous iteration and as a result the value of z has not increased.

However, the objective row still contains a negative entry and so the current solution is not yet optimal. The entering variable is now x_4. Below we show the calculation for the corresponding leaving variable and the fourth tableau.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	0	$5/2$	-2	0	6	-
x_2	1	0	0	1	$1/2$	-1	0	0	no limit on x_4
x_1	2	0	1	0	1	-1	0	2	no limit on x_4
x_5	3	0	0	0	$-3/2$	2	1	3	$x_4 \le 3/2 \leftarrow$
-	0	1	0	0	1	0	1	9	
x_2	1	0	0	1	$-1/4$	0	$1/2$	$3/2$	
x_1	2	0	1	0	$1/4$	0	$1/2$	$7/2$	
x_4	3	0	0	0	$-3/4$	1	$1/2$	$3/2$	

Since there are no negative entries in the objective row of the last tableau, the solution is now optimal. Putting the non-basic variables x_3 and x_5 equal to 0, the fourth tableau reduces to the system

$$z = 9,\ x_2 = 3/2,\ x_1 = 7/2,\ x_4 = 3/2.$$

Thus the basic feasible solution given by the final tableau is $\mathbf{x} = [3.5, 1.5, 0, 1.5, 0]^T$. The optimal solution is therefore $x_1 = 3.5$, $x_2 = 1.5$ and the maximum value of z is 9. □

We learn from the above example that when there is a tie for leaving variable, then the next basis is degenerate. We could also obtain a degenerate basis at the beginning of the algorithm, if one or more of the components of **b** is zero. When we iterate from a degenerate solution, it may happen that instead of moving to a new basic feasible solution at which z has a better value, the algorithm stays at the same degenerate solution (geometrically, the same corner-point of the feasible region) and interchanges a basic variable that has value 0 with a non-basic variable (which by definition also has value 0). Thus the value of z stays the same. If a number of variables tie for leaving variable, then it is theoretically possible for our implementation of the simplex algorithm to go into an endless loop, constantly interchanging the same set of basic variables with value zero, while the basic feasible solution and value of z remain the same. This phenomenom is called **cycling**. Thus, *in theory*, it is important how we break ties for leaving variable.

However, although artificial examples have been contrived to show that cycling is possible in the simplex algorithm, it has very rarely been known to occur in practice. This may be partly because the form of the constraints that give rise to cycling in the contrived examples rarely arise from practical problems and partly because of rounding errors. Thus for most practical purposes, it is sufficient to settle ties for leaving variable arbitrarily. There are rules for avoiding cycling in large scale applications of the simplex algorithm, but you are not required to know them for the examination for this course.

No finite optimal solution

In Exercise 1.5(a), we met an example of an l.p. problem in standard form that has no finite optimal solution, because there is no limit on the amount by which z can be increased. We solve the same problem here by the simplex algorithm, to show how we can recognize when this situation has arisen.

Example 2.13 Find $x_1, x_2 \in \mathbf{R}$ to maximize $z = x_1 + 2x_2$, subject to

$$\begin{aligned} x_1 - 5x_2 &\le 5 \\ -x_1 + x_2 &\le 4 \\ x_1, x_2 &\ge 0. \end{aligned}$$

Adding slack variables, we obtain the following system.

$$\begin{array}{rcrcrcrcl} x_1 & - & 5x_2 & + & x_3 & & & = & 5 \\ -x_1 & + & x_2 & & & + & x_4 & = & 4 \end{array} \tag{2.18}$$

where

$$x_1, x_2, x_3, x_4 \ge 0.$$

Putting the decision variables x_1, x_2 equal to zero, we get the initial basic feasible solution $\mathbf{x} = [0, 0, 5, 4]^T$, giving $z = 0$.

Testing for optimality, we can increase z by increasing x_1 or x_2 from zero. Hence the current solution is not optimal. We choose x_2 as the entering variable, because it has the larger coefficient. Rewriting the objective function with all terms in variables on the left side of the equation gives

$$z - x_1 - 2x_2 = 0.$$

The intitial tableau is shown below, with the calculation for leaving variable shown in the last column.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	RS	*upper bound on EV*
-	0	1	-1	-2	0	0	0	-
x_3	1	0	1	-5	1	0	5	No limit on x_2
x_4	2	0	-1	$\boxed{1}$	0	1	4	$x_2 \le 4 \leftarrow$

Thus the leaving variable is x_4 and the pivot equation is (2). The coefficient of the entering variable in equation (2) is 1, so we copy row (2) as it is to become row (2) of the second tableau and interchange x_4 with x_2 in the basis. Performing Gaussian elimination to convert the column belonging to x_2 to a column of the identity matrix, we obtain the tableaux shown below.

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	RS	*upper bound on EV*
-	0	1	-1	-2	0	0	0	-
x_3	1	0	1	-5	1	0	5	No limit on x_2
x_4	2	0	-1	1	0	1	4	$x_2 \leq 4 \leftarrow$
-	0	1	-3	0	0	2	8	-
x_3	1	0	-4	0	1	5	25	No limit on x_2
x_2	2	0	-1	1	0	1	4	No limit on x_2

Putting the non-basic variables x_1, x_4 equal to zero, the second tableau reduces to the system

$$z = 8,\ x_3 = 25,\ x_2 = 4,$$

and the current basic feasible solution is $\mathbf{x} = [0, 4, 25, 0]^T$.

Since the objective row in the second tableau contains a negative coefficient, this solution is not optimal and we must choose x_1 as entering variable. However, the coefficient of x_1 in both the constraint equations (as well as in the objective equation) is *negative.* This means that we can increase x_1 without limit, because any increase in x_1 will *increase* the values of the variables x_3 and x_2, and so the solution will remain feasible for all positive values of x_1. Thus we can increase z without limit by increasing x_1, and the problem has no finite optimal solution. □

The problem illustrated by Example 2.13 arises in the simplex algorithm when we have a valid candidate for an entering variable that has no corresponding leaving variable. Thus, if we arrive at a tableau in which a variable has *a negative coefficient in every row, including the objective row,* then we can conclude that the problem has no finite optimal solution and terminate the algorithm.

Multiple optimal solutions

We saw in Chapter 1, Example 1.4, that with the objective function $z = 6x_1 + 4x_1$, the Boatyard problem has multiple optimal solutions. We now explore how we recognize that this situation has arisen when using the simplex algorithm.

We already have the solution of the Boatyard problem with the objective function $z = 6x_1 + 5x_2$. In order to solve this problem with the same constraints, but with the new objective function $z_0 = 6x_1 + 4x_2$, it is not necessary to do all the working over again.

We first express our new objective function z_0 in terms of z. In this case we have

$$z_0 = z - x_2.$$

The final tableau for the solution of the Boatyard problem with z as objective function was obtained at the end of the previous section as:

Basis	*Eqn*	z	x_1	x_2	x_3	x_4	x_5	RS	*upper bound on EV*
-	0	1	0	0	0	19/20	3/20	834	
x_1	1	0	1	0	0	1/5	$-1/10$	44	
x_2	2	0	0	1	0	$-1/20$	3/20	114	
x_3	3	0	0	0	1	$-9/2$	3/2	660	

From this, we see that at the optimal solution,

$$\begin{aligned} z + 19/20x_4 + 3/20x_5 &= 834 \\ x_2 - 1/20x_4 + 3/20x_5 &= 114. \end{aligned}$$

Rearranging these equations so that the terms in the non-basic variables are on the right side of the equations, we have

$$\begin{aligned} z &= 834 - 19/20x_4 - 3/20x_5 \\ x_2 &= 114 + 1/20x_4 - 3/20x_5. \end{aligned}$$

Thus we can express z_0 in terms of the current non-basic variables x_4 and x_5 as

$$\begin{aligned} z_0 &= z - x_2 \\ &= 720 - x_4. \end{aligned}$$

We first note that z_0 is optimal at the current basic feasible solution, because its value cannot be *increased* by increasing either of the non-basic variables x_4 or x_5 from zero. However, the expression for z_0 does not involve the non-basic variable x_5. Hence z_0 remains at its current optimal value of 720 for *any* feasible value of x_5. Now suppose we increase x_5 through positive values, while keeping $x_4 = 0$. We calculate the upper bound on the increase in x_5 in the usual way:

Basic variable	*Equation*	*Upper bound on x_5*
x_1	$x_1 - 1/10x_5 = 44$	No limit on x_5
x_2	$x_2 + 3/20x_5 = 114$	$x_5 \leq 114 \times 20/3 = 760$
x_3	$x_3 + 3/2x_5 = 660$	$x_5 \leq 660 \times 2/3 = 440$

Thus the highest value that x_5 can take is 440. Since $x_5 \geq 0$, we can obtain all optimal solutions by putting $x_5 = r$, where r is any real number such that $0 \leq r \leq 440$. Thus the set of optimal solutions is

$$\{(44 + r/10, 114 - 3r/20) : 0 \leq r \leq 440\},$$

and the maximum value of z_0 is 720.

To sum up: if, when we arrive at an optimal solution of an l.p. problem by the simplex algorithm, we find that *one of the non-basic variables has a zero coefficient in the objective function*, then the problem has *multiple* optimal solutions. In this case, we treat this non-basic variable as if it were to be the next entering variable and calculate the maximum amount by which it can be increased before any of the basic variables become negative. Then it can take any real value in the range between 0 and this upper limit, and the values of the basic variables will vary accordingly. Similar, but more complicated, techniques can be used to find the set of all optimal solutions when more than one non-basic variable has a zero coefficient in the objective function at the optimal solution. However, questions set in the examination for this half unit will not require you to handle more than one non-basic variable with a zero coefficient.

2.7 Exercises

Exercise 2.1
Make a sketch of the feasible region for the l.p. problem of Example 2.4.

(a) Use Result 2.5 to find its optimal solution, as illustrated in Example 2.6;

(b) Use the tableau method for the simplex algorithm to solve this l.p. problem.

Exercise 2.2

Consider the following l.p. problem.

Find $x_1, x_2 \in \mathbf{R}$ to maximize $z = 10x_1 + 8x_2$, subject to

$$\begin{array}{rcrcl} x_1 & + & 4x_2 & \leq & 1600 \\ 3x_1 & + & 2x_2 & \leq & 1800 \\ & & x_1, \quad x_2 & \geq & 0. \end{array}$$

(a) Solve this problem by the simplex algorithm using either the algebraic or the tableau method.

(b) Sketch the feasible region for this problem and identify the basic feasible solution obtained at each iteration with a feasible corner-point solution.

Exercise 2.3

Consider the following l.p. problem.

Find $x_1, x_2 \in \mathbf{R}$ to maximize $z = 2x_1 + 3x_2$, subject to

$$\begin{array}{rcrcl} 2x_1 & + & 5x_2 & \leq & 10 \\ 3x_1 & + & x_2 & \leq & 6 \\ 4x_1 & + & 5x_2 & \leq & 12 \\ & & x_1, \quad x_2 & \geq & 0. \end{array}$$

(a) Solve this problem by the simplex algorithm using the tableau method.

(b) Use the final tableau in your solution to part (a) to find all the optimal solutions to this l.p. problem when the coefficient of x_2 in the objective function is changed from 3 to 2.5.

Exercise 2.4

Sketch the feasible region for the l.p. problem of Example 2.12. Identify the basic feasible solution given by each tableau with a feasible corner-point solution and explain geometrically why the tie for the first leaving variable arises.

Exercise 2.5

Use the tableau method for the simplex algorithm to solve the following l.p. problem.

Find $x_1, x_2, x_3 \in \mathbf{R}$ to maximize $z = 2x_1 + 4x_2 + 3x_3$, subject to

$$\begin{array}{rcrcrcl} x_1 & + & 3x_2 & + & 3x_3 & \leq & 24 \\ 3x_1 & + & 6x_2 & + & 4x_3 & \leq & 90 \\ & & x_1, & & x_2, \quad x_3 & \geq & 0. \end{array}$$