Basic Linear Programming

Brian D Bunday,

Basic Linear Programming

Brian D Bunday,
B.Sc., Ph.D., F.S.S., F.I.M.A.

School of Mathematical Sciences, University of Bradford

Edward Arnold

First published 1984 by
Edward Arnold (Publishers) Ltd
41 Bedford Square, London WC1B 3DQ

Edward Arnold
300 North Charles Street, Baltimore,
MD21201, USA

Edward Arnold (Australia) Ltd.,
80 Waverley Road, Caulfield East,
Victoria 3145, Australia

ISBN 0 7131 3509 3

To Joan, Catherine and Andrew for their forbearance.

Printed in Great Britain by
Spottiswoode Ballantyne Ltd. Colchester and London

Preface

This book presents a course in Linear Programming which will be suitable for a wide range of undergraduate and postgraduate courses. Students of Mathematics, Science and Engineering, as well as students on Management courses which contain a solid component of quantitative methods, will find the text very useful and suited to their needs. Postgraduate students, particularly those on conversion courses in Operational Research or Management Science, will also find the material of value.

The mathematical theory is carefully developed, although there are more rigorous treatments. What the text may lack in rigour is more than compensated for by the main thrust of this book, which is to show, in a vivid manner, the way in which the theoretical ideas are transformed into practical computational procedures capable of implementation on a digital computer. The methods are computer methods and should be treated as such. It is no accident that the main theoretical developments coincided in time with the advent of the computer. Without the latter it would not have been possible to exploit the potential of the theory in the actual solution of real problems.

The text assumes that the reader is familiar with the language BASIC, which is widely used on most microcomputers. The programs have been written in this language. Of course many main-frame computers contain packages which will implement the methods discussed in this book. However, these can be remote and can be treated as a black box and thus used without full understanding. The interactive nature of most microcomputers will allow the student to become intimately involved with the programs. It is not claimed that these are incapable of improvement. Indeed the author would be delighted to hear from readers who feel that they can produce better programs. It is through such close involvement that students will gain a real appreciation of the significance of the theory as well as developing their practical computational and programming techniques. The programs given are robust and practical. They could easily be translated into other languages such as FORTRAN or Pascal if it is desired to use them on other computers.

The modelling aspects of Linear Programming are catered for through the worked examples within the text, and the exercises to be found at the end of each chapter. The reader is strongly advised to test his skills with these problems. Although some of them are necessarily abbreviated and simplified, they do point to the wide variety of situations in which Linear Programming can find application.

A few remarks about BASIC and the way in which it is used in this book are appropriate. The programs have been written in such a way that they should run with the minimum of fuss on any microcomputer. Thus no attempt has been made to include colour, sound or high resolution graphics.

In the assignment statements LET has been omitted. On some computers it is

obligatory and so would have to be inserted. The THEN has been included in IF . . . THEN GOTO statements, although on some computers either the THEN or the GOTO may be omitted. No use has been made of the IF . . . THEN . . . ELSE or the REPEAT . . . UNTIL . . . facilities as these are not universally available. It is assumed that arrays start at a zero suffix. On machines whose arrays start with suffix 1 some changes would be necessary. One way which should always work is to increase *all* of the suffices including those in the DIM statements by 1. Thus DIM A(M) becomes DIM A(M + 1); B(K, L) becomes B(K + 1, L + 1). However, in particular cases the reader might well find more elegant modifications. The numerical answers given are those obtained on a PET. Some machines which store numbers to a different accuracy will not precisely reproduce the results given here although the differences should only occur in the least significant digits.

It is a pleasure to thank friends, colleagues and students who have contributed to this work. The students have been willing guinea-pigs for many of the problems. Some of these have been taken from examination papers set at the University of Bradford. I am grateful to the university for permission to use these questions, I would particularly mention Dr R. E. Scraton who, besides improving my Numerical Analysis, has allowed me to use his formatting procedure in some of the programs. I am indebted to Mr C Mack for many beneficial and illuminating discussions on the ideas of his method for the Assignment Problem, and the way to program the method. Last but by no means least I must thank Mrs Valerie Hunter who transformed my scruffy manuscript into a neat and tidy typescript.

BRIAN BUNDAY
1984

Contents

1

Fundamental Ideas

1.1 Introduction

Linear programming has proved itself to be an extremely powerful technique in the solution of certain problems which arise in the field of Operational Research. The word 'programming' means planning in this context, and gives a clue as to the type of application. The ideas were first developed during World War II in connection with finding optimal strategies for conducting the war effort. Since that time they have found wide application in industry, commerce and Government Service, the latter at both local and national level. The methods are of value in the formulation and solution of many (though not all), problems concerned with the efficient use of limited resources.

Some of the ideas can be illustrated from consideration of the following simplified version of a real production scheduling problem.

Example 1

A firm produces self-assembly bookshelf kits in two models, A and B. Production of the kits is limited by the availability of raw material (high quality board) and machine processing time. Each unit of A requires 3 m^2 of board and each unit of B requires 4 m^2 of board. The firm can obtain up to 1700 m^2 of board each week from its suppliers. Each unit of A needs 12 minutes of machine time and each unit of B needs 30 minutes of machine time. Each week a total of 160 machine hours is available. If the profit on each A unit is \$2, and on each B unit is \$4, how many units of each model should the firm plan to produce each week?

In order to formulate this problem in mathematical form let the weekly production of A be x_1 units, and of B, x_2 units. The problem is then to find the *best* values for x_1 and x_2. A fairly obvious way to interpret best for this problem is *so as to maximise profit each week.* The weekly profit can be expressed as

$$P = 2x_1 + 4x_2. \tag{1.1}$$

The firm will achieve its objective by maximising the **objective function** $P = 2x_1 + 4x_2$.

Classical optimisation says that an optimum of a function will occur *either* where the derivatives are zero *or* on the boundary of the domain space. To consider the derivatives only is inadequate.

$$\frac{\partial P}{\partial x_1} = 2 \quad \text{and} \quad \frac{\partial P}{\partial x_2} = 4$$

and it is not possible to make these derivatives zero by choice of x_1 and x_2. Indeed the

way to increase P is to go on increasing x_1 and x_2. But, and this is the essence of the problem, x_1 and x_2 cannot be increased without limit. Their possible values are restricted by physical considerations and by the **constraints** on raw material and machine time.

Because x_1 and x_2 represent the number of units of each model produced each week, it is clear that they cannot be negative:

$$\text{i.e.} \quad x_1 \geqslant 0, x_2 \geqslant 0. \tag{1.2}$$

The constraints on availability of board and machine time can be put in the form:

$$\begin{aligned} &\text{Board:} && 3x_1 + 4x_2 \leqslant 1700 \\ &\text{Machine hours:} && \tfrac{1}{3}x_1 + \tfrac{1}{2}x_2 \leqslant 160 \end{aligned}$$

$$\left.\begin{aligned} \text{i.e.} \quad 3x_1 + 4x_2 &\leqslant 1700 \\ 2x_1 + 5x_2 &\leqslant 1600 \end{aligned}\right\} \tag{1.3}$$

Thus the problem is to find values of x_1 and x_2 which satisfy the **non-negativity** conditions (1.2) and the inequality constraints (1.3) so as to maximise $P = 2x_1 + 4x_2$.

This is a typical linear programming problem. The objective function which is to be maximised is a **linear function** of the variables. The constraints on these variables are also linear. Indeed for this particular two-dimensional problem they can be represented graphically by the lines shown in Fig. 1.1. The non-negativity conditions restrict the variables to the positive quadrant. The constraints are represented by the lines:

$$3x_1 + 4x_2 = 1700$$

$$2x_1 + 5x_2 = 1600.$$

The arrow on each constraint in Fig. 1.1 indicates the side of the line on which the constraint is satisfied. The directions on each arrow can easily be determined by considering whether the origin (0, 0) satisfies the constraint. The shaded area OABC which contains all points (x_1, x_2) satisfying equations (1.2) and (1.3) is called the **feasible region**. Points within and on the boundary of this region represent **feasible solutions** of the constraints. There are plenty of feasible solutions. The problem is to find the one (or might there be more than one?) which maximises P.

The (dashed) lines (a) $2x_1 + 4x_2 = 0$, (b) $2x_1 + 4x_2 = 800$, are shown in Fig. 1.1. These lines are parallel and represent two **contour** lines of the function P with values 0 and 800 respectively. It is clear that the value of P increases as the contour lines move further away from the origin in the positive quadrant. Indeed the vector with components $\begin{pmatrix} \dfrac{\partial P}{\partial x_1} \\ \dfrac{\partial P}{\partial x_2} \end{pmatrix}$ i.e. the vector with components $\begin{pmatrix} 2 \\ 4 \end{pmatrix}$ points in the direction of increasing P and this direction is perpendicular to these parallel lines, away from 0, as shown.

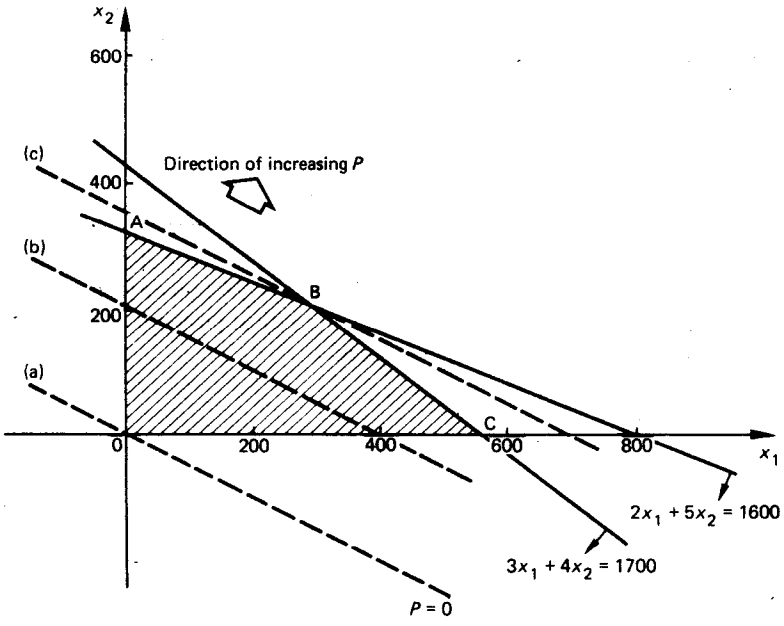

Fig. 1.1

The contour line with the highest value of P which has at least one point in common with the feasible region is the line (c), which passes through the vertex B, and on which P has the value 1400. The point B, $x_1 = 300$, $x_2 = 200$ corresponds to the optimal solution for the problem. These values can be obtained as the solution of the equations

$$3x_1 + 4x_2 = 1700$$

$$2x_1 + 5x_2 = 1600.$$

Of course the maximum profit is then $2 \times 300 + 4 \times 200 = 1400$. In the optimal solution both constraints are satisfied as equalities, and this can be interpreted as meaning that all the available raw material and machine time is utilised.

It is clear that this problem could be extended. There could be three or more models and a corresponding number of non-negative variables. There could be additional constraints representing market capacity, limitations on packaging facilities, etc. The problem would still be one of maximising a *linear* function of several *non-negative* variables which are subject to *linear* inequality constraints.

The general linear programming problem is that of maximising (or it could be minimising) a linear function

$$z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{1.4}$$

of n real variables $x_1, x_2, \ldots, x_n$ satisfying the non-negativity conditions

$$x_1 \geqslant 0, x_2 \geqslant 0, \ldots, x_n \geqslant 0 \tag{1.5}$$

and m linear constraints

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leqslant = \geqslant b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leqslant = \geqslant b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leqslant = \geqslant b_m \end{aligned}\right\} \tag{1.6}$$

The constraints can be a mixture of the '$\leqslant$', '$=$' or '$\geqslant$' variety. The aim will be to maximise the objective or perhaps to minimise the objective (if it represents a cost for example). The values of the b_i, c_j, a_{ij} are assumed to be known constants. They will often have a physical interpretation in terms of a practical problem as in Example 1.

In matrix notation the problem can be written

Maximise (or minimise)

$$z = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}_0 \tag{1.7}$$

where
$$\boldsymbol{x}_0 \geqslant \boldsymbol{0} \tag{1.8}$$

and
$$\boldsymbol{A}_0\boldsymbol{x}_0 \leqslant = \geqslant \boldsymbol{b} \tag{1.9}$$

where $\boldsymbol{x}_0 = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ is an $n \times 1$ column vector,

$\boldsymbol{c}^{\mathrm{T}} = (c_1, c_2, \ldots, c_n)$ is a $1 \times n$ row vector,

$\boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$ is an $m \times 1$ column vector, which can be assumed non-negative,

and $\boldsymbol{A}_0 = (a_{ij})$ is an $m \times n$ matrix.

The suffix 0 on $\boldsymbol{x}_0$ and $\boldsymbol{A}_0$ is to be taken to mean 'original'. The point will be taken up and made clear in Section 1.3.

1.2 Graphical Solution of Two-Dimensional Problems

The two-dimensional example of the previous section served to show how a linear programming problem could arise from a practical situation and also showed a graphical method of solution. Through a study of a few other such examples which are simple enough for us to see 'what is going on', it will be possible to bring out certain general features of linear programming problems whose exploitation can lead to a systematic solution procedure.

Example 1

Minimise
$$z = -3x_1 - 4x_2$$

where $x_1, x_2 \geqslant 0$ and

$$\begin{aligned} x_1 + \ x_2 &\leqslant 20 \\ -x_1 + 4x_2 &\leqslant 20 \\ x_1 \qquad &\geqslant 10 \\ x_2 &\geqslant 5 \end{aligned}$$

PQRS is the feasible region in Fig. 1.2. The last two constraints subsume the non-negativity conditions. z *decreases* in the direction

$$-\begin{pmatrix} \dfrac{\partial z}{\partial x_1} \\ \dfrac{\partial z}{\partial x_2} \end{pmatrix}, \quad \text{i.e.} \begin{pmatrix} 3 \\ 4 \end{pmatrix}.$$

The minimum value of z is -68 and arises at R(12, 8). Note that as in the example of the last section the minimum occurs at a vertex of the feasible region. The optimal solution is $x_1 = 12$, $x_2 = 8$ with the minimum of z at -68.

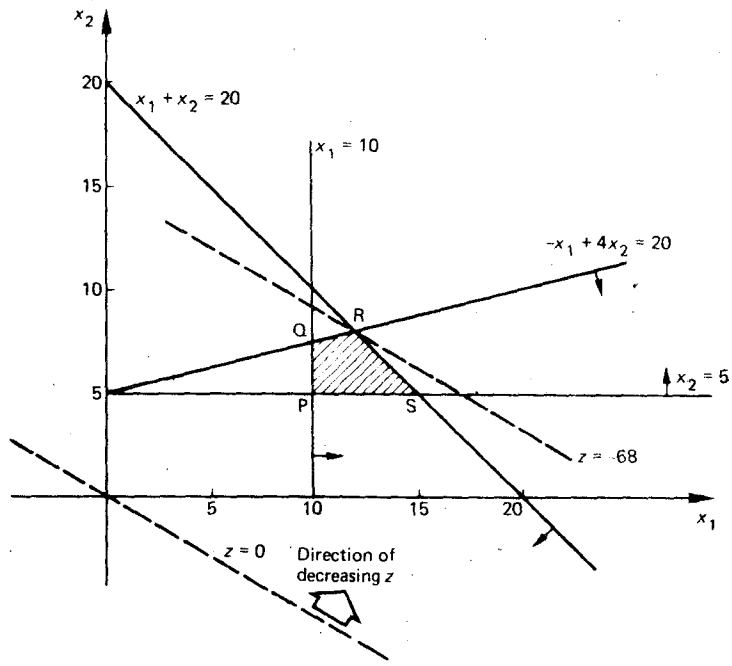

Fig. 1.2

Sometimes there is more than one optimal solution.

Example 2

Minimise $z = -6x_1 - 2x_2$

subject to $x_1, x_2 \geqslant 0$

$$2x_1 + 4x_2 \leqslant 9$$
$$3x_1 + x_2 \leqslant 6$$

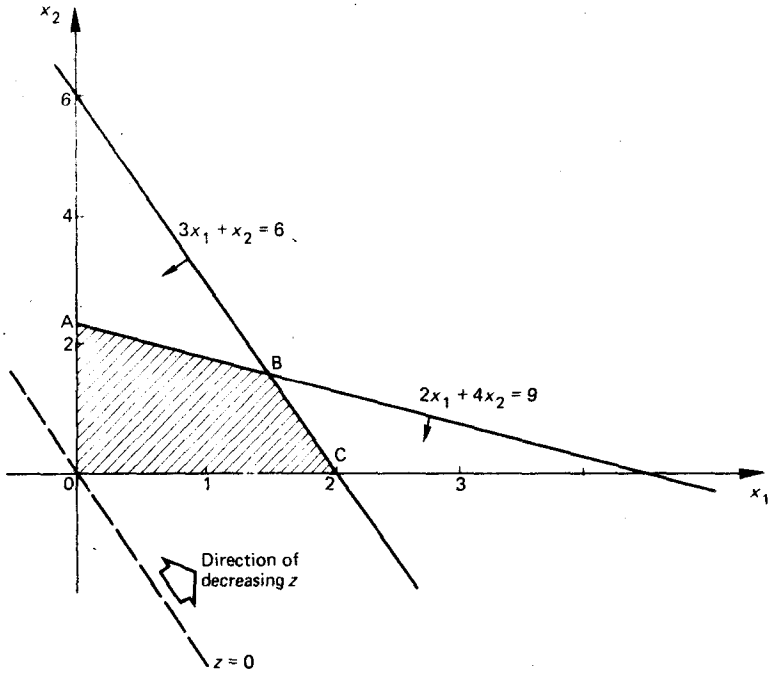

Fig. 1.3

OABC in Fig. 1.3 shows the feasible region.

$\partial z/\partial x_1 = -6$, $\partial z/\partial x_2 = -2$ and so the vector $\begin{pmatrix} 6 \\ 2 \end{pmatrix}$ points in the direction of *decreasing* z. Any point on BC represents an optimal solution. In particular the vertices B($1\frac{1}{2}$, $1\frac{1}{2}$) and C(2, 0) represent optimal solutions corresponding to the (one) minimum value of $z = -12$.

Any point on BC can be represented as

$$\theta(1\tfrac{1}{2}, 1\tfrac{1}{2}) + (1 - \theta)(2, 0) = (2 - \tfrac{1}{2}\theta, 1\tfrac{1}{2}\theta)$$

for $0 \leqslant \theta \leqslant 1$.

For each point the value of z is $-6(2 - \frac{1}{2}\theta) - 2(1\frac{1}{2}\theta) = -12$. There is only *one* minimum value of z.

Sometimes the solution is unbounded.

Example 3

Maximise $z = x_1 + x_2$

subject to $x_1 \geqslant 0, x_2 \geqslant 0$

$$x_1 - x_2 \geqslant 1$$

$$x_2 \leqslant 2.$$

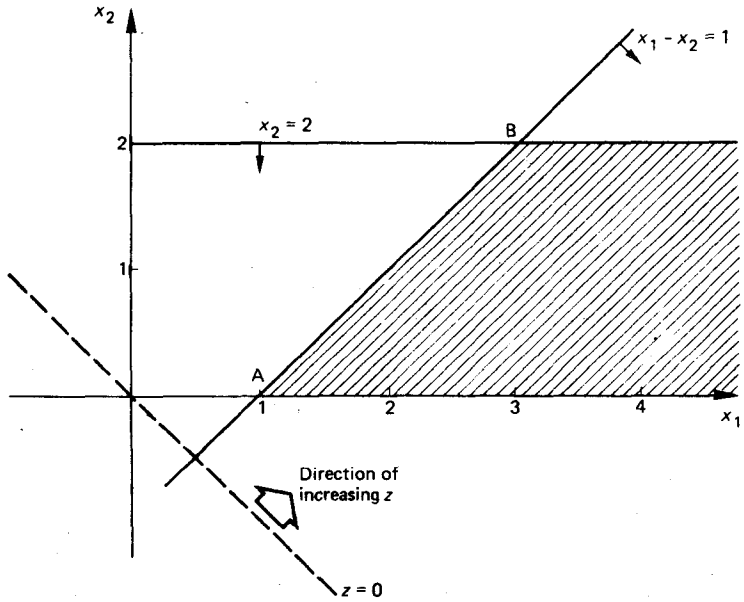

Fig. 1.4

The feasible region shown in Fig. 1.4 is unbounded in the direction in which z increases. There is no finite point in the feasible region at which z attains a maximum. The solution is unbounded and so is the maximum value z. It is possible in some problems for an unbounded solution to occur with a finite maximum for the objective. This would be the case, for example, had the problem been to maximise $z' = x_2$ subject to the constraints.

Of course, had the problem been to *minimise* $z = x_1 + x_2$ subject to the above constraints, there is one finite minimum of z (min) $= 1$ at A (again a vertex of the feasible region) where $x_1 = 1, x_2 = 0$.

Sometimes there is no solution at all because a feasible region does not exist.

Example 4

Minimise $z = 2x_1 + 3x_2$

subject to $x_1, x_2 \geqslant 0$

$$x_1 + x_2 \geqslant 10$$

$$3x_1 + 5x_2 \leqslant 15$$

The constraints are contradictory and have no feasible solution (see Fig. 1.5).

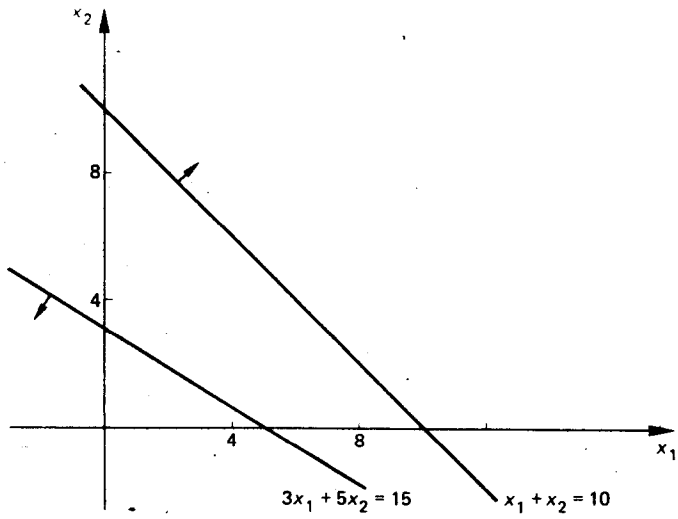

Fig. 1.5

One or two fairly general features of linear programming (L.P.) problems can be deduced from the examples already considered. The first is that the feasible region is always a convex polygon. Even in the case where it was unbounded it was convex on its closed aspect. The second is that the optimum solution always occurs at a vertex of the feasible region. In Example 2 where there were several optimal solutions both the vertices B and C corresponded to optimal solutions.

We shall see that we can generalise these results. First we show that we can put all L.P. problems in a standard form.

1.3 A Standard Form for Linear Programming Problems

It may appear that L.P. problems can take on a variety of forms with the constraints a mixture of '$\geqslant$', '$=$' or '$\leqslant$' type. They can all be put into a standard form in which the objective function is to be **minimised** and all constraints take the form of **equations** in **non-negative variables.**

Problems which do not initially conform to this standard form can be brought to this form quite simply.

(a) Maximising the objective function $z = c_1x_1 + \ldots + c_nx_n$ is equivalent to minimising the objective function

$$z' = -c_1x_1 - c_2x_2 - \ldots - c_nx_n.$$

(b) Inequality constraints.

The constraint $3x_1 + 2x_2 - x_3 \leqslant 6$ can be put in equation form as

$$3x_1 + 2x_2 - x_3 + x_4 = 6$$

where the **slack variable** x_4 is non-negative.

The constraint $x_1 - x_2 + 3x_3 \geqslant 10$ can be put in equation form as

$$x_1 - x_2 + 3x_3 - x_5 = 10$$

where the **slack variable** x_5 is non-negative.

(c) Non-negative variables.
If a particular variable x_k can take on any value then we can write $x_k = x'_k - x''_k$ where $x'_k \geqslant 0$ and $x''_k \geqslant 0$.

Thus bringing a problem into the standard form might involve the introduction of additional variables, (which are still non-negative) into the problem.

Thus following on from equations (1.7), (1.8), (1.9) our most general L.P. problem can be put in the form

minimise $$z = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{1.10}$$

where $$\boldsymbol{x} \geqslant \boldsymbol{0} \tag{1.11}$$

and $$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \text{with } \boldsymbol{b} > \boldsymbol{0}. \tag{1.12}$$

If this problem did indeed arise from that given earlier then $\mathbf{x}$ will contain the slack variables as well as the *original* variables and A will contain the coefficients of the slack variables as well as the original coefficients.

Thus Example 1 of Section 1.2 can be put in the form

minimise $$z = -3x_1 - 4x_2$$

subject to the constraints

$$\begin{aligned} x_1 \quad - x_3 \qquad\qquad &= 10 \\ x_2 \quad - x_4 \qquad &= 5 \\ x_1 + \; x_2 \qquad + x_5 \quad &= 20 \\ -x_1 + 4x_2 \qquad\quad + x_6 &= 20 \end{aligned}$$

and $x_i \geqslant 0$, $i = 1, 2, \ldots, 6$.

Example 1 of Section 1.1 can be put in the form

minimise $$z = -2x_1 - 4x_2$$

subject to the constraints

$$\begin{aligned} 3x_1 + 4x_2 + x_3 \quad &= 1700 \\ 2x_1 + 5x_2 \quad + x_4 &= 1600 \end{aligned}$$

and $x_i \geqslant 0$, $i = 1, \ldots, 4$.

In matrix form the constraints can be written

$$\begin{pmatrix} 3 & 4 & 1 & 0 \\ 2 & 5 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1700 \\ 1600 \end{pmatrix}$$

They consist of 2 equations in 4 unknowns. Any non-negative solution of these constraints is a **feasible solution.**

Of course with 2 equations in 4 unknowns we can hope to get a solution (though not necessarily a feasible solution) by giving two of the variables arbitrary values and solving for the remaining two. Of particular interest are solutions of this type which arise by setting two of the variables to zero. Such a solution, if unique, is referred to as a **basic** solution. If it is also feasible it is a **basic feasible solution (b.f.s.).** For the general linear programming problem with say m linear equation constraints in n variables ($m < n$) a **basic solution** of the constraints is obtained by setting $(n - m)$ of the variables to zero and solving the m equations which result for the remaining m variables, provided these equations have a **unique** solution. The variables put equal to zero are called the **non-basic variables (n.b.v.).** The others are the **basic** variables and form a **basis.**

For the problem just considered we can select the two non-basic variables in $\binom{4}{2} = 6$ ways. The basic solutions are easily seen to be given by

	x_1	x_2	x_3	x_4	
1	0	0	1700	1600	O
2	0	425	0	−525	
3	0	320	420	0	A
4	$566\frac{2}{3}$	0	0	$466\frac{2}{3}$	C
5	800	0	−700	0	
6	300	200	0	0	B

corresponding to non-basic variables (x_1, x_2), (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4), (x_3, x_4). Of these 6 basic solutions, only 4 are also feasible, and it will be seen that these solutions correspond to the vertices of the feasible region in Fig. 1.1 with correspondence as indicated.

In three dimensions the linear constraints take the form of planes (instead of lines). The feasible region, instead of being a convex polygon, is a convex polyhedron. An optimal solution to the problem will correspond to a vertex of this polyhedron since the contours of the objective function will be planes instead of lines, and the plane corresponding to the least value will generally have just one point in common with the feasible region, and this will be a vertex of the convex polyhedron, and will correspond to an optimal solution of the problem.

We shall see that this particular type of result is quite general. The basic feasible solutions of a system of m equations in n unknowns correspond to the vertices of the feasible region. Further an optimal solution, if it exists, corresponds to a basic feasible solution, and hence a vertex of the feasible region.

1.4 Some *n*-Dimensional Geometry

Before establishing the results just mentioned it is necessary to generalise some geometric concepts from two dimensions to n dimensions. The two-dimensional graphical solution method used in Section 1.2 is quite general. However, in n dimensions our intuition and ability to visualise the situation is not so clear. We need algebraic methods to do the geometry.

We first define some terms which allow the concept of a convex set to be understood.