

1 An introduction to linear programming

1.1 Introduction

In the Second World War there were many problems connected with optimal deployment of troops, aircraft, equipment and so on.

Similar problems appeared after the war in both government and industry.

Linear programming was developed to solve such problems.

These problems commonly involve a large number of inequalities rather than equations, so that classical methods of solution fail.

The development of these techniques in conjunction with the rise of computers which could solve large problems meant that linear programming became very successful.

1.2 The basic problem

The general form of a linear programming problem is one of maximising or minimising some linear function

$$z = c_1x_1 + \dots + c_nx_n$$

of n real quantities x_i subject to m linear constraints; z is called the objective function (and might be a total cost or profit).

A constraint is either an equation or an inequality, so we have

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq, =, \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq, =, \leq b_2$$

$\vdots$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq, =, \leq b_m$$

where the a_{ij} s and b_j are given constants.

It is also usually the case that $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$.

The *simplex algorithm* can solve all such problems with a finite number of steps (though for specific problems it is not the most efficient algorithm). We will consider the simplex algorithm in detail later.

1.3 Graphical methods

Consider the following linear programming problem.

Example 1

Maximise the sum $x_1 + x_2$ subject to the constraints

$$x_1 \geq 0, x_2 \geq 0 \text{ and}$$

$$x_1 + 2x_2 \leq 4$$

$$4x_1 + 2x_2 \leq 12$$

$$-x_1 + x_2 \leq 1$$

We have five constraints and two unknowns in this problem.

Any solution that satisfies all of the constraints is called a *feasible solution*.

Given that there are only two unknowns, the best approach is to draw a graph of the problem.

In Figure 1 we want to find (x_1, x_2) which maximises the objective function $x_1 + x_2$ over the feasible values.

FIGURE 1

$x_1 + x_2$ is constant on all lines of slope -1 (e.g. $x_1 + x_2 = 1$). It increases as we move the line further from the origin in the positive direction.

We thus find the line furthest from the origin which still touches the allowable region.

This occurs at the intersection of the lines

$$x_1 + 2x_2 = 4 \text{ and}$$

$$4x_1 + 2x_2 = 12.$$

This has solution $x_1 = 8/3, x_2 = 2/3$, giving the value of the objective function as $10/3$.

This approach works for all linear programming problems with two variables.

Assuming that the region is bounded, then the solution always occurs at a corner on the boundary of the region.

Sometimes the objective function is maximised along a whole edge (but it will still also be a maximum at the two corner points on that edge).

e.g. if the objective function $x_1 + x_2$ in Example 1 was replaced by $x_1 + 2x_2$, it coincides with a boundary.

If the region is unbounded, it is possible to have an unbounded solution to the objective function. This is true for the following example

Example 2

Maximise the sum $x_1 + x_2$ subject to the constraints

$$x_1 \geq 0, x_2 \geq 0 \text{ and}$$

$$x_1 - 3x_2 \geq -3$$

$$-x_1 + x_2 \leq 1$$

FIGURE 2

It is also possible to have a finite solution to the objective function for an unbounded region, if the objective function coincides with one of the boundaries of the region.

Thus if the objective function $x_1 + x_2$ in Example 2 was replaced by $x_1 - 3x_2$, the maximum of the objective function is then -3, along this whole (infinite) boundary.

Example 3

Maximize $z = 2x_1 + 9x_2$

subject to the constraints

$$4x_1 + 5x_2 \leq 20$$

$$3x_1 + 7x_2 \leq 21$$

1.4 The simplex algorithm in two dimensions

We will apply the simplex method to a modified version of the problem in Example 1 (with the third constraint removed).

Example 4

The problem is to maximise the sum $x_1 + x_2$ subject to the constraints

$$x_1 \geq 0, x_2 \geq 0 \text{ and}$$

$$x_1 + 2x_2 \leq 4$$

$$4x_1 + 2x_2 \leq 12$$

The last two inequalities are equivalent to

$$x_1 + 2x_2 + x_3 = 4$$

$$4x_1 + 2x_2 + x_4 = 12$$

where we have introduced two new variables x_3, x_4 and

$$x_3 \geq 0, x_4 \geq 0$$

x_3 and x_4 are known as *slack variables*.

One solution of the above system is $x_1 = 0, x_2 = 0, x_3 = 4, x_4 = 12$ which yields $z = 0$.

Note that in the above solution the number of non-zero variables is 2, the same as the number of main constraints. In general this is a property of the simplex method, that the number of non-zero solutions is less than or equal to the number of main constraints.

Each iteration of the simplex method typically makes one of the values x_i that had been zero become non-zero, and one that had been non-zero become zero.

Step 1:

Let us leave $x_2 = 0$ and try to increase x_1 by as much as we can.

We have

$$x_1 + 2 \times 0 + x_3 = 4$$

$$4x_1 + 2 \times 0 + x_4 = 12$$

Thus the second equation is the most restrictive, and we can increase x_1 to 3, whilst setting $x_4 = 0$. From the first equation we obtain $x_3 = 1$.

Alternatively writing the two equations by giving the non-zero terms x_1 and x_3 in terms of the zero ones x_2 and x_4 , we obtain

$$4x_1 = 12 - 2x_2 - x_4 \Rightarrow x_1 = 3 - x_2/2 - x_4/4 \quad (1)$$

$$x_3 = 4 - 2x_2 - x_1 = 4 - 2x_2 - (3 - x_2/2 - x_4/4) = 1 - 3x_2/2 + x_4/4 \quad (2)$$

Similarly we can write the objective function in the same way as

$$z = x_1 + x_2 = 3 - x_2/2 - x_4/4 + x_2 = 3 + x_2/2 - x_4/4 \quad (3)$$

At the point $x_1 = 3, x_2 = 0$ the objective function is equal to 3.

Step 2:

We have increased z , but can we increase it further? The objective function (3) above indicates that we can increase the value by increasing x_2 .

In (1) replacing x_1 would allow x_2 to go up to 6, but in (2) replacing x_3 is more restrictive, allowing it to go to $2/3$. Thus we replace x_3 , setting it to be equal to zero, and setting $x_2 = 2/3$.

Writing the restrictions in terms of x_1 and x_2 , we now obtain
 $3x_2/2 = -x_3 + 1 + x_4/4 \Rightarrow x_2 = 2/3 - 2x_3/3 + x_4/6$ and
 $x_1 = 3 - x_4/4 - (2/3 - 2x_3/3 + x_4/6)/2 = 8/3 + x_3/3 - x_4/3$.

This gives the objective function z as

$$z = 3 + (2/3 - 2x_3/3 + x_4/6)/2 - x_4/4 = 10/3 - x_3/3 - x_4/6 \quad (4)$$

Thus we have the value of the objective function as $10/3$.

Can we increase this further? From (4) we can see that the coefficients of both x_3 and x_4 are negative, so the answer is no. Thus the solution of the problem is:

$x_1 = 8/3, x_2 = 2/3$ with $z = 10/3$ as in Example 1 (recall that the constraint that we omitted from Example 4 had no effect in Example 1).

Example 5

Minimize $z = 20x_1 + 21x_2$

subject to the constraints

$$4x_1 + 3x_2 \geq 2$$

$$5x_1 + 7x_2 \geq 9$$

Notice the relationship between this example and Example 3.

2 The dual program and some simplex algorithm preliminaries

2.1 Pivoting

Example 6

Consider the following system of equations

$$3y_1 + 2y_2(+0y_3) = s_1$$

$$y_1 - 3y_2 + 3y_3 = s_2$$

$$5y_1 + y_2 + y_3 = s_3$$

This expresses the *dependent variables* s_1, s_2 and s_3 in terms of the *independent variables* y_1, y_2 and y_3 .

If we wish to obtain y_2, s_2 and s_3 in terms of y_1, s_1 and y_3 , we use the first equation to see that

$$y_2 = s_1/2 - 3y_1/2$$

and substitute into the other two equations to give

$$y_1 - 3(s_1/2 - 3y_1/2) + 3y_3 = s_2$$

$$5y_1 + (s_1/2 - 3y_1/2) + y_3 = s_3$$

This gives us the three equations

$$-1.5y_1 + 0.5s_1(+0y_3) = y_2$$

$$5.5y_1 - 1.5s_1 + 3y_3 = s_2$$

$$3.5y_1 + 0.5s_1 + y_3 = s_3$$

This is the type of operation that we carried out above when considering the simplex method in two dimensions.

This is an example of considering the system $\mathbf{y}^T A = \mathbf{s}^T$ where $\mathbf{y}^T = (y_1, \dots, y_m)$, $\mathbf{s}^T = (s_1, \dots, s_n)$ and

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

This is equivalent to

$$y_1 a_{11} + y_2 a_{21} + \dots y_m a_{m1} = s_1$$

$\vdots$

$$y_1 a_{1j} + y_2 a_{2j} + \dots y_m a_{mj} = s_j$$

$\vdots$

$$y_1 a_{1n} + y_2 a_{2n} + \dots y_m a_{mn} = s_n$$

As above s_j are the dependent variables, and y_i the independent variables. To exchange one of the dependent variables s_j for one of the independent ones y_i , provided that $a_{ij} \neq 0$, we take the j th equation above and express y_i in terms of the other variables, giving

$$y_i = \frac{1}{a_{ij}}(-a_{1j}y_1 - \dots - a_{i-1,j}y_{i-1} + s_j - a_{i+1,j}y_{i+1} - \dots a_{mj}y_m)$$

Substituting into the other equations, we obtain

$$\left(a_{1k} - \frac{a_{ik}a_{1j}}{a_{ij}}\right)y_1 + \dots + s_j \left(\frac{a_{ik}}{a_{ij}}\right) + \dots + \left(a_{mk} - \frac{a_{ik}a_{mj}}{a_{ij}}\right)y_m = s_k$$

We thus get the following system

$$\begin{aligned}
y_1 \hat{a}_{11} + \dots + s_j \hat{a}_{i1} + \dots y_m \hat{a}_{m1} &= s_1 \\
\vdots \\
y_1 \hat{a}_{1j} + \dots + s_j \hat{a}_{ij} + \dots y_m \hat{a}_{mj} &= y_i \\
\vdots \\
y_1 \hat{a}_{1n} + \dots + s_j \hat{a}_{in} + \dots y_m \hat{a}_{mn} &= s_n
\end{aligned}$$

where

$$\hat{a}_{ij} = \frac{1}{a_{ij}}, \quad \hat{a}_{hj} = -\frac{a_{hj}}{a_{ij}} \quad h \neq i$$

$$\hat{a}_{ik} = \frac{a_{ik}}{a_{ij}} \quad k \neq j, \quad \hat{a}_{hk} = a_{hk} - \frac{a_{ik}a_{hj}}{a_{ij}} \quad k \neq j, h \neq i$$

We summarise the original equations in the following tabular form

	s_1	$\dots$	s_j	$\dots$	s_n
y_1	a_{11}	$\dots$	a_{1j}	$\dots$	a_{1n}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
y_i	a_{i1}	$\dots$	a_{ij}	$\dots$	a_{in}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
y_m	a_{m1}	$\dots$	a_{mj}	$\dots$	a_{mn}

We represent the change we have made as follows

$$\begin{array}{c|ccccc} & s_1 & \dots & s_j & \dots & s_n \\ \hline y_1 & a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i & a_{i1} & \dots & [\mathbf{a}_{ij}] & \dots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m & a_{m1} & \dots & a_{mj} & \dots & a_{mn} \end{array} \rightarrow \begin{array}{c|ccccc} & s_1 & \dots & y_i & \dots & s_n \\ \hline y_1 & \hat{a}_{11} & \dots & \hat{a}_{1j} & \dots & \hat{a}_{1n} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_j & \hat{a}_{i1} & \dots & \hat{a}_{ij} & \dots & \hat{a}_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m & \hat{a}_{m1} & \dots & \hat{a}_{mj} & \dots & \hat{a}_{mn} \end{array}$$

Note that the pivoting rules can be summarised in the following manner.

$$\begin{array}{|c|c|} \hline [\mathbf{p}] & \mathbf{r} \\ \hline \mathbf{c} & \mathbf{q} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|} \hline 1/\mathbf{p} & \mathbf{r}/\mathbf{p} \\ \hline -\mathbf{c}/\mathbf{p} & \mathbf{q}-\mathbf{rc}/\mathbf{p} \\ \hline \end{array}$$

Example 6 cont.

We will solve the equations

$$3y_1 + 2y_2(+0y_3) = s_1$$

$$y_1 - 3y_2 + 3y_3 = s_2$$

$$5y_1 + y_2 + y_3 = s_3$$

i.e. find the y_i s in terms of the s_j s.

We use repeated applications of the pivoting mechanism.

$$\begin{array}{c|ccc} & s_1 & s_2 & s_3 \\ \hline y_1 & 3 & 1 & 5 \\ y_2 & [\mathbf{2}] & -3 & 1 \\ y_3 & 0 & 3 & 1 \end{array} \rightarrow \begin{array}{c|ccc} & y_2 & s_1 & s_3 \\ \hline y_1 & -3/2 & 11/2 & 7/2 \\ s_1 & 1/2 & -3/2 & 1/2 \\ y_3 & 0 & 3 & 1 \end{array}$$

gives the application already carried out.

Continuing, we obtain

$$\begin{array}{c|ccc} & y_2 & s_2 & s_3 \\ \hline y_1 & -3/2 & 11/2 & 7/2 \\ s_1 & 1/2 & -3/2 & 1/2 \\ y_3 & 0 & 3 & [\mathbf{1}] \end{array} \rightarrow \begin{array}{c|ccc} & y_2 & s_2 & y_3 \\ \hline y_1 & -3/2 & [-\mathbf{5}] & -7/2 \\ s_1 & 1/2 & -3 & -1/2 \\ s_3 & 0 & 3 & 1 \end{array} \rightarrow \begin{array}{c|ccc} & y_2 & y_1 & y_3 \\ \hline s_2 & 0.3 & -0.2 & 0.7 \\ s_1 & 1.4 & -0.6 & 1.6 \\ s_3 & -0.9 & 0.6 & -1.1 \end{array}$$

Thus

$$y_1 = -0.6s_1 - 0.2s_2 + 0.6s_3$$

$$y_2 = 1.4s_1 + 0.3s_2 - 0.9s_3$$

$$y_3 = 1.6s_1 + 0.7s_2 - 1.1s_3$$

Example 7

Solve the system of equations $\mathbf{y}^T A = \mathbf{s}^T$ given by

$$\begin{array}{c|cccc} & s_1 & s_2 & s_3 & s_4 \\ \hline y_1 & 1 & 1 & 0 & 2 \\ y_2 & 0 & 2 & -1 & 1 \\ y_3 & -1 & 1 & 0 & 3 \\ y_4 & 1 & -2 & 2 & 1 \end{array}$$

2.2 Linear programming problems in standard form

Consider the following linear programming problem.

Find a vector

$\mathbf{x} = (x_1, \dots, x_n)$, to maximise

$$\mathbf{c}^T \mathbf{x} = c_1x_1 + \dots + c_nx_n$$

subject to the constraints

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$\vdots$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

(equivalently written $A\mathbf{x} \leq \mathbf{b}$)

and $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$

(equivalently $\mathbf{x} \geq 0$).

This is the *standard maximum problem*.

Example 1: revisited

Maximise the sum $x_1 + x_2$ subject to the constraints

$x_1 \geq 0, x_2 \geq 0$ and

$$x_1 + 2x_2 \leq 4$$

$$4x_1 + 2x_2 \leq 12$$

$$-x_1 + x_2 \leq 1$$

We can see that Example 1 is an example of the standard maximum problem.

Now consider the following linear programming problem.

Find a vector

$\mathbf{y} = (y_1, \dots, y_m)$, to minimise

$$\mathbf{y}^T \mathbf{b} = y_1 b_1 + \dots + y_m b_m$$

subject to the constraints

$$y_1 a_{11} + y_2 a_{21} + \dots + y_m a_{m1} \geq c_1$$

$$y_1 a_{12} + y_2 a_{22} + \dots + y_m a_{m2} \geq c_2$$

$\vdots$

$$y_1 a_{1n} + y_2 a_{2n} + \dots + y_m a_{mn} \geq c_n$$

(equivalently written $\mathbf{y}^T A \geq \mathbf{c}^T$)

and $y_1 \geq 0, y_2 \geq 0, \dots, y_m \geq 0$

(equivalently $\mathbf{y} \geq 0$).

This is the *standard minimum problem*.

Example 5: revisited

Minimize $z = 20x_1 + 21x_2$

subject to the constraints

$$4x_1 + 3x_2 \geq 2$$

$$5x_1 + 7x_2 \geq 9$$

We can see that Example 5 is an example of the standard minimum problem.

Example 8: The transportation problem

There are I ports, or production plants, $P_1, \dots, P_I$ that supply a certain commodity, and there are J markets, $M_1, \dots, M_J$, to which the commodity must be shipped.

Port P_i possesses an amount s_i of the commodity ($i = 1, 2, \dots, I$), and market M_j must receive amount r_j of the commodity ($j = 1, 2, \dots, J$). Let b_{ij} be the cost of transporting one unit of the commodity from port P_i to market M_j .

The problem is to meet the market requirements at minimum transportation cost. Defining y_{ij} as the quantity of the commodity shipped from port P_i to market M_j , the total transportation cost will be

$$\sum_{i=1}^I \sum_{j=1}^J y_{ij} b_{ij}.$$

The amount sent from port P_i is $\sum_{j=1}^J y_{ij}$ which cannot exceed s_i and the amount sent to market M_j is $\sum_{i=1}^I y_{ij}$ which must be at least r_j .

This means that our problem reduces to minimising

$$\sum_{i=1}^I \sum_{j=1}^J y_{ij} b_{ij},$$

subject to

$$\sum_{j=1}^J y_{ij} \leq s_i \quad i = 1, \dots, I$$

$$\sum_{i=1}^I y_{ij} \geq r_j \quad j = 1, \dots, J$$

and

$$y_{ij} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J$$

We will look at this problem in detail in Section 4.

The way the transportation problem is written above means that it is not in the form of the standard maximum or minimum problem.

In general any linear programming problem of the form:
find a vector $\mathbf{y} = (y_1, \dots, y_m)$, to minimise/maximise

$$\mathbf{y}^T \mathbf{b} = y_1 b_1 + \dots + y_m b_m$$

subject to the constraints

$$\begin{aligned} y_1 a_{11} + y_2 a_{21} + \dots + y_m a_{m1} &\geq, =, \leq c_1 \\ y_1 a_{12} + y_2 a_{22} + \dots + y_m a_{m2} &\geq, =, \leq c_2 \\ &\vdots \\ y_1 a_{1n} + y_2 a_{2n} + \dots + y_m a_{mn} &\geq, =, \leq c_n \end{aligned}$$

and $y_1 \geq 0, y_2 \geq 0, \dots, y_m \geq 0$

can be converted to the standard minimum problem as follows.

For any constraint

$$y_1 a_{1i} + y_2 a_{2i} + \dots + y_m a_{mi} \leq c_i$$

multiply both sides by -1 to give

$$\begin{aligned} -(y_1 a_{1i} + y_2 a_{2i} + \dots + y_m a_{mi}) &\geq -c_i \Rightarrow \\ y_1(-a_{1i}) + y_2(-a_{2i}) + \dots + y_m(-a_{mi}) &\geq -c_i \end{aligned}$$

Thus any $\leq$ constraint can be converted to a $\geq$ constraint.

Any constraint $y_1a_{1i} + y_2a_{2i} + \dots + y_ma_{mi} = c_i$ is equivalent to saying that both $y_1a_{1i} + y_2a_{2i} + \dots + y_ma_{mi} \geq c_i$ and $y_1a_{1i} + y_2a_{2i} + \dots + y_ma_{mi} \leq c_i$ are true.

The first of these is a $\geq$ constraint. The second can be converted into a $\geq$ constraint by multiplying both sides by -1 as before. Thus every $=$ constraint can be replaced by two $\geq$ constraints.

Some of the y_i s may not be restricted to being nonnegative. An unrestricted variable y_i can be replaced by two nonnegative ones by setting $y_i = u_i - v_i$.

Finally if the objective function needs to be maximised, multiplying by -1 reverses this to require the function to be minimised.

Thus we can convert the Transportation problem of Example 8 to standard form as follows: We need to multiply all of the supply constraints $\sum_{j=1}^J y_{ij} \leq s_i$ by -1 which gives the problem as minimising

$$\sum_{i=1}^I \sum_{j=1}^J y_{ij} b_{ij}$$

subject to

$$\sum_{j=1}^J (-1)y_{ij} \geq -s_i \quad i = 1, \dots, I$$

$$\sum_{i=1}^I y_{ij} \geq r_j \quad j = 1, \dots, J$$

and

$$y_{ij} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J$$

2.3 The dual program

For every linear program there is a dual program. There is a strong connection between the primal program and its dual, as we shall see.

The dual of the standard maximum problem i.e.
maximise $\mathbf{c}^T \mathbf{x}$
subject to the constraints $A\mathbf{x} \leq \mathbf{b}$ and $\mathbf{x} \geq 0$,

is defined to be the standard minimum problem
minimise $\mathbf{y}^T \mathbf{b}$
subject to the constraints $\mathbf{y}^T A \geq \mathbf{c}^T$ and $\mathbf{y} \geq 0$.

Similarly, the dual of the standard minimum problem is the standard maximum problem.

Example 3: continued

Maximize $z = 2x_1 + 9x_2$
subject to the constraints
 $4x_1 + 5x_2 \leq 20$
 $3x_1 + 7x_2 \leq 21$

The dual of this program is
Minimize $z = 20y_1 + 21y_2$
subject to the constraints
 $4y_1 + 3y_2 \geq 2$
 $5y_1 + 7y_2 \geq 9$

which is just Example 5.

Terminology

The function to be maximised or minimised is called the *objective function*.

A vector $\mathbf{x}$ ($\mathbf{y}$) is said to be *feasible* for the standard maximum (minimum) problem if it satisfies the constraints of the problem.

If a linear program has no feasible vectors, it is *infeasible*.

A feasible maximum (minimum) problem is *unbounded* if the objective function can be arbitrarily large and positive (large and negative) at feasible vectors. Otherwise it is *bounded*.

Some results relating to the dual program

If $\mathbf{x}$ is feasible for the maximum problem and $\mathbf{y}$ is feasible for the dual minimum problem, then $\mathbf{c}^T \mathbf{x} \leq \mathbf{y}^T \mathbf{b}$.

If $\mathbf{x}^*$ and $\mathbf{y}^*$ are feasible for the standard maximum problem and its dual and $\mathbf{c}^T \mathbf{x}^* = \mathbf{y}^{*T} \mathbf{b}$, then both are optimal for their respective problems.

The Duality Theorem

If a standard linear programming problem is bounded feasible, then so is its dual, their values are equal, and there exists optimal vectors for both problems.

The Equilibrium Theorem

Let $\mathbf{x}^*$ and $\mathbf{y}^*$ be feasible vectors for the standard maximum problem and its dual. Then $\mathbf{x}^*$ and $\mathbf{y}^*$ are optimal if and only if

$y_i^* = 0$ for all i such that $\sum_{j=1}^n a_{ij}x_j^* < b_i$ and
 $x_j^* = 0$ for all j such that $\sum_{i=1}^m y_i^*a_{ij} > c_j$.

Example 9

Minimise $2y_1 - y_2$ subject to the constraints $y_i \geq 0$ and

$$2y_1 - y_2 + y_3 \geq -1$$

$$-7y_1 + 4y_2 - 6y_3 \geq 1$$

3 Solving the full simplex problem

Consider the standard minimum problem. We write this problem in tabular form, the *simplex tableau*.

	s_1	$\dots$	s_j	$\dots$	s_n	
y_1	a_{11}	$\dots$	a_{1j}	$\dots$	a_{1n}	b_1
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_i	a_{i1}	$\dots$	a_{ij}	$\dots$	a_{in}	b_i
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_m	a_{m1}	$\dots$	a_{mj}	$\dots$	a_{mn}	b_m
	$-c_1$	$\dots$	$-c_j$	$\dots$	$-c_n$	$(v=)0$

If $-\mathbf{c} \geq 0$ and $\mathbf{b} \geq 0$ the solution is obvious. The minimum occurs at $\mathbf{y} = 0$ (i.e. all of the row entries are zero) with $\mathbf{s} = -\mathbf{c}$, with minimum value $\mathbf{y}^T \mathbf{b} = 0$.

This is a feasible solution since $\mathbf{y} \geq 0, \mathbf{s} \geq 0$ and $\mathbf{s}^T = \mathbf{y}^T A - \mathbf{c}$, and $\sum y_i b_i$ cannot be made less than zero as all of the terms are positive.

Suppose that there is at least one negative entry in the last row or last column of the tableau (not including the bottom corner).

If we pivot around a_{11} (assuming it is non-zero), we obtain the alternative tableau

	y_1	$\dots$	s_j	$\dots$	s_n	
s_1	$\hat{a}_{11}$	$\dots$	$\hat{a}_{1j}$	$\dots$	$\hat{a}_{1n}$	$\hat{b}_1$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_i	$\hat{a}_{i1}$	$\dots$	$\hat{a}_{ij}$	$\dots$	$\hat{a}_{in}$	$\hat{b}_i$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_m	$\hat{a}_{m1}$	$\dots$	$\hat{a}_{mj}$	$\dots$	$\hat{a}_{mn}$	$\hat{b}_m$
	$-\hat{c}_1$	$\dots$	$-\hat{c}_j$	$\dots$	$-\hat{c}_n$	$\hat{v}$

If in the new problem $\hat{\mathbf{b}} \geq 0$, $-\hat{\mathbf{c}} \geq 0$, then again we have a solution, where all of the row entries are zero (this now includes s_1 and excludes y_1 , which may thus take a positive value).

We could keep pivoting, in the hope of chancing on the optimal solution, but there is a more efficient method, which we saw when looking at the two variable case.

In general we can investigate a program and its dual simultaneously. We write the simplex tableau as

	x_1	$\dots$	x_j	$\dots$	x_n	
y_1	a_{11}	$\dots$	a_{1j}	$\dots$	a_{1n}	b_1
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_i	a_{i1}	$\dots$	a_{ij}	$\dots$	a_{in}	b_i
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
y_1	a_{m1}	$\dots$	a_{mj}	$\dots$	a_{mn}	b_m
	$-c_1$	$\dots$	$-c_j$	$\dots$	$-c_n$	$(v=)0$

If we pivot until entries in the last row and column (excluding the right-hand corner) are non-negative, then the value of the program and the dual are the entry in the bottom right-hand corner.

The solution of the minimum problem is found by letting the y_i s on the left be zero and those on top equal to that column entry in the bottom row.

The solution of the maximum problem is found by letting the x_j s on top be zero, and those on the left be equal to that row's entry in the final column.

Case 1

Suppose that we wish to solve the maximum problem and in the original tableau, all of the entries $b_i \geq 0$.

We take any column e.g. j_0 which has a negative final entry $-c_{j_0}$. We then consider all of the entries in the column for which $a_{ij_0} > 0$, and choose the one for which the ratio b_i/a_{ij_0} is the smallest (call this row i_0). We then pivot around $a_{i_0j_0}$.

Example 10

Maximise $x_1 + x_2 + 2x_3$ subject to $x_i \geq 0$ and

$$x_2 + 2x_3 \leq 3$$

$$-x_1 + 3x_3 \leq 2$$

$$2x_1 + x_2 + x_3 \leq 1$$

	x_1	x_2	x_3	
y_1	0	1	2	3
y_2	-1	0	3	2
y_3	2	[1]	1	1
	-1	-1	-2	0

	x_1	y_3	x_3	
y_1	-2	-1	1	2
y_2	-1	0	[3]	2
x_2	2	1	1	1
	1	1	-1	1

	x_1	y_3	y_2	
y_1				4/3
x_3				2/3
x_2				1/3
	2/3	1	1/3	5/3

The value for the program is thus $5/3$, achieved when $x_1 = 0, x_2 = 1/3, x_3 = 2/3$ and the first inequality has a slack of $4/3$.

Notice that the slack terms in the constraints in the maximum problem are given by the y values, and similarly the *surplus* terms in the minimum problem are given by the x values.

Example 11

Maximise $x_1 - 2x_2 - 3x_3 - x_4$ subject to the constraints

$x_j \geq 0$ and

$$x_1 - x_2 - 2x_3 - x_4 \leq 4$$

$$2x_1 + x_3 - 4x_4 \leq 2$$

$$-2x_1 + x_2 + x_4 \leq 1$$

Case 2

Suppose that some of the b_i s are negative. Take the first of these, say b_k . Find any negative entry in row k , say a_{kj_0} (if there is no such entry the maximum problem is not feasible).

We will pivot in column j_0 . Compare b_k/a_{kj_0} and the b_i/a_{ij_0} for which $b_i \geq 0$ and $a_{ij_0} > 0$, and select the i (i_0) for which the ratio is the smallest. Pivot around $a_{i_0j_0}$.

Example 12

Minimise $3y_1 - 2y_2 + 5y_3$ subject to $y_i \geq 0$ and

$$-y_2 + 2y_3 \geq 1$$

$$y_1 + y_3 \geq 1$$

$$2y_1 - 3y_2 + 7y_3 \geq 5$$

3.1 The dual simplex method

Recall that the tableau has been written in such a way as to address both the maximum and minimum problems.

Suppose that we want to solve the maximum problem as before, and we already have a feasible vector for the minimum problem (so that the bottom row of the tableau is non-negative). It is better to improve this vector than apply the rule from case 2 of the maximum problem.

Case 1

If all $-c_j \geq 0$, we take any row where the last entry is negative $b_{i_0} < 0$.

Choose j_0 for which the ratio $-c_j/a_{i_0j}$ is closest to zero out of the j for which $a_{i_0j} < 0$. Pivot on $a_{i_0j_0}$.

More generally we can apply this method even if all of the bottom row entries are not non-negative.

Case 2

Some $-c_j < 0$. Take the first k s.t. $-c_k < 0$. Find any positive entry in column k , a_{i_0k} .

Compare $-c_k/a_{i_0k}$ and the $-c_j/a_{i_0j}$ for which $c_j \leq 0$ and $a_{i_0j} < 0$ and choose j_0 for which this is closest to 0 (j_0 may be k). Pivot on $a_{i_0j_0}$.

Example 13

	x_1	x_2	x_3	
y_1	[-2]	-1	-2	-1
y_2	2	3	5	2
y_3	-3	4	-1	1
	1	2	3	0

	y_1	x_2	x_3	
x_1				1/2
y_2				1
y_3				5/2
	1/2	3/2	2	-1/2

Since we had a feasible solution to the minimum problem, we applied case 1 above, leading directly to the solution.

3.2 Cycling

The simplex method as we have described it gives no improvement if b_r , the constant term in the pivot row, is 0. A sequence of pivots, chosen according to the rules, can cycle round and leave you back at the original tableau.

Consider the following example

Example 14

Maximise $3x_1 - 5x_2 + x_3 - 2x_4$ subject to $x_i \geq 0$ and
 $x_1 - 2x_2 - x_3 + 2x_4 \leq 0$
 $2x_1 - 3x_2 - x_3 + x_4 \leq 0$
 $x_3 \leq 1$

The “wrong” choice of pivots leads back to the original tableau. Specifically this occurs if we choose to pivot by swapping

x_1 and y_1 then

x_2 and y_2 then

x_3 and x_1 then

x_4 and x_2 then

y_1 and x_3 then

y_2 and x_4 .

This problem is actually quite rare in practice, and is due to degeneracy in some of the program conditions.

The following procedure, called the *smallest subscript rule*, avoids cycling.

If there is a choice of candidate pivot columns or a choice of pivot rows, choose the candidate with the x variable having the lowest subscript, or if there are no x variables, with the y variable with the lowest subscript.

Example 14 continued

Solve Example 14 using the smallest subscript rule.

3.3 The general simplex method

Allowing for equalities in the constraints and unrestricted variables, we have the *general maximum problem*

Find a vector

$\mathbf{x} = (x_1, \dots, x_n)$, to maximise

$$\mathbf{c}^T \mathbf{x} = c_1 x_1 + \dots + c_n x_n$$

subject to the constraints

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad i = 1, \dots, k$$

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \quad i = k+1, \dots, m$$

$$x_j \geq 0 \quad j = 1, \dots, l$$

$$x_j \text{ unrestricted} \quad j = l+1, \dots, n$$

and the *general minimum problem*

Find a vector

$\mathbf{y} = (y_1, \dots, y_m)$, to minimise

$$\mathbf{y}^T \mathbf{b} = y_1 b_1 + \dots + y_m b_m$$

subject to the constraints

$$y_1 a_{1j} + y_2 a_{2j} + \dots + y_m a_{mj} \geq c_j \quad j = 1, \dots, l$$

$$y_1 a_{1j} + y_2 a_{2j} + \dots + y_m a_{mj} = c_j \quad j = l+1, \dots, n$$

$$y_i \geq 0 \quad i = 1, \dots, k$$

$$y_i \text{ unrestricted} \quad i = k+1, \dots, m$$

A strict equality in the constraints of one program corresponds to an unrestricted variable in the dual.

Suppose that we wish to solve the general minimum problem. We add the *slack variables* $\mathbf{s}^T = \mathbf{y}^T A - \mathbf{c}^T$ so that the constraints demand that $s_1 \geq 0, \dots, s_l \geq 0, s_{l+1} = 0, \dots, s_n = 0$.

If we pivot the tableau so that s_{l+1} goes from the top to the left, the corresponding $\hat{b}_i$ is irrelevant, as it is the coefficient multiplied by s_{l+1} in the objective function, and $s_{l+1} = 0$; we can just delete the row.

Similarly the unconstrained y_i s must be pivoted to the top where they represent dependent variables. We again do not care about whether $-\hat{c}_j$ are positive or negative since y_i is unconstrained. Thus we can just delete the column.

The general simplex method comprises three stages.

In the first stage, all equality constraints and unconstrained variables are pivoted (and removed if desired).

In the second stage, we use the simplex pivoting rules to obtain a feasible solution for either the original problem, or its dual.

In the third stage, we improve the feasible solution according to the standard simplex pivoting rules until we find the optimal solution.

Example 15

Maximise $5x_2 + x_3 + 4x_4$ subject to the constraints

$x_1 \geq 0, x_2, x_4 \geq 0, x_3$ unrestricted and

$$-x_1 + 5x_2 + 2x_3 + 5x_4 \leq 5$$

$$3x_2 + x_4 = 2$$

$$-x_1 + x_3 + 2x_4 = 1$$

4 Two key problems

4.1 The transportation problem

We consider the transportation problem of Example 8.

Minimise

$$\sum_{i=1}^I \sum_{j=1}^J y_{ij} b_{ij},$$

subject to

$$\sum_{j=1}^J y_{ij} \leq s_i \quad i = 1, \dots, I \quad (5)$$

$$\sum_{i=1}^I y_{ij} \geq r_j \quad j = 1, \dots, J \quad (6)$$

and

$$y_{ij} \geq 0 \quad i = 1, \dots, I; j = 1, \dots, J$$

The dual problem is to find $u_1, \dots, u_I$ and $v_1, \dots, v_J$ to maximise

$$\sum_{j=1}^J r_j v_j - \sum_{i=1}^I s_i u_i$$

subject to the constraints $v_j \geq 0, u_i \geq 0$ and $v_j - u_i \leq b_{ij}$ for all i, j .

Clearly the transportation problem is not feasible unless supply is not exceeded by demand i.e.

$$\sum_{i=1}^I s_i \geq \sum_{j=1}^J r_j$$

The following procedure finds a solution of the transportation problem assuming that total demand is equal to total supply. If this is not true, we can easily make it so by creating a “dummy” market where all transportation costs are zero. Thus we assume

$$\sum_{i=1}^I s_i = \sum_{j=1}^J r_j$$

This in turn means that all of the inequalities in equations (5) and (6) must be satisfied as equalities, so that

$$\sum_{j=1}^J y_{ij} = s_i \quad i = 1, \dots, I \quad (7)$$

$$\sum_{i=1}^I y_{ij} = r_j \quad j = 1, \dots, J \quad (8)$$

Due to the equalities in (7) and (8) and the link between the general minimum problem and its dual, the equilibrium conditions reduce to

$$y_{ij} > 0 \Rightarrow v_j - u_i = b_{ij}$$

Note that we can of course use the simplex method to solve this problem. This involves an $IJ \times (I + J)$ constraint matrix, and we use a more efficient method. We use the *transportation array* below.

	M_1	M_2	$\dots$	M_J	
P_1	b_{11}	b_{12}	$\dots$	b_{1J}	
	y_{11}	y_{12}	$\dots$	y_{1J}	s_1
P_2	b_{21}	b_{22}	$\dots$	b_{2J}	
	y_{21}	y_{22}	$\dots$	y_{2J}	s_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
P_I	b_{I1}	b_{I2}	$\dots$	b_{IJ}	
	y_{I1}	y_{I2}	$\dots$	y_{IJ}	s_I
	r_1	r_2	$\dots$	r_J	

We will look at the following example

Example 16

	M_1	M_2	M_3	M_4	
P_1	4	7	11	3	
					5
P_2	7	5	6	4	
					7
P_3	1	3	4	8	
					8
	2	9	4	5	

Thus e.g. P_1 has 5 units of commodity, M_1 requires 2 units, and it costs 4 per unit to move the commodity from P_1 to M_1 .

There are three steps to our algorithm.

- 1) Find a basic feasible shipping schedule, y_{ij} ;
- 2) test for optimality;
- 3) if the test fails, find an improved basic feasible schedule, and repeat 2).

1) Firstly choose any square, e.g. S_{11} and make the value of y_{11} as large as possible. In our example $y_{11} = 2$. Thus we fill M_1 's needs from P_1 and so $y_{21} = y_{31} = 0$.

Now choose S_{12} and make y_{12} as large as possible. Now $y_{12} = 3$, and the remainder of P_1 's supply goes to M_2 . Hence $y_{13} = y_{14} = 0$.

Now pick S_{22} , and we set $y_{22} = 6$, so the remainder of M_2 's supply comes from P_2 . Hence $y_{32} = 0$.

We continue in this way until all the values of y_{ij} are chosen (at each step we will either exhaust a port (row) and move to the next one, or fill a market (column) and move to the next one. This is the *Northwest corner rule*.

The initial solution for Example 16 is given by

	M_1	M_2	M_3	M_4	
P_1	4 [2]	7 [3]	11	3	5
P_2	7	5 [6]	6 [1]	4	7
P_3	1	3	4 [3]	8 [5]	8
	2	9	4	5	

Note that we have put in bold all of the y_{ij} s that appear in the solution.

A better initial solution can (usually) be found by using the *least cost rule*. This simply involves repeatedly allocating as much as possible to the square with the least cost remaining.

Thus in Example 16 the least cost is in the lower left square, and $y_{31} = 2$ is circled and M_1 is filled.

The next smallest score is 3 (twice). We will fill the upper right box next but this fills both P_1 and M_4 at once, and the rule only allows us to discard one of these.

This is why we have a ringed zero entry in the table. Finally we get the array

	M_1	M_2	M_3	M_4	
P_1	4	7	11	3	5
				[5]	
P_2	7	5	6	4	7
		[3]	[4]	[0]	
P_3	1	3	4	8	8
	[2]	[6]			
	2	9	4	5	

2) Checking for optimality.

Set $v_j - u_i = b_{ij} \quad \forall i, j$

Set one of the u_i s equal to zero and solve for all of the others (there are now $I + J - 1$ unknowns and $I + J - 1$ equations), using the circled entries in the table.

Feasibility of the dual variables must then be checked i.e. that

$$v_j - u_i \leq b_{ij}$$

for all of the remaining squares. if this is not the case, then our solution is not optimal (recall the Duality Theorem and related ideas).

Some of the u_i, v_j s may be negative. Adding a large constant K to all entries preserves the equalities/inequalities whilst ensuring all of the u_i, v_j s are now positive.

Let us return to Example 16 with the least cost rule. First we solve for the u_i and v_j . Setting $u_2 = 0$ we get $v_2 = 5, v_3 = 6$ and $v_4 = 4$. This then gives $u_1 = 1, u_3 = 2$ and then $v_1 = 3$.

Writing the u_i and v_j in the row/column places instead of the P_i and M_j gives

	3	5	6	4	
1	4	7	11	3	
0				[5]	5
2	1	3	4	8	
	[2]	[6]			8
	2	9	4	5	

Checking each entry in turn, we have a feasible solution if the non-circled squares all satisfy $v_j - u_i \leq b_{ij}$.

The top left does since $3 - 1 = 2 \leq 4$. Similarly the others also do. We can see that all the squares also satisfy the constraints, so we have a solution to both the primal and dual problems.

Thus we have the solution to the optimal transportation rule, with value $\sum_{i,j} y_{ij} b_{ij} = 74$.

3) Improving our solution.

If we have tested a solution and the dual variables are not feasible, we need to find a better solution.

We choose any square $S_{i,j}$ where $v_j - u_i > b_{ij}$, set $y_{ij} = \theta$ in that square, but keep the constraints satisfied by adding and subtracting θ to appropriate circled variables.

We choose θ to be the minimum of the variables in the squares where θ is subtracted. We circle the new variable and remove one of the variables (there will usually be exactly one) that has become zero.

Example 17

Consider the above example with $b_{21} = 2$. This means that the duals are no longer feasible for that square. We add θ to S_{21} , which means we must take θ from S_{31} and S_{22} , and then add θ to S_{32} .

We now put one of y_{31} or y_{22} to zero by selecting θ to be the smallest of these two values. Here that is 2.

In the array this gives

	M_1	M_2	M_3	M_4	
P_1	4	7	11	3	
				[5]	5
P_2	2	5	6	4	
	$+\theta$	$-\theta$ [3]	[4]	[0]	7
P_3	1	3	4	8	
	$-\theta$ [2]	$+\theta$ [6]			8
	2	9	4	5	

This gives

	2	5	6	4	
1	4	7	11	3	
				[5]	5
0	2	5	6	4	
	[2]	[1]	[4]	[0]	7
2	1	3	4	8	
		[8]			8
	2	9	4	5	

Example 18

Solve the transportation problem

1	3	5	8	
				6
2	5	6	7	
				5
1	2	3	5	

4.2 The assignment problem

Suppose that each of n candidates is to be assigned to one of n jobs, and we know the ability of candidate i to do job j , r_{ij} . The assignment problem asks how to allocate the candidates in such a way so as to maximise

$$\sum_{i=1}^n \sum_{j=1}^n r_{ij}$$

We can think of $R = (r_{ij})$ as a matrix and the problem reduces to picking n entries from the matrix, with exactly one entry in each row and column, so that the sum of the entries is maximised.

Note that the assignment problem is a version of the transportation problem.

We will use the following definitions to help solve the assignment problem.

Definition 1 A line is a row or column of a matrix.

Definition 2 A set of elements in a matrix are independent if no two elements lie in the same line.

Definition 3 If the letter c is written on the left of a row or at the top of a column, then the corresponding row or column is said to be *covered*. An element in a covered line is said to be covered, and may be covered twice.

Konig's Theorem

Let A be a square matrix. Then the maximum number of independent zeros of A is equal to the minimum number of lines required to cover all of the zeros of A .

Example 19

	c				
c	8	0*	0	2	0
c	2	0	5	0*	6
	1	1	0*	1	4
	3	2	0	2	1
c	0*	0	4	7	0

In the above matrix the maximum number of independent zeros is 4 (one possibility is given by the asterisks). The minimum number of lines needed to cover all 0s is 4, as illustrated.

The following method to solve the assignment problem is called the *Hungarian Method*.

Define $r = \max_{i,j} r_{ij}$, and the matrix S by $S = \{s_{ij} : s_{ij} = r - r_{ij}\}$.

This reduces the problem to finding n independent elements of S which give a minimum sum.

Note that the problem is unchanged if we subtract an arbitrary u_i from all elements in row i of S , or subtract an arbitrary v_j from all elements of column j of S .

The aim is to alter the matrix S so that we can apply Konig's Theorem.

Step 1 Subtract the smallest element in each row of S from every element in that row, giving S_1 .

Step 2 Subtract the smallest element in each column of S_1 from every element in that column, giving A . Then A contains at least one zero in every row and column.

Step 3 Find k_m , the maximum number of independent zeros of A . By Konig's Theorem, k_m is also the minimum number of lines needed to cover all the zeros of A .

If $k_m = n$, then there are n independent zeros of A , and the elements in the corresponding positions of R give a solution to the original assignment problem.

Step 4 If $k_m < n$, let N denote the matrix of non-covered elements of A , and let h be the smallest element of N . Add h to each twice covered element of A , and subtract h from each non-covered element of A , denoting the new matrix as A^* .

This step keeps every element of A non-negative, but decreases the sum of all elements in A by $nh(n - k_m)$.

Now repeat step 3 using A^* instead of A . If k_m is now equal to n , the problem is solved. If not treat A^* as A and thus obtain a new A and A^* . This is continued until $k_m = n$.

Example 20

Cand/Job	1	2	3	4
1	15	30	72	18
2	58	82	90	65
3	45	35	95	45
4	80	45	35	57

How should we allocated the candidates to the jobs?

Finding S gives

80	65	23	77
37	13	5	30
50	60	0	50
15	50	60	38

Step 1 gives us

57	42	0	54
32	8	0	25
50	60	0	50
0	35	45	23

Step 2 gives

	c			
	57	34	0*	31
c	32	0*	0	2
	50	52	0	27
c	0	27	45	0*

There is a maximum of three independent zeros. Step 4 yields N as

$$\begin{array}{ccc} 57 & 34 & 31 \\ 50 & 52 & 27 \end{array}$$

which gives $h = 27$. This then yields A^* as

$$\begin{array}{cccc} 30 & 7 & 0^* & 4 \\ 32 & 0^* & 27 & 2 \\ 23 & 25 & 0 & 0^* \\ 0^* & 27 & 72 & 0 \end{array}$$

which has four independent zeros. Thus the solution is given by giving
 job 3 to candidate 1,
 job 2 to candidate 2,
 job 4 to candidate 3,
 job 1 to candidate 4.

The sum of the values of r that we have maximised is $72+82+45+80 = 279$.

Example 21

Solve the following assignment problem

Cand/Job	1	2	3	4
1	14	7	22	11
2	32	33	41	15
3	14	20	29	6
4	23	16	17	22