MA30087/50087: Optimisation methods of operational research

Brief notes to accompany the lectures.[1]

Prepared by A. E. Kyprianou[2]
Department of Mathematical Sciences
The University of Bath
Claverton Down
BA2 7AY

Suggested further reading: Elementary Linear Programming With Applications, (Second Edition). B. Kolman and R. E. Beck, Academic Press 1980. ISBN 0-12-417910-X. This book goes at rather a slow pace but spells the theory and practice out very clearly.

Other books:
- Understanding and Using Linear Programming Series: Universitext J. Matoušek, B. Gärtner, Springer (2007).
- B.D. Bunday, Basic Optimization Methods, Hodder Arnold (1984) .
- Bazaran, J.J. Harvis and H.D. Shara'i, Linear Programming and Network Flows, Wiley (2005).
- V. Chvatal, Linear Programming, Freeman (1983).
- D. G. Luenberger, Linear and Nonlinear Programming, Addison-Wesley (1984).
- F. S. Hillier, G. J. Lieberman, Introduction to Operations Research. McGraw-Hill (2001), 7th Edition.
- A. M. Glicksman, An Introduction to Linear Programming and the Theory of Games. Dover.

A very interesting source of material: The internet - try googling 'linear programming' and look for lecture notes given at other universities!!

[1]Corrections are strongly encouraged to be sent to a.kyprianou@bath.ac.uk
[2]Also partly based on the older notes of M.E. Brigden. Thanks also to M.E. Brigden for supplying me with a comprehensive set of notes with additional examples and exercises.

Contents

1 Introduction: aims of the course

The principle aim of this course will be to cover the basic ideas and techniques which lie behind modern-day linear programming. One may think of the latter as a field of applied mathematics which concerns itself with resource allocation making use of classical elements of Linear Algebra. Specifically the reader, who should be well versed in an understanding of basic Linear Algebra, will see that many of the results and techniques presented boil down to an application of concepts such as

linear independence and matrix inversion,

convexity in Euclidian spaces and

geometric interpretation of linear equations.

The course is divided into essentially two parts. First we consider the classical **linear programming problem** and show that there exists a generic method for analysing such problems with the so called **the simplex method**. In the second part of the course we look at a more complicated class of linear programming problems which come under the heading of **the transport problem** and **optimisation on networks**. We consider methods of solution to the latter class of problems which can be thought of as variants of the simplex method.

Some of the proofs in this course are rather lengthy but none are particularly demanding in view of the volume of mathematics that 3rd year students in their first semester will have seen thus far. Although you may be expecting this course to be more 'applied' and therfore 'easier', please remember that even applications require some degree of intellectual justification for them to be of any value. Therefore the algorithms alone will be meaningless to you unless you make the effort to understand the mathematics behind them.

2 The linear programming problem

2.1 A Simple Example

Let us begin by with an example. Betta Machine Products plc makes two different products. Each requires casting, machining and assembly time. Products are made at a fixed cost and likewise sold at a fixed cost. This information is given in the table below.

	Casting	Machining	Assembly	Cost	Selling Price
Product 1	5	5	1	25	40
Product 2	8	4	3	30	50

Times are given in minutes per unit. Costs and Prices are in pounds. Each week there are 16,000 mins of casting time, 14,000 mins of machining time and 5,000 mins of assembly time available and there is no limit to numbers that can be sold. **The objective of this company is to maximise the difference between total revenue and total cost.**

We may now formulate this optimisation problem in a purely mathematical context. Define x_1 units of product 1 and x_2 units of product 2 each week. Our objective is to:

$$\begin{array}{l} \text{maximise } z = 15x_1 + 20x_2 \\ \text{subject to:} \\ 5x_1 + 8x_2 \leq 16,000 \\ 5x_1 + 4x_2 \leq 14,000 \\ x_1 + 3x_2 \leq 5,000 \\ x_1 \geq 0, x_2 \geq 0. \end{array}$$

This is a linear programming problem with two variables and three constraints. (Strictly speaking there are in fact five constraints but, as we shall shortly see, positivity of variables is a usual requirement). Many practical problems have literally thousands of variables and constraints.

2.2 Another example

We wish to feed cattle, meeting nutritional requirements at minimum cost. A farmer has two feeds A and B at her disposal. There are certain nutritional requirements which stipulate that cattle should receive minimum quantities of carbohydrate, vitamins and protein. The table below gives the number of units per kilo of the nutrients.

Food	Carbo.	Vit.	Pro.	Cost (pence/kilo)
A	10	2	2	40
B	5	3	8	80
Daily Requirement	10	3	4	

The mathematical formulation of the above problem is as follows. Use x_1 kilos of food A and x_2 kilos of B per day. Our objective is to:

$$\begin{aligned}&\text{minimise } z = 40x_1 + 80x_2\\ &\text{subject to:}\\ &10x_1 + 5x_2 \geq 10\\ &2x_1 + 3x_2 \geq 3\\ &2x_1 + 8x_2 \geq 4\\ &x_1, x_2 \geq 0.\end{aligned}$$

This is an example of a **diet problem**. Note that where as in the previous example we were asked to maximise a linear function, in this problem we are asked to minimise a linear function. None the less, as we see in the next section, both problems fit into the same framework.

2.3 A third example

The R.H.Lawn Products Co. has available 80 metric tons of nitrate and 50 metric tons of phosphate to use in producing its three types of fertiliser during the coming week. The mixture ratios and profit figures are given in the table below. Determine how the current inventory should be used to maximise profit.

	Metric tons / 1000 bags		Profit
	Nitrate	Phosphate	(dollars / 1000 bags)
Regular lawn	4	2	300
Super lawn	4	3	500
Garden	2	2	400

The above problem is again a maximisation problem. In this case we let x_1, x_2, x_3 denote the quantities of the number of bags of the three products in the left hand column of the given table respectively. Our objective is to:

$$\begin{array}{l} \text{maximise } z = 300x_1 + 500x_2 + 400x_3 \\ \text{subject to:} \\ 4x_1 + 4x_2 + 2x_3 \leq 80 \\ 2x_1 + 3x_2 + 2x_3 \leq 50 \\ x_1, x_2, x_3 \geq 0. \end{array}$$

Thus we reach a linear programming problem with three variables and two constraints.

2.4 General statement of the linear programming problem

In general we may identify a linear programming problem to take the following **standard form**. Suppose that $\mathbf{A}$ is a matrix belonging to $\mathbb{R}^{m\times n}$ where $m, n \geq 1$, $\mathbf{b}$ is a vector belonging to $\mathbb{R}^m$ and $\mathbf{c}$ is a vector belonging to $\mathbb{R}^n$ and $\mathbf{0}_n$ is an n-dimensional vector filled with zeros. Our objective is to find a vector $\mathbf{x} \in \mathbb{R}^n$ that will:

$$\begin{array}{l} \text{maximise } z := \mathbf{c} \cdot \mathbf{x} \\ \text{subject to:} \\ \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array} \tag{1}$$

This is a compact way of saying:

$$\begin{array}{l} \text{maximise } z := c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{subject to:} \\ A_{11}x_1 + A_{12}x_2 \cdots A_{1n}x_n \leq b_1 \\ A_{21}x_1 + A_{22}x_2 \cdots A_{2n}x_n \leq b_2 \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 \cdots A_{mn}x_n \leq b_m \\ x_j \geq 0,\ j = 1, 2, ..., n. \end{array}$$

Notes.

1. In general there are no restrictions on the dimension of the matrix $\mathbf{A}$ so that either $m \leq n$ or $n \leq m$.

2. A ‘minimisation’ problem (i.e. a problem which requires one to minimise a linear function of the form $\mathbf{c} \cdot \mathbf{x}$) can easily be converted to a maximisation problem by changing $\mathbf{c}$ to $-\mathbf{c}$.

3. A problem which appears to require $\mathbf{Ax} \geq \mathbf{b}$ (i.e. the inequality goes the wrong way) can reduced to the standard form by writing it in the form $-\mathbf{Ax} \leq -\mathbf{b}$.

4. A problem which appears to require that $\mathbf{Ax} = \mathbf{b}$ (i.e. an equality instead of an inequality) can be re-expressed in terms of the requirements that $\mathbf{Ax} \leq \mathbf{b}$ and $-\mathbf{Ax} \leq -\mathbf{b}$. However, as we shall see later when we define the **canonical form** of a linear programming problem, having an equality in place of an inequality is an advantage.

5. Sometimes one may find that instead of $\mathbf{x} \geq 0$, one has unconstrained values for the vector $\mathbf{x}$. In that case, one may always write $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$ where $\mathbf{x}^\pm \geq 0$ and then one may re-write the conditions on $\mathbf{x}$ as conditions on the concatenation of $\mathbf{x}^+$ and $\mathbf{x}^-$.

We conclude this section with a definition of two types of solutions.

Definition 2.1 *Consider the standard form of the linear programming problem (2). We say that $\mathbf{x} \in \mathbb{R}^n$ is a* **feasible solution** *if it respects all the constraints of the specified problem. If $\mathbf{x}$ is feasible and maximises the value of z then we call it an* **optimal solution***. (We will also refer to it on occasion as an* **optimal feasible solution**[3]*).*

Note, we have no reason to believe at this point in time that any given linear programming problem has a feasible solution, let alone an optimal feasible solution. Moreover, even if it does have an optimal solution, then it may not necessarily be unique.

2.5 Slack variables and canonical form

If instead of $\mathbf{Ax} \leq \mathbf{b}$ we have $\mathbf{Ax} = \mathbf{b}$ in the formulation of the linear programming problem then we say that it is in **canonical form**. In other

[3]Sometimes we will drop the word ‘feasible’ as clearly an optimal solution must necessarily be feasible!

words, a linear programming problem in canonical form requires one to find a vector $\mathbf{x} \in \mathbb{R}^n$ that will:

$$\begin{array}{l} \text{maximise } z := \mathbf{c} \cdot \mathbf{x} \\ \text{subject to:} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array} \tag{2}$$

The way to convert a problem from standard form into canonical form is to introduce **slack variables**. That is to say, if $\mathbf{A}$ is a matrix belonging to $\mathbf{A}^{m \times n}$ then we introduce m new variables, say $u_1, ..., u_m$ so that the i-th constraint becomes

$$A_{i1}x_1 + \cdots + A_{in}x_n + u_i = b_i$$

for $i = 1, ..., m$. If we now introduce the new vector belonging to $\mathbb{R}^{n+m}$ which is the concatenation of $\mathbf{x}$ and $\mathbf{u}^{\mathrm{T}} = (u_1, ..., u_m)$, say

$$\mathbf{x}' = \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}.$$

Similarly we introduce the vector $\mathbf{c}'$ which is the concatenation of $\mathbf{c}$ and $\mathbf{0}_m$,

$$\mathbf{c}' = \begin{pmatrix} \mathbf{c} \\ \mathbf{0}_m \end{pmatrix}$$

and define a new $\mathbb{R}^{m \times (m+n)}$ matrix

$$\mathbf{A}' = (\mathbf{A}|\mathbf{I}_m)$$

where $\mathbf{I}_m$ is the m-dimensional identity matrix, then the linear programming problem in standard form can be re-expressed as a linear programming problem in linear form by identifying it as having objective to find a vector $\mathbf{x}'$ in $\mathbb{R}^{n+m}$ that will:

$$\begin{array}{l} \text{maximise } z' := \mathbf{c}' \cdot \mathbf{x}' \\ \text{subject to:} \\ \mathbf{A}'\mathbf{x}' = \mathbf{b} \\ \mathbf{x}' \geq \mathbf{0}_{m+n}. \end{array}$$

As with the standard form of the linear programming problem, we call a **feasible solution** for the canonical form to be any solution which satisfies

$\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}_n$. Further, an **optimal solution** for the canonical form is any feasible solution which optimises the value of objective function.

An important observation is that if one is presented with a linear programming problem in standard form, solving the associated linear programming problem in canonical form will produce the same optimal value of the objective. Further, restricting the optimal solution of the canonical problem to the original variables will also give an optimal solution to the problem in standard form. In principle this is all intuitively obvious, however for a proof, see the accompanying exercise sheet.

Let us conclude this chapter with a **note on notation**. In the above discussion on the introduction of slack variables, we started with n variables and m constraints which meant that we needed to introduce m slack variables. Hence we ended up with a matrix of dimension $m \times (m+n)$. However, more often than not, when introducing a linear programming problem **already** in canonical form for the purpose of theoretical discussion, we shall write the constraints simply as $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is a matrix of dimension $m \times n$. That is to say, we shall not bother indicating the number of slack variables that have been introduced (if al all) and use m and n as generic terms for the dimensions of the matrix $\mathbf{A}$ as it appears in the given linear programming problem.

3 Geometrical considerations

3.1 Constraints, objective functions and hyperplanes

There is a very natural geometrical interpretation of a system of linear equations which are presented in the form $\mathbf{Ax} \leq \mathbf{b}$ where $\mathbf{A}$ is a prespecified $\mathbb{R}^{m \times n}$ matrix, $\mathbf{x}$ is an $\mathbb{R}^n$ vector and $\mathbf{b}$ is a prespecified $\mathbb{R}^m$ vector. Suppose for $i = 1, ..., m$ we write $\mathbf{r}_i$ as the vector corresponding to the i-th row of $\mathbf{A}$ then the system $\mathbf{Ax} \leq \mathbf{b}$ may otherwise be written

$$\mathbf{r}_i \cdot \mathbf{x} \leq b_i \text{ for } i = 1, ..., m$$

where b_i is the i-th entry of $\mathbf{b}$.

At this point it is instructive to recall that in n-dimensional Euclidian space one may describe any hyperplane in the the form

$$\mathbf{n} \cdot \mathbf{x} = d$$

where $\mathbf{n}$ is a vector of unit length orthogonal to the hyperplane and d is the orthogonal distance of the hyperplane from the origin. Indeed the intuition behind this formula is that any vector $\mathbf{x}$ which belongs to such a hyperplane can be decomposed into the sum of two vectors. The first vector moves from the origin onto the hyperplane at the closest position to the origin and the second vector moves from the aforementioned point on the hyperplane to $\mathbf{x}$. That is to say

$$\mathbf{x} = d\mathbf{n} + \mathbf{y}$$

where obviously the first vector on the right hand side is the shortest straight line from the origin to the hyperplane and the vector $\mathbf{y}$ is the relative position of $\mathbf{x}$ to the closest point on the hyperplane to the origin and must be **orthogonal** to $\mathbf{n}$. See Figure 1.

Hence $\mathbf{n} \cdot \mathbf{x} = d\mathbf{n} \cdot \mathbf{n} + \mathbf{n} \cdot \mathbf{y} = d + 0$. Note in particular that when each of the vectors is taken in two dimensional Euclidian space then

$$\mathbf{n} \cdot \mathbf{x} = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = n_1 x_1 + x_2 x_2 = d$$

which is the equation of a line. If we are working in three-dimensional Euclidian space then

$$\mathbf{n} \cdot \mathbf{x} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = n_1 x_1 + n_2 x_2 + n_3 x_3 = d$$

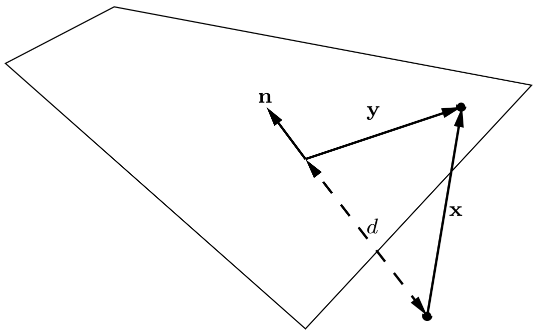


Figure 1: A visial representation of the plane $\mathbf{n} \cdot \mathbf{x} = d$.

which is the equation of a plane.

If instead we now consider the equation

$$\mathbf{n} \cdot \mathbf{x} \leq d$$

then in light of the above intuitive explanation one sees that the latter equation describes a closed half space which lies 'to one side' of the hyperplane $\mathbf{n} \cdot \mathbf{x} = d$. Indeed, if a vector $\mathbf{x}$ satisfies $\mathbf{x} \cdot \mathbf{n} < d$ then we can imagine that it lies on another hyperplane with the same normal $\mathbf{n}$ which is at some other orthogonal distance, say D, from the origin; i.e. $\mathbf{n} \cdot \mathbf{x} = D$. Then if, for example, $d > 0$ and $D < d$ then $\mathbf{x}$ lies on a hyperplane parallel to $\mathbf{n} \cdot \mathbf{x} = d$, which is contained in the same half-space as the origin is and which is an orthogonal distance D from the origin.

Returning to the linear programming problem in standard form (1) we now see that if the vector $\mathbf{x}$ satisfies $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ then it must lie within the intersection of a number of closed half spaces whose boundaries are described by the hyperplanes

$$\mathbf{r}_i \cdot \mathbf{x} \leq b_i \text{ for } i = 1, ..., m.$$

Indeed, even the additional positivity constraint $\mathbf{x} \geq \mathbf{0}_n$ is also a requirement that $\mathbf{x}$ lies to one side of a hyperplane. It says that $\mathbf{x}$ lies in the positive orthant of $\mathbb{R}^n$ which is a quick way of saying that

$$\begin{pmatrix} -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \leq 0, \begin{pmatrix} 0 \\ -1 \\ \vdots \\ 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \leq 0, \cdots \text{ and } \begin{pmatrix} 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \leq 0.$$

This interpretation also has meaning when we look at the objective function of a linear programming problem too. Indeed in the standard form we are required to maximise the function $\mathbf{c} \cdot \mathbf{x}$. If we write

$$z = \mathbf{c} \cdot \mathbf{x}$$

then we have the usual constraints $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}_n$ then are required to: *find a vector* $\mathbf{x}$ *which belongs to a hyperplane which is orthogonal to* $\mathbf{c}$ *and whose distance from the origin is as large as as possible*[4] *so that* $\mathbf{x}$ *is still contained in the region which lies in the intersection of the closed half spaces specified by the constraints.* We call the hyperplane given by the objective equation $z = \mathbf{c} \cdot \mathbf{x}$ the **objective hyperplane** or **objective function**.

To some extent it is easier to visualise what is being said above with a diagram. This can be done however only in dimensions 2 and 3 for obvious reasons.[5] Below are several examples of systems of constraints in either two or three variables together with a plot of the region which lies in the intersection of the closed half spaces specified by the constraints.

Example 3.1 This is a system of constraints in two dimensions,

$$\begin{aligned} &x_1 + x_2 \leq 5 \\ &2x_1 + x_2 \leq 8 \\ &x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

As mentioned earlier, a hyperplane in two dimensions is nothing more than a line. If we take the first constraint and change the inequality to an equality then it reads

$$x_2 = 5 - x_1.$$

[4]Even if this is negative.

[5]I have not yet found a way of drawing diagrams in 4 or more dimensional space which can be put effectively on paper.

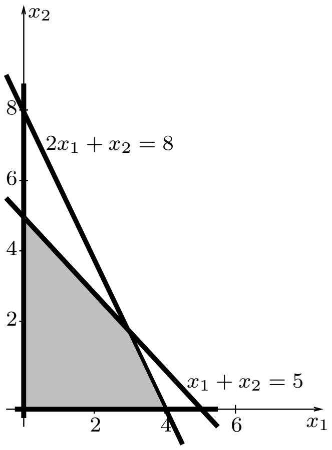


Figure 2: The region described by the equations $x_1 + x_2 \leq 5$, $2x_1 + x_2 \leq 8$ and $x_1, x_2 \geq 0$.

That is to say it describes the unique line which passes through the points $(x_1, x_2) = (0, 5)$ and $(x_1, x_2) = (5, 0)$ (and therefore has gradient -1). Or, said another way, it describes the line which is an orthogonal distance $5/\sqrt{2}$ for the origin with orthogonal unit vector

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}. \end{pmatrix}$$

When we put the inequality back in so that it reads

$$x_2 \leq 5 - x_1$$

then it describes all pairs (x_1, x_2) which are 'below' the line $x_2 = 5 - x_1$. A similar analysis of the second constraint requires us to consider all points which are 'below' the points described by the line $x_2 = 8 - 2x_1$. Finally taking account of the fact that both variables must be positive, we should only consider points which are 'to the right' of the line described by the vertical axis $x_1 = 0$ and 'above' the line described by the horizontal axis $x_2 = 0$. We thus arrive at the representation in Figure 2 for points satisfying the given constraints (shaded in grey).

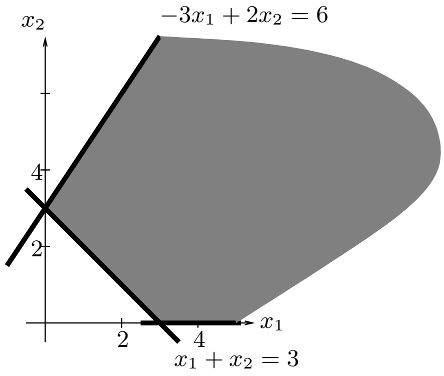


Figure 3: The region described by the equations $x_1 + x_2 \geq 3$, $-3x_1 + 2x_2 \leq 6$ and $x_1, x_2 \geq 0$.

Example 3.2 A similar analysis of the constraints

$$\begin{aligned} & x_1 + x_2 \geq 3 \\ & -3x_1 + 2x_2 \leq 6 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

yields Figure 3 for the points satisfying the constraints (shaded in grey).

Example 3.3 Now consider the following system of constraints

$$\begin{aligned} & 6x_1 + 4x_2 + 9x_3 \leq 36 \\ & 2x_1 + 5x_2 + 4x_3 \leq 20 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

Take for example the first constraint. Changing the inequality to an equality it may be written in the form $6x_1 + 4x_2 + 9x_3 = 36$ or equivalently in the form

$$\begin{pmatrix} \frac{6}{\sqrt{133}} \\ \frac{4}{\sqrt{133}} \\ \frac{9}{\sqrt{133}} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \frac{36}{\sqrt{133}}$$

suggesting that it is a plane whose orthogonal distance from the origin is $36/\sqrt{133}$ which has a unit normal vector equal to the first vector in the inner product above. It might seem difficult to imagine how one can easily draw this plane, however looking again at the equation of this plane in the form $6x_1 + 4x_2 + 9x_3 = 36$ one easily sees that it passes through the points $(x_1, x_2, x_3) = (6, 0, 0)$, $(0, 9, 0)$ and $(0, 0, 4)$ which is enough information to see how this plane appears in the positive orthant $x_1, x_2, x_3 \geq 0$. This appears as one of the planes in Figure 4. Returning to the original constraint $6x_1 + 4x_2 + 9x_3 \leq 36$, we are required to look for points (x_1, x_2, x_3) which are on the 'same side' of the plane $6x_1 + 4x_2 + 9x_3 = 36$ as the origin is.

The other plane in the diagram in Figure 4 represents the second constraint. Note that, for example, the constraint $x_1 \geq 0$ corresponds to the requirement that points (x_1, x_2, x_3) lie 'on the positive side' of the (x_2, x_3) plane. All points satisfying the constraints are again shaded in grey.

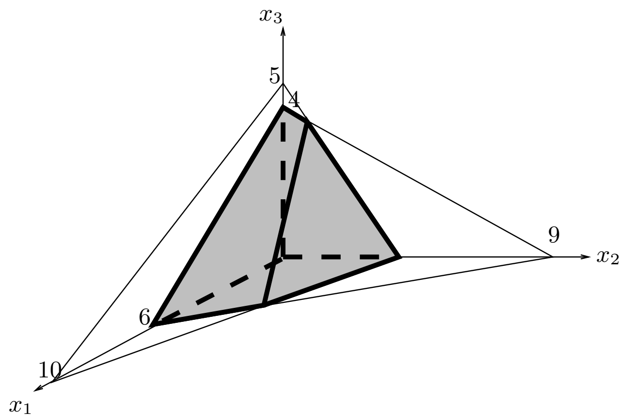


Figure 4: The region described by the equations $6x_1 + 4x_2 + 9x_3 \leq 36$, $2x_1 + 5x_2 + 4x_2 \leq 20$ and $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$.

Example 3.4 In our final example we leave the reader to examine the following system of equations,

$$\begin{aligned}
&x_1 + x_2 + x_3 \leq 1 \\
&2x_1 + 5x_2 + 3x_2 \leq 4 \\
&4x_1 + x_2 + 3x_3 \leq 2 \\
&x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.
\end{aligned}$$

We leave it as an exercise to associate them with the sketch in Figure 5.

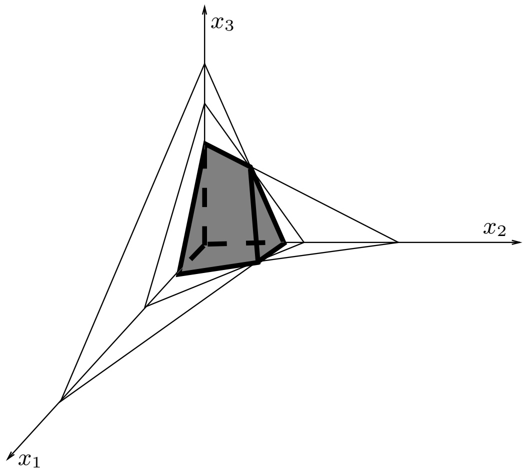


Figure 5: The region described by the equations $x_1 + x_2 + x_3 \leq 1$, $2x_1 + 5x_2 + 3x_2 \leq 4$, $4x_1 + x_2 + 3x_3 \leq 2$ and $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$.

Examples 3.1 and 3.2 examples above in the context of a linear programming problem, would require an objective function of the form $z = c_1x_1+c_2x_2$. For a fixed objective value z we thus have an equation of a straight line which passes through the points $(x_1, x_2) = (0, z/c_2)$ and $(z/c_1, 0)$. Maximising the value of z constitutes increasing the value of z, thereby moving the line in the direction of the vector

$$\begin{pmatrix} c_1 \\ c_2, \end{pmatrix}$$

until the value z^* for which there is no intersection between the line and region described by the constraints for all $z > z^*$.

The second two examples in the context of a linear programming problem would require an objective function taking the form $z = c_1x_1 + c_2x_2 + c_3x_3$ which is the equation of a plane. Analogously to the two dimensional case, maximising the value of z requires us to move the plane in the direction of its orthogonal such that z increases until we reach a value z^* beyond which there is no longer contact with the region described by the constraints.

How should one understand the canonical form of the linear programming problem in the above geometrical context? The introduction of slack variables increases the dimension of the problem and one looks for a vector $\mathbf{x}$ which solves a system of m linear equations in the positive orthant. Therefore any solution, written in terms of the original and slack variables, will lie at the intersection of the m specified hyperplanes. In general as there are less equations than variables, in dimension three and above, there must be an infinite number of intersection points whose intersection with the positive orthant describe the feasible region. This is difficult to see in higher dimensions, but note that two planes in three dimensional space with distinct normal vectors intersect on a line; that is to say common solutions to both planes are points on a line.

Alternatively, since an equation of the form $\mathbf{r} \cdot \mathbf{x} = d$ may be written as the two constraints $\mathbf{r} \cdot \mathbf{x} \leq d$ and $\mathbf{r} \cdot \mathbf{x} \geq d$, then the feasible region of a linear programming problem in canonical form is still the intersection of a finite number of closed half spaces. There is a special definition for the latter.

Definition 3.5 *The intersection of a finite number of closed half spaces is a* **closed polyhedron***.*

Note that the intersection of a finite number of closed spaces is still closed[6] which qualifies the use of the word 'closed' in the definition.

3.2 Feasible solutions and convexity

For linear programming problems in either canonical or standard form, let us define F, the **feasible domain**, to be the set of feasible solutions. From the previous section we argued that F consists of the intersection of a finite number of closed half spaces and hence constitutes a closed polyhedron.

In this section our objective is to give a mathematical description of the set F. Indeed we shall see that F is a closed **convex** polyhedron. To this end, let us recall the notion of a convex set in $\mathbb{R}^n$.

Definition 3.6 *A non-empty set S in $\mathbb{R}^n$ is a convex set if for all $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$ it holds that*

$$\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in S.$$

[6]Suppose that $C_1, ..., C_n$ are closed spaces. If $\{x_n : n \geq 1\}$ is a sequence of points in $\bigcap_{i=1}^n C_i$ with an accumulation point, say x. Then for each $i = 1, ..., n$ we have $\{x_n : n \geq 1\}$ is a sequence of points in C_i. Since, for each $i = 1, ..., n$, we are given that C_i is closed then it follows that $x \in C_i$. Hence $x \in \bigcap_{i=1}^n C_i$ and thus $\bigcap_{i=1}^n C_i$ is also closed.

Roughly speaking what this definition means that S is a convex set if, when choosing any two points in that set, the line joining those points is also contained in S. Here are some relevant examples of convex sets.

Example 3.7 *Suppose that for some given vector* $\mathbf{n}$ *in* $\mathbb{R}^n$ *and constant* $d \geq 0$, $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{n} \cdot \mathbf{x} = d\}$ *(i.e. S is a hyperplane in $\mathbb{R}^n$). Then S is a convex set. To see why note that if* $\mathbf{x}_1, \mathbf{x}_2 \in S$ *then since for* $\lambda \in (0, 1)$

$$\begin{aligned} \mathbf{n} \cdot (\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) &= \lambda \mathbf{n} \cdot \mathbf{x}_1 + (1-\lambda)\mathbf{n} \cdot \mathbf{x}_2 \\ &= \lambda d + (1-\lambda) d \\ &= d \end{aligned}$$

it follows that $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in S$.

Example 3.8 Suppose now that we take $S = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{n} \cdot \mathbf{x} \leq d\}$ where the quantities $\mathbf{n}$ and d are as in the previous example (i.e. S is the closed half space with boundary given by the hyperplane in the previous example). Then again S is convex. In this case the proof of this fact is almost the same as before. Indeed as before we may write

$$\begin{aligned} \mathbf{n} \cdot (\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) &= \lambda \mathbf{n} \cdot \mathbf{x}_1 + (1-\lambda)\mathbf{n} \cdot \mathbf{x}_2 \\ &\leq \lambda d + (1-\lambda) d \\ &= d \end{aligned}$$

showing that $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in S$ whenever $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$.

Example 3.9 Suppose that $S_1, ..., S_n$ are convex sets in $\mathbb{R}^m$ such that $\bigcap_{i=1}^m S_i \neq \emptyset$. Then $\bigcap_{i=1}^m S_i$ is also a convex set. In this case, one supposes that $\mathbf{x}_1, \mathbf{x}_2 \in \bigcap_{i=1}^n S_i$ then in particular $\mathbf{x}_1, \mathbf{x}_2 \in S_i$ for each $i = 1, 2, ..., m$ and hence by convexity of each of the latter sets, $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in S_i$ for any $\lambda \in (0, 1)$ and $i = 1, 2, ..., m$. Consequently, $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \bigcap_{i=1}^n S_i$ as required.

The last three examples can be used to prove in particular the following result.

Theorem 3.10 *Consider either the standard or the canonical form of the linear programming problem, (1) and (2). Then the set F of feasible solutions is a closed convex polyhedron.*

Proof. By the examples above and the earlier observation that F is the intersection of a finite number closed half spaces, the statement of the theorem follows. ■

3.3 Extreme points

Let us conclude this chapter on the geometry of the linear programming problem by hinting at *where* we should expect to find optimal solutions in the feasible region F.

Definition 3.11 *Given a non-empty convex set S, we say that $\mathbf{x} \in S$ is an* **extreme** *point in S if $\mathbf{x}$ is not an* **interior point** *of any line segment in S. That is to say, there exists no two vectors $\mathbf{x}_1, \mathbf{x}_2 \in S$ and $\lambda \in (0, 1)$ such that*

$$\mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2.$$

A good example to visualise extreme points is to think of the vertices of any **Platonic Solid** or indeed any n**-dimensional simplex**[7]. The following result is intuitively obvious and we exclude its proof.

Lemma 3.12 *Any closed convex polyhedron generated by the intersection of a finite number of half spaces (and hence any feasible region) has at most a finite number of extreme points.*

The importance of extreme points is that one should look for optimal solutions to linear programming problems by considering the extreme points of the feasible region as the following theorem confirms. The proof is omitted.

Theorem 3.13 *Consider a linear programming problem either in standard or canonical form. Let F be the space of feasible solutions. Then one of the following three scenarios necessarily holds.*

(i) *If $F \neq \emptyset$ and is bounded*[8] *and there is an optimal solution which occurs at an extreme point.*

(ii) *If $F \neq \emptyset$ and not bounded, but an optimal solution exists which occurs at an extreme point.*

(iii) *There exists no optimal solution in which case F is either unbounded or $F = \emptyset$.*

[7] Look them up on Google!

[8] One may understand bounded in this context to mean that $F \subset B(r)$ where $B(r) = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}|| < r\}$ for some $0 < r < \infty$. In other words, there exists a big enough hyper-sphere of radius r, centred at the origin such that F fits inside this sphere.

4 The fundamental theorem of linear programming

4.1 Basic solutions

Recall that, by the introduction of slack variables, the linear programming problem can be expressed in canonical form. That is to say one has a given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and given vector $\mathbf{b} \in \mathbb{R}^m$ and the objective is to find a vector $\mathbf{x} \in \mathbb{R}^n$ that will:

$$\begin{array}{l} \text{maximise } z := \mathbf{c} \cdot \mathbf{x} \\ \text{subject to:} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array} \tag{3}$$

Let us assume that $\mathbf{c}_1, ..., \mathbf{c}_n$ are the columns of $\mathbf{A}$ so that we may write the constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$ as

$$x_1\mathbf{c}_1 + \cdots + x_n\mathbf{c}_n = \mathbf{b}$$

where x_i is the i-th entry in the vector $\mathbf{x}$.

Note we may **assume without loss of generality** that $m \leq n$ and that $\text{rank}(\mathbf{A}) = m$. Otherwise when $m > n$ there are more linear equations than unknowns and by a series of linear row operations, one may use row reduction[9] to decrease the constraints $\mathbf{A}\mathbf{x} = \mathbf{b}$ to a system of linear equations whose rank can be no greater than the number of unknowns. Also if $\text{rank}(\mathbf{A}) < m$ then one may again reduce the constraints to a linear system where the rank is equal to the number of equations.

The assumption that $\text{rank}(\mathbf{A}) = m$ implies that there are precisely m columns of $\mathbf{A}$ which are linearly independent. For the sake of notational convenience it will be henceforth **assumed without loss of generality** that the last m columns of $\mathbf{A}$ are the ones which are linearly independent. Under such circumstances we may write $\mathbf{A} = (\mathbf{A}^\circ|\mathbf{B})$ where $\mathbf{B} \in \mathbb{R}^{m \times m}$ and $\text{rank}(\mathbf{B}) = m$ so that in particular $\mathbf{B}$ is invertible.

Definition 4.1 *Consider the canonical form of the linear programming problem (3). A* **basic solution** *to* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *is a vector* $\mathbf{x}$ *such that if* $\mathcal{I} = \{i : x_i \neq$

[9] Row reduction means taking linear combinations of the rows of $\mathbf{A}$ to produce as many rows of zeros as possible. Recall that the latter is only possible if there is linear dependence between rows.

$0\}$ *(that is to say the set of indices for which* $\mathbf{x}$ *has a non-zero entry) then the columns of* $\mathbf{A}$ *corresponding to the index set* $\mathcal{I}$ *are linearly independent. Note that necessarily* $|\mathcal{I}| \leq \mathrm{rank}(\mathbf{A})$*. If* $\mathbf{x}$ *is a basic solution and* $\mathbf{x} \geq \mathbf{0}_n$ *then we say that* $\mathbf{x}$ *is a* **basic feasible solution***.*

Another way of understanding what a basic solution is as follows. Suppose we decompose the matrix $\mathbf{A} = (\mathbf{A}^{\circ}|\mathbf{B})$ as above and consider solutions of the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{0}_{n-m} \\ \mathbf{x_B} \end{pmatrix} \tag{4}$$

where $\mathbf{x_B} \in \mathbb{R}^m$. In particular $\mathbf{B}\mathbf{x_B} = \mathbf{b}$ and since $\mathbf{B}$ is invertible, then $\mathbf{x_B}$ is uniquely identifiable as $\mathbf{x_B} = \mathbf{B}^{-1}\mathbf{b}$. Here $\mathbf{x}$ is a **basic solution**.

In the next theorem it is proved that basic solutions are necessarily extreme points. This conclusion suggests that **since optimal solutions occur at extreme points, one may try to find an optimal solution in the class of basic solutions.** The reader should naturally be concerned that the optimal solution may of course occur at an extreme point which is not a basic solution. However we shall dispel this concern in the next section. For now we conclude this section with the aforementioned theorem.

Theorem 4.2 *Suppose that* $\mathbf{x}$ *is a basic feasible solution of the linear programming problem (3). Then* $\mathbf{x}$ *is an extreme point of the feasible region.*

Proof. Without loss of generality we may arrange the given basic solution in the form (4). Let us assume for contradiction that $\mathbf{x}$ is not an extreme point so that we may write it as

$$\mathbf{x} = \lambda\mathbf{v} + (1-\lambda)\mathbf{w}$$

where $\lambda \in (0,1)$ and $\mathbf{v}$ and $\mathbf{w}$ belong to the feasible region of the linear programming problem (3). Let us write in the usual way v_j and w_j for the j-th elements of the vectors $\mathbf{v}$ and $\mathbf{w}$. Since $x_1, .., x_{n-m}$ are all equal to zero, it follows that for $j = 1, ..., n-m$,

$$0 = \lambda v_j + (1-\lambda)w_j.$$

Since $\lambda \in (0,1)$ and $v_j, w_j \geq 0$ by assumption, it follows that $v_j = w_j = 0$ for $j = 1, ..., n-m$.

Recalling that $\mathbf{v}$ is a feasible solution, we may now write it in the form

$$\mathbf{v} = \begin{pmatrix} \mathbf{0}_{n-m} \\ \mathbf{v_B} \end{pmatrix}$$

where $\mathbf{B}\mathbf{v_B} = \mathbf{b}$. Consequently we have

$$\mathbf{0}_m = \mathbf{B}\mathbf{v_B} - \mathbf{B}\mathbf{x_B} = (v_{n-m+1} - x_{n-m+1})\mathbf{c}_{n-m+1} + \cdots + (v_n - x_n)\mathbf{c}_n$$

and hence by the linear independence of $\mathbf{c}_{n-m+1}, ..., \mathbf{c}_n$ it follows that $v_j = x_j$ for $j = n-m+1, ..., n$. However this contradicts the assumption that $\mathbf{x}$ was not an extreme point (i.e. the assumption that $\lambda \in (0,1)$). ■

4.2 The fundamental theorem

Thus far we have seen that a bounded optimal solution, when it exists, can always be found at an extreme point and that all basic solutions are extreme points. This makes the class of basic solutions a good class in which to look for solutions. As mentioned earlier however, one may worry that the optimal solution may be an extreme point (or indeed any other point) which cannot be expressed as a basic solution. The fundamental theorem dismays this concern.

Theorem 4.3 *Consider the linear programming problem in canonical form (3). Then if there is a finite optimal feasible solution then there is an optimal basic feasible solution. (In other words, by Theorem 4.2, an extreme point corresponding to a finite optimal solution can be chosen to be a basic feasible solution).*

Proof. Suppose that $\mathbf{x}$ is an optimal feasible solution and without loss of generality we may write it in the form

$$\mathbf{x} = \begin{pmatrix} \mathbf{0}_{n-k} \\ \mathbf{y} \end{pmatrix}$$

where $n-k$ is the number of zeros entries in $\mathbf{x}$ and $\mathbf{y} > \mathbf{0}_k$. Next partition $\mathbf{A} = (\mathbf{H}|\mathbf{G})$ where the columns of $\mathbf{H}$ are equal to the first $n-k$ columns of $\mathbf{A}$ and the columns of $\mathbf{G}$ are equal to the last k columns of $\mathbf{A}$. Consequently we have

$$\mathbf{A}\mathbf{x} = \mathbf{G}\mathbf{y} = \mathbf{b}.$$

If it so happens that the columns of $\mathbf{G}$ are linearly independent (in which case it must necessarily be the case that $k \leq m$)[10] and the proof is complete.

Assume then that the columns of $\mathbf{G}$ are not linearly independent so that

$$\alpha_1 \mathbf{c}_{n-k+1} + \cdots + \alpha_k \mathbf{c}_n = \mathbf{0}_m \tag{5}$$

implies that for at least one $j = 1, ..., k$, we have $\alpha_j \neq 0$. Without loss of generality we can assume that $\alpha_j > 0$.[11] Now writing the equation $\mathbf{Gy} = \mathbf{b}$ in the form

$$y_1 \mathbf{c}_{n-k+1} + \cdots y_k \mathbf{c}_n = \mathbf{b},$$

since (5) holds, we have for any $\varepsilon > 0$ that

$$(y_1 - \epsilon\alpha_1)\mathbf{c}_{n-k+1} + \cdots + (y_k - \epsilon\alpha_k)\mathbf{c}_k = \mathbf{b}.$$

Now let us define a new vector

$$\mathbf{x}'_\epsilon = \mathbf{x} - \epsilon\alpha$$

where the first $n - k$ entries of α are zero and the remaining k entries are equal to $\alpha_1, ..., \alpha_k$ respectively. It follows that $\mathbf{Ax}'_\epsilon = \mathbf{b}$.

Note that for ϵ sufficiently small, it can be arranged that all the entries of $\mathbf{x}'_\epsilon$ are positive. Indeed let

$$\delta_1 = \max_{\alpha_j < 0} \frac{y_j}{\alpha_j} \text{ or } -\infty \text{ if all } \alpha_j \geq 0$$

$$\delta_2 = \min_{\alpha_j > 0} \frac{y_j}{\alpha_j}$$

and note that it is always true that $\delta_1 < 0 < \delta_2$ and that $\mathbf{x}'_\epsilon \geq 0$ for all $\epsilon \in [\delta_1, \delta_2]$. In other words, $\mathbf{x}'_\epsilon \geq 0$ is a feasible solution for $\epsilon \in [\delta_1, \delta_2]$. On account of the optimality of $\mathbf{x}$ we have for $\epsilon \in [\delta_1, \delta_2]$ that

$$\mathbf{c} \cdot \mathbf{x} \geq \mathbf{c} \cdot \mathbf{x}'_\epsilon = \mathbf{c} \cdot \mathbf{x} - \epsilon \mathbf{c} \cdot \alpha.$$

In other words

$$\epsilon \mathbf{c} \cdot \alpha \geq 0$$

[10] Otherwise one would have $k > m$ vectors in $\mathbb{R}^m$ which necessarily must have linear dependency.

[11] Note that at least one of the $\alpha_j \neq 0$. If that $\alpha_j < 0$ then multiplying (5) by -1 then relabelling each α_i by $-\alpha_i$ we have the existence of at least one $\alpha_j > 0$.

for all $\epsilon \in [\delta_1, \delta_2]$. This can only happen if $\mathbf{c} \cdot \alpha = 0$ and so for $\epsilon \in [\delta_1, \delta_2]$ we have shown that $\mathbf{x}'_\epsilon$ is also an optimal feasible solution! Now choose $\epsilon = \delta_2$ and note that by doing so, the number of zero terms in the vector $\mathbf{x}'_{\delta_2}$ has increased by at least one over and above the number of zeros in $\mathbf{x}$.

We may now take return to the very beginning of this proof working with $\mathbf{x}'_{\delta_2}$ instead of $\mathbf{x}$ and apply the same reasoning to deduce that either we have found an optimal basic feasible solution, or we can find another optimal feasible solution which has at least one more zero in its entries. In the latter case, an iterative procedure will eventually bring us to an optimal basic feasible solution. ■

Let us conclude this chapter by reiterating the main drive of the discussion.

For a given linear programming problem, an optimal solution (if it exists) will occur at an extreme point and by the fundamental theorem of linear programming this extreme point may be chosen to correspond to a basic feasible solution. We therefore need a mechanism which looks systematically through the basic feasible solutions in order to identify one which gives an optimal value.

Thus enter the simplex method!

5 The simplex method

In 1947 **George Dantzig** carried out an analysis of military programming and realised that many of the activities of large organisations could be viewed as linear programming problems. During a major international economic conference in 1949, Dantzig presented a method of solution for these problems which became known as the **simplex method**. It also became even clearer at the conference how wide were the possible applications.

The simplex method is an iterative procedure for solving high-dimensional[12] linear programming problems. The simplex method gives an algorithm by which one may search for an optimal basic feasible solution by proceeding from a vertex of the feasible region to an adjacent vertex, moving in a direction that improves the objective function.

The algorithm is quite straight-forward, but it is quite difficult to describe in a precise way, mainly because of the notation required to cope with different possible choices of basic feasible solutions and how to move between them.

We shall present the algorithm in this chapter, initially with three concrete examples, followed by some theoretical considerations for a general set-up and conclude with some discussion on degenerate solutions and extra care that one needs to take in certain situations.

5.1 Simplex algorithm in action: example 1

Let us consider the following explicit problem. Our objective is to:

$$
\begin{array}{l}
\text{maximise } z = 5x_1 + 6x_2 + 4x_3 \\
\text{subject to:} \\
x_1 + x_2 + x_3 \leq 10 \\
3x_1 + 2x_2 + 4x_3 \leq 21 \\
3x_1 + 2x_2 \leq 15 \\
x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.
\end{array}
$$

[12] High dimensional essentially means of dimension three or above since these problems cannot be illustrated graphically.

Step 1. Let us introduce slack variables u_1, u_2, u_3 allowing us to replace inequalities by equalities so that the constraints become

$$\begin{array}{llllll} x_1 + x_2 + x_3 & +u_1 & & & = 10 \\ 3x_1 + 2x_2 + 4x_3 & & +u_2 & & = 21 \\ 3x_1 + 2x_2 & & & +u_3 & = 15 \end{array}$$

and the new variables $u_1 \geq 0, u_2 \geq 0, u_3 \geq 0$. We may now take as our first basic feasible solution

$$(x_1, x_2, x_3, u_1, u_2, u_3) = (0, 0, 0, 10, 21, 15).$$

Take a moment and look back at the previous chapter to see why this is a basic feasible solution! Note that with this choice of basic feasible solution, our objective function $z = 0$. The reason for this is that our basic feasible solution has not included any of the variables x_1, x_2, x_3 and clearly including at least one of these variables into the basic feasible solution would increase the objective function from it current value.

Step 2. Recall the conclusion of the previous chapter, that it suffices to look through all the basic feasible solutions in order to find the solution to a linear programming problem when it is finite. Further from the previous step, we can already see that moving to another basic feasible solution which includes one of the variables x_1, x_2, x_3 would increase the objective function. Keeping the constraints tight, the simplest thing to do is consider introducing the variable x_1 into the **basis** and removing one of the **basis variables** u_1, u_2 or u_3. From the three main constraints, our options are:

$$\begin{array}{rll} (i) & x_1 : 0 \to 10 & \text{and } u_1 : 10 \to 0 \\ \text{or } (ii) & x_1 : 0 \to 7 & \text{and } u_2 : 21 \to 0 \\ \text{or } (iii) & x_1 : 0 \to 5 & \text{and } u_3 : 15 \to 0. \end{array}$$

Whichever of these options we choose, we must respect the feasibility of the solution. Below we see what the values of the basis variables would become in the three scenarios above:

$$\begin{array}{l} (i)\ (x_1, x_2, x_3, u_1, u_2, u_3) = (10, 0, 0, 0, -9, -15) \\ (ii)\ (x_1, x_2, x_3, u_1, u_2, u_3) = (7, 0, 0, 3, 0, -6) \\ (iii)\ (x_1, x_2, x_3, u_1, u_2, u_3) = (5, 0, 0, 5, 6, 0). \end{array}$$

Clearly only the third case respects the constraints of non-negativity and so we move to a new basis containing the variables (x_1, u_1, u_2).

Step 3. The next step is to reformulate the linear programming problem in terms of the new basis variables so that they appear only once in each constraint and have unit coefficients. A little linear algebra[13] gives us the following

$$\begin{array}{ccccccc} & \frac{1}{3}x_2 & +x_3 & +u_1 & & -\frac{1}{3}u_3 & =5 \\ & & 4x_3 & & +u_2 & -u_3 & =6 \\ x_1 & +\frac{2}{3}x_2 & & & & \frac{1}{3}u_3 & =5. \end{array}$$

Further, by writing the objective function in terms of the non-basic elements, we see that

$$z = 5(5 - \frac{1}{3}u_3 - \frac{2}{3}x_2) + 6x_2 + 4x_3 = 25 - \frac{5}{3}u_3 + \frac{8}{3}x_2 + 4x_3.$$

This tells us immediately two things. Firstly that with the new basis, the objective function is valued at $z = 25$. Secondly, if we are to change to a new basic feasible solution, it would only make sense to introduce the variables x_2 or x_3 as otherwise introducing u_3 would decrease the value of z.

Step 4. Let us now consider the two alternatives of introducing x_3 or x_2 into the basis.

If we introduce x_3 then our options are:

$$\begin{array}{rll} (i) & x_3 : 0 \to 5 & \text{and } u_1 : 5 \to 0 \\ \text{or } (ii) & x_3 : 0 \to \frac{3}{2} & \text{and } u_2 : 6 \to 0 \\ \text{or } (iii) & x_3 \in \mathbb{R} & \text{and } x_1 = 5. \end{array}$$

In order to respect the positivity of all basis variables as well as maximising the contribution to the objective function, the best choice from the above is (ii). In that case z increases by 6 units.

If we introduce x_2 then our options are:

$$\begin{array}{rll} (i) & x_2 : 0 \to 15 & \text{and } u_1 : 5 \to 0 \\ \text{or } (ii) & x_2 \in \mathbb{R} & \text{and } u_2 = 6 \\ \text{or } (iii) & x_2 : 0 \to \frac{15}{2} & \text{and } x_1 : 5 \to 0. \end{array}$$

In order to respect the positivity of all basis variables as well as maximising the contribution to the objective function, the best choice from the above is (iii). In that case z increases by 20.

[13] We did row 1 - $\frac{1}{3}$ row 3 and row 2 - row 1.

The conclusion of these calculations is that we should move to a new basic feasible solution

$$(x_1, x_2, x_3, u_1, u_2, u_3) = (0, \frac{15}{2}, 0, \frac{5}{2}, 6, 0).$$

giving value $z = 45$ to the objective function.

Step 5. Again writing the constraints so that the basic variables appear only once in each constraint with coefficient 1, and in particular keeping x_2 in the third row, we have[14]:

$$\begin{array}{llllll} -\frac{1}{2}x_1 & & +x_3 & +u_1 & & -\frac{1}{2}u_3 & = \frac{5}{2} \\ & & 4x_3 & & +u_2 & -u_3 & = 6 \\ \frac{3}{2}x_1 & +x_2 & & & & +\frac{1}{2}u_3 & = \frac{15}{2} \end{array}$$

Further, writing z in terms of the non-basic variables we obtain

$$z = 15 - \frac{5}{3}u_3 + \frac{8}{3}(\frac{15}{2} - \frac{8}{2}x_1 - \frac{1}{2}u_3) + 4x_3 = 45 - 4x_1 + 4x_3 - 3u_3.$$

The only way we can improve on the value of the objective function is to include the variable x_3 into the basis and this becomes the final step of the algorithm.

Step 6. Introducing x_3, our options are:

$$\begin{array}{rll} (i) & x_3 : 0 \to \frac{5}{2} & \text{and } u_1 : \frac{5}{2} \to 0 \\ \text{or } (ii) & x_3 : 0 \to \frac{3}{2} & \text{and } u_2 : 6 \to 0 \\ \text{or } (iii) & x_3 \in \mathbb{R} & \text{and } x_2 = \frac{15}{2}. \end{array}$$

The first option would not work as it would force u_2 to take a negative value. The second and third options are essentially the same when one takes account of the need for positive values. In conclusion the new basic feasible solution

$$(x_1, x_2, x_3, u_1, u_2, u_3) = (0, \frac{15}{2}, \frac{3}{2}, 1, 0, 0).$$

Increasing the value of z by 6 to 51.

Step 7. A quick calculation allows us to see that

$$z = 51 - 4x_1 - 2u_3 - u_2$$

[14]We did row operations: row1 $\to$ row1 - $\frac{1}{3}$row3 and row3 $\to$ $\frac{3}{2}$row3.

and there is no point to introduce any further variables into the basis as this would only decrease the value of the objective problem.

Step 8. The fundamental theorem of linear programming tells us that we must have found the optimal solution since we have found the optimal value of z amongst all the basic feasible solutions.

5.2 Simplex algorithm in action: example 2

In this second example, we present our computations in a more sophisticated way by making use of a standard tool called a **tableau**. Let us briefly introduce the latter. Suppose that we are given a linear programming problem with m equations in n variables in standard form which we then extend to canonical form with the introduction of slack variables which form the basis of an initial basic feasible solution. We can represent all this information in a tableau of the form given below.

	x_1	x_2	$\cdots$	x_n	u_1	u_2	$\cdots$	u_m	
u_1	a_{11}	a_{12}	$\cdots$	a_{1n}	1	0	$\cdots$	0	b_1
u_2	a_{21}	a_{22}	$\cdots$	a_{2n}	0	1	$\cdots$	0	b_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_m	a_{m1}	a_{m2}	$\cdots$	a_{mn}	0	0	$\cdots$	1	b_m
	$-c_1$	$-c_2$	$\cdots$	$-c_n$	0	0	$\cdots$	0	0

The left most column describes the basis variables and the bottom row gives the negative of the coefficients of the objective function in terms of the non-basic variables. The bottom right entry is the value of the objective function with the current choice of basis. The rest is self-explanatory.

Let us now state an example of a linear programming problem and, following similar logic to the example given in the previous section, we shall produce a series of tableaus which describe how the optimal solution is obtained.

Consider now the linear programming problem. Our objective is to:

$$\begin{aligned}&\text{maximise } z = 8x_1 + 9x_2 + 5x_3\\&\text{subject to:}\\&x_1 + x_2 + 2x_3 \le 2\\&2x_1 + 3x_2 + 4x_3 \le 3\\&6x_1 + 6x_2 + 2x_3 \le 8\\&x_1 \ge 0, x_2 \ge 0, x_3 \ge 0.\end{aligned}$$

Moving to the tableau for the canonical form we have

	$\mathbf{x_1}$	x_2	x_3	u_1	u_2	u_3	
u_1	1	1	2	1	0	0	2
u_2	2	3	4	0	1	0	3
$\mathbf{u_3}$	**6**	6	2	0	0	1	8
	-8	-9	-5	0	0	0	0

Following the logic of the previous example, we note that non-basic variables which have a negative coefficient in the final row will contribute a positive increase in the value of the objective if introduced to the basis. For each such variable x_j we compute $\theta_j = \min_i\{b_i/a_{ij} : b_j/a_{ij} \ge 0\}$. Of these x_j introduce the one for which $o_j\theta_j$, the increase in the objective, is the greatest where o_j is the coefficient of x_j when the objective function is written in terms of the non-basic variables and x_j. (In other words o_j is minus the entry in the j-th column of the last row and o_j).

Pivoting about the entry in bold in the above tableau so that x_1 is introduced into the basis and u_3 is removed from the basis (note that one will have chosen this change of basis variable on by considering feasibility) one obtains the following.

	x_1	$\mathbf{x_2}$	x_3	u_1	u_2	u_3	
u_1	0	0	5/3	1	0	$-1/6$	2/3
$\mathbf{u_2}$	0	**1**	10/3	0	1	$-1/3$	1/3
x_1	1	1	1/3	0	0	1/6	4/3
	0	-1	$-7/3$	0	0	4/3	32/3

Pivoting again about the entry shown in bold we obtain:

	x_1	x_2	x_3	u_1	u_2	u_3	
u_1	0	0	5/3	1	0	−1/6	2/3
x_2	0	1	10/3	0	1	−1/3	1/3
x_1	1	0	−3	0	−1	1/2	1
	0	0	1	0	1	1	11

Since all the coefficients of the non-basic variables in the bottom row are now positive, this means that an increase in their value would cause a decrease in the value of the objective function which is now identifiable as

$$z = 11 - x_3 - u_2 - u_3.$$

As an exercise, go back to the first example and see if you can summarise the key steps in a sequence of tableaus too. Alternatively consider the linear programming problem in the next section whose solution is given simply as a sequence of tableaus.

5.3 Simplex algorithm in action: example 3

Here is a third example of the simplex algorithm where we discover at the end that there is in fact no finite optimal solution. The problem is to:

$$\begin{aligned}
&\text{maximise } z = 2x_1 + 3x_2 + x_3 + x_4 \\
&\text{subject to:} \\
&x_1 - x_2 - x_3 \leq 2 \\
&-2x_1 + 5x_2 - 3x_3 - 3x_4 \leq 10 \\
&2x_1 - 5x_2 + 3x_4 \leq 5 \\
&x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.
\end{aligned}$$

Introducing slack variables u_1, u_2, u_3 we move straight to to the first tableau and from which we produce two further tableaus. As usual we have pivoted about the entry indicated in bold. The details are left to the reader.

	x_1	$\mathbf{x_2}$	x_3	x_4	u_1	u_2	u_3	
u_1	1	−1	−1	0	1	0	0	2
$\mathbf{u_2}$	−2	**5**	−3	−3	0	1	0	10
u_3	2	−5	0	3	0	0	1	5
	−2	−3	−1	−1	0	0	0	0

	$\mathbf{x_1}$	x_2	x_3	x_4	u_1	u_2	u_3	
$\mathbf{u_1}$	**3/5**	0	−8/5	−3/5	1	1/5	0	4
x_2	−2/5	1	−3/5	−3/5	0	1/5	0	2
u_3	0	0	−3	0	0	1	1	15
	−16/5	0	−14/5	−14/5	0	3/5	0	6

	x_1	x_2	x_3	x_4	u_1	u_2	u_3	
x_1	1	0	−8/3	−1	5/3	1/3	0	20/3
x_2	0	1	−5/3	−1	2/3	1/3	0	14/3
u_3	0	0	−3	0	0	1	1	15
	0	0	−34/3	−6	16/3	5/3	0	82/3

In this last tableau all the coefficients of x_3 are negative and hence it would be easy to create a solution with the **four** variables, x_1, x_2, x_3 and u_3 by setting $x_3 = M > 0$ and then taking

$$\begin{aligned} x_1 &= \tfrac{20}{3} + \tfrac{8}{3}M \\ x_2 &= \tfrac{14}{3} + \tfrac{5}{3}M \\ u_3 &= 15 + 3M. \end{aligned}$$

Note that all three of the above variables are, like x_3, strictly positive and hence we have constructed a feasible (but not necessarily basic) solution; and this reasoning is valid for any $M > 0$. Note also from the last row of the final tableau above that since one may write the objective as

$$z = \frac{34}{3}x_3 + 6x_4 - \frac{16}{3}u_1 + \frac{5}{3}u_2 + \frac{82}{3}$$

and hence including x_3 in the solution increases the objective by 34/3 for every unit increase in x_3. The preceding analysis would seem to indicate that we can set $x_3 = M > 0$ for any arbitrary large M and hence z can be made arbitrarily large with a feasible solution. In other words there is no bounded optimal solution.

5.4 The theory behind the simplex method

Given a linear programming problem in canonical form

$$\begin{aligned}&\text{maximise } z := \mathbf{c}\cdot\mathbf{x}\\&\text{subject to:}\\&\mathbf{A}\mathbf{x} = \mathbf{b}\\&\mathbf{x} \geq \mathbf{0}_n.\end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and we are at some point in the simplex algorithm were we are in possession of a basic feasible solution of the form

$$\mathbf{x} = \begin{pmatrix}\mathbf{0}_{n-m}\\ \mathbf{x_B}\end{pmatrix}$$

where $\mathbf{x_B} > \mathbf{0}_m$ represents the basic variables corresponding to the decomposition of the matrix $\mathbf{A} = (\mathbf{A}^\circ|\mathbf{B})$. Arranging the constraints so that the basic variables appear in only one row and with unit coefficient we obtain the tableau

$\mathbf{B}^{-1}\mathbf{A}^\circ$	$\vdots$	$\mathbf{I}_m$	$\mathbf{x_B}$
$\mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}^\circ - (\mathbf{c}^\circ)^{\mathrm{T}}$	$\vdots$	$\mathbf{0}_m^{\mathrm{T}}$	$z = \mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b}$

The last row gives the coefficients in the objective function when it is written in terms of the non-basic variables. To see where this comes from, suppose that we remove one basic variable from the current solution and introduce another variable giving a new solution

$$\mathbf{y} = \begin{pmatrix}\mathbf{y}^\circ\\ \mathbf{y_B}\end{pmatrix}$$

where $\mathbf{y}^\circ$ are the first $n-m$ entries and $\mathbf{y_B}$ are the last m entries corresponding to the old basis variables. Since

$$\mathbf{A}\mathbf{y} = \mathbf{A}^\circ\mathbf{y}^\circ + \mathbf{B}\mathbf{y_B} = \mathbf{b}$$

it follows that

$$\mathbf{y_B} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}^\circ\mathbf{y}^\circ.$$

If we also write the objective vector in obvious notation

$$\mathbf{c} = \begin{pmatrix}\mathbf{c}^\circ\\ \mathbf{c_B}\end{pmatrix}$$

then since the objective function valued with the initial basic feasible solution $z(\mathbf{x}) = \mathbf{c_B} \cdot \mathbf{x_B}$ and $\mathbf{B x_B} = \mathbf{b}$ so that $z(\mathbf{x}) = \mathbf{c_B} \cdot \mathbf{B}^{-1}\mathbf{b}$ it follows that the new value of the objective function

$$\begin{aligned} z(\mathbf{y}) &= \mathbf{c}^\circ \cdot \mathbf{y}^\circ + \mathbf{c_B} \cdot \mathbf{y_B} \\ &= \mathbf{c}^\circ \cdot \mathbf{y}^\circ + \mathbf{c_B} \cdot (\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{A}^\circ \mathbf{y}^\circ) \\ &= z(\mathbf{x}) + ((\mathbf{c}^\circ)^{\mathrm{T}} - \mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}^\circ)\mathbf{y}^\circ. \end{aligned}$$

The last equality shows us that the coefficients in the objective function when written in terms of non-basic variables which fill $\mathbf{y}^\circ$ are the entries of the vector $(\mathbf{c}^\circ)^{\mathrm{T}} - \mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}^\circ$. This explains where the final row of the tableau given above comes from.[15]

For $i = 1, ..., n-m$, write o_i for the i-th element of the vector $(\mathbf{c}^\circ)^{\mathrm{T}} - \mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}^\circ$ so that

$$z(\mathbf{y}) = z(\mathbf{x}) + \sum_{i=1}^{n-m} o_i y_i.$$

If we see that $o_j > 0$ for some $j = 1, ..., n-m$ then it would be profitable to take our new basis $\mathbf{y}$ so that

$$\mathbf{y}^\circ = (0, \cdots, y_j, 0, \cdots, 0) \text{ where } y_j > 0$$

and remove one of the elements of the basis variables in $\mathbf{x_B}$. This would ensure that we have improved on the value of the objective function since then $z(\mathbf{y}) = z(\mathbf{x}) + o_j y_j > z(\mathbf{x})$. It is of course necessary to do so in a way that $\mathbf{y}$ remains in the feasible region. (This corresponds to considering how one may introduce the variable y_j into each of the m equations given by $\mathbf{Ay} = \mathbf{b}$ whilst respecting the requirement that $\mathbf{y} \geq \mathbf{0}_n$).

The essence of the algorithm is thus updating the basis of the feasible solution in such a way that one is always introducing a variable from the non-basic variables to create a new basic feasible solution such that the value of the objective function increases. One keeps a tab on which variables to introduce by always writing the objective function in terms of non-basic variables. Eventually, all the coefficients o_i will be negative in which case one has obtained the optimal basic feasible solution. The fundamental theorem of linear programming then tells us that we have found an optimal solution.

[15] Don't forget that one negates the coefficients before filling them in the tableau.

5.5 Degeneracy and cycling

There is always the possibility that, when introducing a new variable into the basis, one of the basic variables coincidentally becomes equal to zero. We define such basic feasible solutions as **degenerate**. Recall that a basic feasible solution is an extreme point of the convex polyhedron describing the feasible region and the simplex method is an algorithm which systematically moves through extreme points of the feasible region.

It may so happen that the simplex method brings you to an extreme point which lies on the the hyperplane $x_i = 0$ for some $i = 1, ..., n$. (Recall that there are n constraining hyperplanes to the feasible region which are given by the condition $\mathbf{x} \geq \mathbf{0}_n$). In that case, the basic feasible solution will be such that x_i belongs to the basis and yet $x_i = 0$.

As an exercise, one may draw a picture and consider the extreme points of a linear programming problem with the constraints

$$\begin{aligned} &x_1 - x_2 \leq 2 \\ &2x_1 + x_2 \leq 4 \\ &-3x_1 + 2x_2 \leq 6 \\ &x_1, x_2 \geq 0. \end{aligned}$$

For this reason, one can say that although the dimension of the basis is always equal to m =rank($\mathbf{A}$) it may happen that the number of zeros in a basic feasible solution may be greater than $n - m$.

Note that the analysis in the previous section concerning the introduction of a non-basic variable y_j to improve the objective value assumed that $\mathbf{x_B} > \mathbf{0}_m$ and $y_j > 0$. The analysis is not necessarily the same if the solution $\mathbf{x_B}$ is degenerate.

Because of the phenomenon of degeneracy, it is possible to slip into a cycle with the simplex method as we now explain. Suppose one has a basic feasible solution which is degenerate. In that case, recalling that the simplex tableau may always be rearranged in the form

$\mathbf{B}^{-1}\mathbf{A}^\circ$	$\vdots$	$\mathbf{I}_m$	$\mathbf{x_B}$
$\mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}^\circ - (\mathbf{c}^\circ)^{\mathrm{T}}$	$\vdots$	$\mathbf{0}_m^{\mathrm{T}}$	$z = \mathbf{c_B}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{b}$

then a degenerate solution implies that one of the entries of $\mathbf{x_B}$ is equal to zero. Then removing the corresponding element of the basis and introducing a new basic variable will have the effect that the new basic variable will also be zero. In other words, after pivoting, the next basic feasible solution will be degenerate too. Further, the objective will remain at the same value after pivoting since the new basic variable, being equal to zero, contributes no value of the objective function. Consequently, the simplex algorithm will move a degenerate solution to a degenerate solution without increasing the objective value and ultimately one ends up cycling between degenerate solutions. Most linear programming problems do not exhibit this phenomena however.

5.6 The two-phase method

There is a particular point that we need to be aware of in what we have discussed thus far. In the three examples of Sections 5.1, 5.2 and 5.3 **we were able to specify an initial basic feasible solution** by simply setting the slack variables equal to the entries of the vector $\mathbf{b}$. The above theoretical considerations also show how one moves from one basic feasible solution to another *under the assumption that one has a basic feasible solution in the first place*.

What would happen if it is not immediately obvious how to form an initial basic feasible solution?

Here is a very likely scenario in which one might find this to be an issue. Suppose that at least one of the entries of the vector $\mathbf{b}$ is strictly negative. Naively following what we have done in the two examples would tell us to set the slack variable as equal to that value. However one should of course not forget about the constraints of positivity which would then be violated! So how does one set up the initial basic feasible solution from which to proceed with the simplex method? This is where the two-phase method can prove to be useful.

Phase I. The first step of the two-phase method is to introduce **even more**[16] variables which are known as **artificial variables**. The way to do this is via the following series of actions.

(i) Arrange the linear programming problem in standard form so that the constraints are given as $\mathbf{Ax} \leq \mathbf{b}$ (even if the entries in the vector $\mathbf{b}$ are negative) and $\mathbf{x} \geq \mathbf{0}_n$. We assume as usual that $\mathbf{A} \in \mathbb{R}^{m\times n}$.

[16] In this context 'even more' means over and above the slack variables

(ii) Introduce slack variables in the usual way.

(iii) For each row that has a negative entry in $\mathbf{b}$, multiply by -1 producing constraints of a new linear programming problem, say $\mathbf{A}'\mathbf{x}' = \mathbf{b}'$ and $\mathbf{x}' \geq \mathbf{0}_{m+n}$, where $\mathbf{x}'$ includes the slack variables and now $\mathbf{b}' \geq \mathbf{0}_m$. Note at this point, if i is a row which has been multiplied through by -1, the slack variables u_i has coefficient -1 and hence cannot be set as equal to $b'_i = -b_i$ as part of a basis solution.

(iv) Introduce to each row a new artificial variable, say w_i for $i = 1, ..., m$.

The principle action of phase I is to solve the following auxiliary linear programming problem.

Let $\mathbf{A}'' = (\mathbf{A}'|\mathbf{I}_m)$,

$$\mathbf{x}'' = \begin{pmatrix} \mathbf{x}' \\ \mathbf{w} \end{pmatrix}$$

and write $\mathbf{1}_m$ for the vector in $\mathbb{R}^m$ which has all of its entries equal to one.

$$\begin{aligned} &\text{maximise } z'' = -\mathbf{1}_m \cdot \mathbf{w} \\ &\text{subject to:} \\ &\mathbf{A}''\mathbf{x}'' = \mathbf{b}' \\ &\mathbf{x}'' \geq \mathbf{0}_{n+2m}. \end{aligned}$$

The importance of this linear programming problem is that since it is clear that the last condition also implies that $\mathbf{w} \geq \mathbf{0}_m$, then the maximal value of z'' is in fact zero. Further, if we solve it using the simplex method, then the solution which will achieve this maximal value of zero will necessarily be a basic solution and it will necessarily have $\mathbf{w} = \mathbf{0}_m$. In other words, solving the above auxiliary linear programming problem will provide a basic feasible solution to the system

$$\begin{aligned} &\mathbf{A}'\mathbf{x}' = \mathbf{b}' \\ &\mathbf{x}' \geq \mathbf{0}_{n+m}, \end{aligned}$$

which is exactly what we are after to proceed to phase II.

Phase II. Taking the solution of the auxiliary linear programming problem we may now return to the original linear programming problem in canonical form using it as an initial basic solution from which to proceed in the usual way with the simplex method. One should take care however, to immediately write the original objective function in terms of the the non-basic variables at the end of phase I in order to proceed to the first tableau.

Roughly speaking, the two-phase method introduces an additional set of variables for the purpose of establishing an initial basic feasible solution in the easiest way of 'setting $\mathbf{x_B} = \mathbf{b}$' (once things have been set up in the right way of course) and then 'pushing' this basic feasible solution (with the help of the simplex algorithm) onto another basic feasible solution in terms of the original set of variables.

5.7 An example of the two-phase method

These theoretical considerations of the the two-phase method are perhaps best illustrated with an example. Consider the problem:

$$\begin{array}{l} \text{maximise } z = 3x_1 + x_2 \\ \text{subject to:} \\ x_1 + x_2 \leq 6 \\ 4x_1 - x_2 \geq 8 \\ 2x_1 + x_2 = 8 \\ x_1, x_2 \geq 0 \end{array}$$

This problem has a mixture of constraint types [17] and after a slack variable x_3 is added to the first constraint, a surplus variable x_4 subtracted from the second, we have the following problem

$$\begin{array}{l} \text{maximise } z = 3x_1 + x_2 \\ \text{subject to:} \\ x_1 + x_2 + x_3 \qquad = 6 \\ 4x_1 - x_2 \qquad - x_4 = 8 \\ 2x_1 + x_2 \qquad\qquad = 8 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array}$$

and there is no obvious basic feasible solution.

In order to overcome this, non negative **artificial variables**, x_5 and x_6 are added to the left hand side of the second and third constraints. Note that this is in slight contradiction with the method described in the previous section which dictates that we should add an artificial variable to the first

[17] Note that this problem is slightly different to the ones discussed up until now in the sense that some of the constraints are slack and some are already tight. The reader will learn with experience that one only need introduce as many slack variables as there are slack constraints and proceed in the usual way with the simplex method.

constraint too. However a little thought regarding how we proceed in this example reveals that in the end this will make no difference. This creates the **auxiliary** linear programming problem,

$$\begin{aligned}
&\text{maximise } z'' = -x_5 - x_6 \\
&\text{subject to:} \\
&x_1 + x_2 + x_3 = 6 \\
&4x_1 - x_2 - x_4 + x_5 = 8 \\
&2x_1 + x_2 + x_6 = 8 \\
&x_1, ..., x_6 \geq 0
\end{aligned}$$

In order for the augmented system to correspond to the original constraint system, the artificial variables must be zero. This is achieved by using the Simplex method to minimise their sum (or maximise the negative sum) and this is what we called **Phase I**. If this terminates with all artificial variables at zero, a basic feasible solution to the original constraint system will have been obtained. The Simplex method is then used to maximise the original objective in **Phase II** of the procedure.

Of course, in this example it would be a simple matter to use one of the constraints to eliminate x_2, or x_1 from the other constraints and objective, leaving a trivial problem. In general however, this would not be at all convenient with many variables and many constraints. This simple example shows how the general procedure would be used.

Phase I

	$\mathbf{x_1}$	x_2	x_3	x_4	x_5	x_6	
x_3	1	1	1	0	0	0	6
$\mathbf{x_5}$	**4**	-1	0	-1	1	0	8
x_6	2	1	0	0	0	1	8
	-6	0	0	1	0	0	-16

Now x_1 enters and x_5 leaves the set of basic variables. We shall not bother to transform the x_5 column for the next tableau as the artificial variable will not be considered to enter the basis again.

	x_1	$\mathbf{x_2}$	x_3	x_4	x_5	x_6	
x_3	0	5/4	1	1/4	-	0	4
x_1	1	-1/4	0	-1/4	-	0	2
$\mathbf{x_6}$	0	**3/2**	0	1/2	-	1	4
	0	-3/2	0	-1/2	-	0	-4

Now x_2 enters and x_6 leaves the set of basic variables and we reach the final tableau for phase I. In this tableau we have added an extra row for the coefficients of the objective function z in terms of the basis variables which conclude phase I and which initiate phase II.

	x_1	x_2	x_3	$\mathbf{x_4}$	x_5	x_6	
x_3	0	0	1	-1/6	-	-	2/3
x_1	1	0	0	-1/6	-	-	8/3
$\mathbf{x_2}$	0	1	0	**1/3**	-	-	8/3
z''	0	0	0	0	-	-	0
z	0	0	0	-1/6	-	-	32/3

The last row indicates that we can improve the value of z by introducing x_4 into the basis which leads to the following tableau (now for the original problem in canonical form - ie we have now entered phase II).

	x_1	x_2	x_3	x_4	
x_3	0	1/2	1	0	2
x_1	1	1/2	0	0	4
x_4	0	3	0	1	8
	0	1/2	0	0	12

From this final table we see that there are no further variables which can be introduced into the basis in order to improve the value function and we have reached an optimal solution with the basis $x_1 = 4, x_2 = 0, x_3 = 2, x_4 = 8$ and $z = 12$.

6 Duality

6.1 Introduction

Recall the Betta Machine Product problem.

$$\begin{aligned}
&\text{Maximise } z = 15x_1 + 20x_2 \\
&\text{subject to:} \\
&5x_1 + 8x_2 \leq 16,000 \\
&5x_1 + 4x_2 \leq 14,000 \\
&x_1 + 3x_2 \leq 5,000 \\
&x_1 \geq 0, x_2 \geq 0.
\end{aligned}$$

The variables x_1 and x_2 represent quantities of products 1 and 2 whilst the constraints are concerned with the available numbers of minutes of time for casting, machining and assembly respectively.

Multiplying the first constraint by 3 gives:

$$15x_1 + 24x_2 \leq 48,000.$$

Comparing this with the equation for z shows that we have found an upper bound for z as we must have $z \leq 48,000$. Could we use a similar trick to find a tighter bound? By adding twice the first constraint to the second we obtain

$$15x_1 + 20x_2 \leq 46,000$$

from which we can deduce that $z \leq 46,000$.

How can we find the best possible bound on z ? Suppose we follow the above technique and look for linear combinations of the three constraints that would give us the best upper bound. Let us add multiples of the linear constraints together using the multipliers w_1, w_2 and w_3. We would need

$$\begin{aligned}
&5w_1 + 5w_2 + w_3 \geq 15 \\
&8w_1 + 4w_2 + 3w_3 \geq 20 \\
&w_1, w_2, w_3 \geq 0
\end{aligned}$$

giving an upper bound

$$16,000w_1 + 14,000w_2 + 5,000w_3.$$

Note that the positivity of the variables w_1, w_2 and w_3 is necessary to preserve the inequality in each of the original constraints.

Hence, if we wish to find the smallest bound possible, we have to solve another linear programming problem; namely

$$\begin{aligned}
&\text{minimise } 16,000w_1 + 14,000w_2 + 5,000w_3 \\
&\text{subject to:} \\
&5w_1 + 5w_2 + w_3 \geq 15 \\
&8w_1 + 4w_2 + 3w_3 \geq 20 \\
&w_1, w_2, w_3 \geq 0
\end{aligned}$$

This is called the **dual problem**. To distinguish the two problems, the original is called the **primal problem**.

6.2 Symmetric dual

Consider the **standard maximisation problem** (P): given, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$,

$$\begin{aligned}
&\text{maximise } \mathbf{c} \cdot \mathbf{x} \\
&\text{subject to:} \\
&\mathbf{Ax} \leq \mathbf{b} \\
&\mathbf{x} \geq \mathbf{0}_n.
\end{aligned}$$

Its **symmetric** dual is the **standard minimisation problem** (D): given $\mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{n \times m}$, $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$,

$$\begin{aligned}
&\text{minimise } \mathbf{b} \cdot \mathbf{w} \\
&\text{subject to:} \\
&\mathbf{A}^{\mathrm{T}}\mathbf{w} \geq \mathbf{c} \\
&\mathbf{w} \geq \mathbf{0}_m.
\end{aligned}$$

Remarks

1. The primal has n variables and m constraints whilst the dual has m variables and n constraints. In the last chapter we had $n \geq m$ and m =rank($\mathbf{A}$) because the number of variables, n, included slack and surplus variables. In the present situation slack and surplus variables have not been counted so either n or m could be the larger.

2. **Any** problem may be put into form (P) **or** (D). To do this you may need to multiply by -1 or to write an equation as two inequalities. Note it is not required here that the elements of $\mathbf{b}$ should be non negative.

3. It is a **very important** point that the dual of the dual is the primal.

4. To appreciate why the dual problem we have identified is **symmetric** it is necessary to consider the **asymmetric** dual problem given below.

6.3 The asymmetric dual

Suppose that instead of starting with the linear programming problem in standard form, we started with the linear programming problem in canonical form. That is to say: given, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$,

$$\begin{array}{l} \text{maximise } \mathbf{c}\cdot\mathbf{x} \\ \text{subject to:} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array}$$

Following previous advice, we can write this linear programming problem in the form

$$\begin{array}{l} \text{maximise } \mathbf{c}\cdot\mathbf{x} \\ \text{subject to:} \\ \binom{\mathbf{A}}{-\mathbf{A}}\mathbf{x} \leq \binom{\mathbf{b}}{-\mathbf{b}} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array}$$

Now following the formulation for the dual but partitioning the dual vector

$$\mathbb{R}^{2m} \ni \mathbf{y} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix},$$

we have dual problem

$$\begin{array}{l} \text{minimise } \binom{\mathbf{b}}{-\mathbf{b}}\cdot\binom{\mathbf{u}}{\mathbf{v}} \\ \text{subject to:} \\ (\mathbf{A}^{\mathrm{T}}| - \mathbf{A}^{\mathrm{T}})\binom{\mathbf{u}}{\mathbf{v}} \geq \mathbf{c} \\ \mathbf{u},\mathbf{v} \geq 0. \end{array}$$

For convenience we can write $\mathbf{z} = \mathbf{u} - \mathbf{v}$ and now the dual can be more conveniently written as

$$\begin{array}{l} \text{minimise } \mathbf{b}\cdot\mathbf{z} \\ \text{subject to:} \\ \mathbf{A}^{\mathrm{T}}\mathbf{z} \geq \mathbf{c} \end{array}$$

noting in particular that $\mathbf{z} \in \mathbb{R}^m$. Hence the dual of the canonical linear programming problem **has no positivity constraint** and hence is **asymmetric** to its primal.

6.4 Primal to the Dual via the diet problem

Here is an example of how we may interpret duality by considering a classic diet problem.

Suppose that there are nutrients N_i, $i = 1, ..., m$ and foods F_j, $j = 1, ..., n$ with food costs c_j per unit of food F_j. Suppose further that each food F_j contains A_{ij} units of of nutrient N_i. The diet problem is to minimise the cost of the diet subject to the constraints that the diet provides the minimal nutritional requirements which are specified by the quantities b_i; $i = 1, ..., m$ for each nutrient N_i. If $(x_1, ..., x_n)$ represents the quantity of the respective foods $F_1, ..., F_n$ in the diet (and clearly we must have $x_j \geq 0$ for all $j = 1, ..., n$), then the cost of the diet becomes $\sum_j c_j x_j$. On the other hand, to respect the minimum requirement of nutrient N_i we must also have that $\sum_j A_{ij} x_j \geq b_i$. In other words the problem becomes:

$$\begin{array}{l} \text{minimise } \mathbf{c} \cdot \mathbf{x} \\ \text{subject to} \\ \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n. \end{array}$$

The above we may consider as our **primal problem**.

Now suppose that there is a company which, instead of selling the foods $F_1, ..., F_n$, sells the raw nutrients (for example in the form of pills) at respective costs $y_1, ..., y_m$. Since food F_j contains A_{ij} units of nutrient of N_i, then $\sum_i y_i A_{ij}$ is the equivalent cost of food F_j for $j = 1, ..., n$. The manufacturer of the raw nutrients wishes to maximise the profits of selling the recommended dietary intake, $\sum_i b_i y_i$, whilst competing with the primal diet. In other words, the equivalent cost of food F_j should not be more expensive than the primal cost of food F_j so that $\sum_i y_i A_{ij} \leq c_j$.

From the perspective of the manufacturer of the raw nutrients the problem is to:

$$\begin{array}{l} \text{maximise } \mathbf{b} \cdot \mathbf{y} \\ \text{subject to} \\ \mathbf{y}^{\mathrm{T}}\mathbf{A} \leq \mathbf{c}^{\mathrm{T}} \\ \mathbf{y} \geq \mathbf{0}_m. \end{array}$$

We see that the manufacturer of the raw nutrients is thus concerned with the **dual problem**.

Intuitively we would expect that any feasible solution to the above problem, which would specify a profit for a particular diet of the raw nutrients $N_1, ..., N_m$, should not be greater than any feasible solution to the primal problem which is the cost of a diet using the foods $F_1, ..., F_n$. That is to say, we would expect $\mathbf{b} \cdot \mathbf{y} \leq \mathbf{c} \cdot \mathbf{x}$ where $\mathbf{x}$ and $\mathbf{y}$ are feasible solutions to the primal and dual respectively. Moreover in that case, it would seem sensible that both have optimal solutions if there exist feasible solutions, $\mathbf{x}$ and $\mathbf{y}$, such that $\mathbf{b} \cdot \mathbf{y} = \mathbf{c} \cdot \mathbf{x}$. This is the beginning of duality theory which consumes the remainder of this chapter.

6.5 The Duality Theorem

We shall henceforth continue our analysis for the case of the symmetric duality. That is to say we shall analyse the relationship between the primal (P) and dual (D) given in Section 6.2.

As we have seen in the preliminary example, the dual objective gives a bound for the primal objective (and vice versa). This is exactly what the Weak Duality Theorem says.

Theorem 6.1 (Weak Duality Theorem) *Suppose that* $\mathbf{x}_0$ *is a feasible solution to the primal problem (P) and* $\mathbf{y}_0$ *is a feasible solution to its dual (D). Then*

$$\mathbf{c} \cdot \mathbf{x}_0 \leq \mathbf{b} \cdot \mathbf{y}_0.$$

Proof. Since $\mathbf{A}\mathbf{x}_0 \leq \mathbf{b}$ and $\mathbf{y}_0 \geq \mathbf{0}_m$ it follows that

$$\mathbf{y}_0 \cdot \mathbf{A}\mathbf{x}_0 \leq \mathbf{y}_0 \cdot \mathbf{b}$$

Similarly, since $\mathbf{A}^{\mathrm{T}}\mathbf{y}_0 \geq \mathbf{c}$ and $\mathbf{x}_0 \geq \mathbf{0}_n$ it follows that $\mathbf{x}_0 \cdot \mathbf{A}^{\mathrm{T}}\mathbf{y}_0 \geq \mathbf{x}_0 \cdot \mathbf{c}$; or transposing both sides, that

$$\mathbf{y}_0 \cdot \mathbf{A}\mathbf{x}_0 \geq \mathbf{c} \cdot \mathbf{x}_0.$$

Comparing the established inequalities we see immediately that

$$\mathbf{c} \cdot \mathbf{x}_0 \leq \mathbf{b} \cdot \mathbf{y}_0$$

as required. ∎

Since in the proof above, $\mathbf{x}_0$ and $\mathbf{y}_0$ are abitrary, it follows that

$$\sup_{\mathbf{x}\in F(P)} \mathbf{c}\cdot\mathbf{x} \leq \inf_{\mathbf{y}\in F(D)} \mathbf{b}\cdot\mathbf{y} \tag{6}$$

where $F(P)$ and $F(D)$ are the feasible regions of (P) and (D) respectively. Two consequences follow immediately from this observation which we give below.

Corollary 6.2 *If (P) has a feasible solution but no bounded optimal solution then (D) has no feasible solution. Since (P) is the dual of (D) it also follows that if (D) has a feasible solution but no bounded optimal solution then (P) has no feasible solution.*

Proof. When (P) has an unbounded optimal solution then (6) shows that, should it be the case that $F(D) \neq \emptyset$, $\inf_{\mathbf{y}\in F(D)} \mathbf{b}\cdot\mathbf{y} = \infty$ which is a contradiction. It must therefore follow that $F(D) = \emptyset$. ■

Corollary 6.3 *Suppose there exist feasible solutions for (P) and (D), say* $\mathbf{x}_0$ *and* $\mathbf{y}_0$ *respectively, such that* $\mathbf{c}\cdot\mathbf{x}_0 = \mathbf{y}_0\cdot\mathbf{b}$ *then necessarily* $\mathbf{x}_0$ *and* $\mathbf{y}_0$ *are optimal solutions for (P) and (D) respectively.*

Proof. Note that for any pair of solutions $\mathbf{x}_0$ and $\mathbf{y}_0$ we have

$$\mathbf{c}\cdot\mathbf{x}_0 \leq \sup_{\mathbf{x}\in F(P)} \mathbf{c}\cdot\mathbf{x} \leq \inf_{\mathbf{y}\in F(D)} \mathbf{b}\cdot\mathbf{y} \leq \mathbf{b}\cdot\mathbf{y}_0$$

and hence if $\mathbf{c}\cdot\mathbf{x}_0 = \mathbf{y}_0\cdot\mathbf{b}$ then all of the above inequalities are necessarily equalities an in particular we see immediately that $\mathbf{x}_0$ and $\mathbf{y}_0$ are optimal. ■

The main result of this section is the following.

Theorem 6.4 (Duality Theorem) *The primal problem (P) has a finite optimal solution if and only if the dual problem (D) does too in which case the optimal value of their objective functions are the same.*

The way we will prove the Duality Theorem is to consider first four auxiliary results. The first two of these four are classic results coming from convex analysis.

The first result makes the somewhat intuitively obvious statement that for any point which lies outside of a closed convex set, it is possible to pass a hyperplane between that point and the convex set.

Lemma 6.5 (Separating Hyperplane Lemma) *Suppose that C is a closed convex set in $\mathbb{R}^n$ and that $\mathbf{b} \notin C$. Then there exists $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{y} \cdot \mathbf{b} < \mathbf{y} \cdot \mathbf{z}$ for all $\mathbf{z} \in C$.*

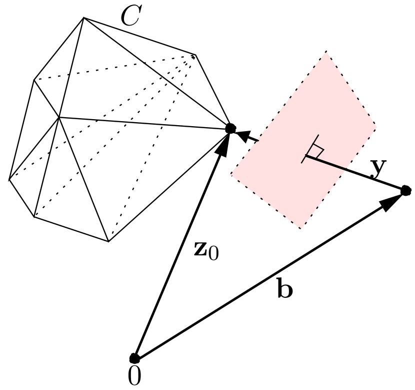


Figure 6: A diagrammatic representation of elements of the Separating Hyperplane Lemma.

Proof. Let $\overline{B}(\mathbf{b}, r)$ be a closed hyper-sphere whose centre is at $\mathbf{b}$ and radius r is sufficiently large that $C \cap \overline{B}(\mathbf{b}, r) \neq \emptyset$. Now consider the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(\mathbf{z}) = \{(\mathbf{b} - \mathbf{z}) \cdot (\mathbf{b} - \mathbf{z})\}^{1/2} = ||\mathbf{b} - \mathbf{z}||$$

for $\mathbf{z} \in \mathbb{R}^n$. As f is a continuous function and $C \cap \overline{B}(\mathbf{b}, r)$ is a closed bounded convex domain[18] then f is bounded and there exists a $\mathbf{z}_0 \in C \cap \overline{B}(\mathbf{b}, r)$ such that [19]

$$r \geq f(\mathbf{z}_0) = \inf\{f(\mathbf{z}) : \mathbf{z} \in C \cap \overline{B}(\mathbf{b}, r)\} =: \delta > 0.$$

Note that the first inequality is a consequence of the fact that $\mathbf{z} \in C \cap \overline{B}(\mathbf{b}, r) \subseteq \overline{B}(\mathbf{b}, r)$ can never be further than distance r from $\mathbf{b}$ and the last inequality is due to the fact that $\mathbf{b} \notin C$. Hence for all $\mathbf{z} \in C \cap \overline{B}(\mathbf{b}, r)$

$$\delta = ||\mathbf{b} - \mathbf{z}_0|| \leq ||\mathbf{b} - \mathbf{z}|| \tag{7}$$

[18] Recall that the intersection of two closed domains is closed and that the non-empty intersection of two convex domains is convex

[19] Google Weierstrauss' Theorem!

and otherwise for $\mathbf{z} \in C \backslash \overline{B}(\mathbf{b}, r)$

$$\delta = ||\mathbf{b} - \mathbf{z}_0|| \leq r \leq ||\mathbf{b} - \mathbf{z}||.$$

In other words (7) holds for all $\mathbf{z} \in C$ (even when C is an unbounded domain). As C is a convex domain, it follows in particular that for each $\lambda \in (0, 1)$ and $\mathbf{z} \in C$

$$(\mathbf{b} - \mathbf{z}_0) \cdot (\mathbf{b} - \mathbf{z}_0) \leq (\mathbf{b} - \lambda\mathbf{z} - (1 - \lambda)\mathbf{z}_0) \cdot (\mathbf{b} - \lambda\mathbf{z} - (1 - \lambda)\mathbf{z}_0).$$

This may otherwise be written as a quadratic inequality in λ,

$$0 \leq \lambda^2 ||\mathbf{z}_0 - \mathbf{z}||^2 + 2\lambda(\mathbf{b} - \mathbf{z}_0) \cdot (\mathbf{z}_0 - \mathbf{z})$$

and since $\lambda \in (0, 1)$ we may divide through by λ and then let $\lambda \downarrow 0$ to discover that necessarily $(\mathbf{b} - \mathbf{z}_0) \cdot (\mathbf{z}_0 - \mathbf{z}) \geq 0$. Note we have used the fact that $||\mathbf{z}_0 - \mathbf{z})||^2 \geq 0$. We may now write

$$\begin{aligned} (\mathbf{b} - \mathbf{z}_0) \cdot \mathbf{b} &= (\mathbf{b} - \mathbf{z}_0) \cdot (\mathbf{b} - \mathbf{z}_0) + (\mathbf{b} - \mathbf{z}_0) \cdot \mathbf{z}_0 \\ &= ||\mathbf{b} - \mathbf{z}_0||^2 + (\mathbf{b} - \mathbf{z}_0) \cdot \mathbf{z}_0 \\ &> (\mathbf{b} - \mathbf{z_0}) \cdot \mathbf{z_0} \\ &\geq (\mathbf{b} - \mathbf{z}_0) \cdot \mathbf{z} \end{aligned} \tag{8}$$

for all $\mathbf{z} \in C$. Now take $\mathbf{y} = \mathbf{z}_0 - \mathbf{b}$ and we see that

$$\mathbf{y} \cdot \mathbf{b} < \mathbf{y} \cdot \mathbf{z}$$

for all $\mathbf{z} \in C$ as required. Note that the strict inequality follows from (8). ■

The second result is slighly less intuitively obvious and says that the range of a linear transformation, when restricted in its domain to the positive orthant, is a closed convex domain. Convexity is the easy part, being closed is difficut!

Lemma 6.6 *Suppose as usual that* $\mathbf{A} \in \mathbb{R}^{m \times n}$. *Define*

$$C = \{\mathbf{z} = \mathbf{A}\mathbf{x} | \mathbf{x} \geq \mathbf{0}_n\}$$

Then C is closed and convex.

Proof. To check that C is convex, suppose that $\mathbf{z}_1, \mathbf{z}_2 \in C$ and $\lambda \in (0,1)$. Then $\lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2 \in C$ because

$$\lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2 = \mathbf{A}(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)$$

and $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \geq \mathbf{0}_n$ (the latter is a consequence of the fact that the positive orthant is a convex space).

To show that C is closed, suppose that $\{\mathbf{z}_n : n \geq 1\}$ is a sequence in C which has some limit $\mathbf{z}$. We need to show that $\mathbf{z} \in C$. To do this, note that by definition there exists a sequence of vectors $\{\mathbf{x}_n : n \geq 1\}$ such that $\mathbf{z}_n = \mathbf{A}\mathbf{x}_n$. If it so happens that $\mathbf{x}_n$ has a limit, say $\mathbf{x}$, then since the positive orthant is a closed space, it must follow that $\mathbf{x} \geq \mathbf{0}_n$. Then by continuity of the mapping $\mathbf{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ it follows that $\mathbf{z} = \mathbf{A}\mathbf{x}$.

If the sequence of vectors $\mathbf{x}_n$ does not converge then we can look for a convergent subsequence instead. We do this by assuming without loss of generality that $\mathbf{x}_n$ is a basic feasible solution and hence can be written in the form

$$\mathbf{x}_n = \begin{pmatrix} \mathbf{0} \\ \mathbf{x}_{\mathbf{B}_n} \end{pmatrix}$$

where $\mathbf{x}_{\mathbf{B}_n}$ is the basic part which induces a partition $\mathbf{A} = (\mathbf{A}_n^\circ | \mathbf{B}_n)$ which depends on n. As there can only be a finite number of variables we can include in the basis, along the sequence $\mathbf{x}_n$ we will see the same basis being used infinitely often. For any particular basis which partitions $\mathbf{A} = (\mathbf{A}^\circ | \mathbf{B})$ we can therefore identify a subsequence $\{\mathbf{x}_{\mathbf{B}_{n_k}} : k \geq 1\}$ of the basic feasible solutions $\mathbf{x}_n$ which may be written in terms of that basis. Since $\mathbf{B}$ is inveritble we thus have $\mathbf{x}_{\mathbf{B}_{n_k}} = \mathbf{B}^{-1}\mathbf{z}_{n_k}$ and hence the existence of the limit $\mathbf{z}$ and continuity of the inverse mapping implies the existence of a limit of the $\mathbf{x}_{n_k}$ to some $\mathbf{x}$ satisfying

$$\mathbf{x} = \begin{pmatrix} \mathbf{0} \\ \mathbf{B}^{-1}\mathbf{z} \end{pmatrix}.$$

As before, we necessarily have $\mathbf{x}$ in the positive orthant as it is a closed set and clearly $\mathbf{z} = \mathbf{A}\mathbf{x}$. Hence C contains all its limit points and thus, by definition, is closed. ■

The remaining two lemmas give complementary existence results using the Separating Hyperplane Lemma. The first is used to prove the second and the second is the principle result which is instrumental in proving the Duality Theorem. We assume that the quantities $\mathbf{A}$ and $\mathbf{b}$ are given as in the primal linear programming problem.

Lemma 6.7 (Farkas' Lemma) *Either*

(i) $\mathbf{Ax} = \mathbf{b}$ *and* $\mathbf{x} \geq \mathbf{0}_n$ *has a solution or*

(ii) $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{0}_n^{\mathrm{T}}$ *and* $\mathbf{y} \cdot \mathbf{b} < 0$ *has a solution*

but not both.

Proof. Suppose both (i) and (ii) hold. Then, on the one hand from (i), since $\mathbf{x} \geq \mathbf{0}_n$, we have $\mathbf{y}^{\mathrm{T}}\mathbf{Ax} \geq 0$. On the other hand from (ii) we have $\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \mathbf{y} \cdot \mathbf{b} < 0$. Hence (i) and (ii) cannot be true at the same time. In particular when (i) is true then (ii) is false. To complete the proof it suffices to show that when (i) is false then (ii) is true.

Suppose now that (i) is false and recall the definition

$$C = \{\mathbf{z} = \mathbf{Ax} | \mathbf{x} \geq \mathbf{0}_n\}$$

which was shown to be closed and convex in the previous Lemma. Note that by assumption $\mathbf{b} \notin C$. Then by The Separating Hyperplane Lemma we have the existence of $\mathbf{y}$ such that $\mathbf{y} \cdot \mathbf{b} < \mathbf{y} \cdot \mathbf{z}$ for all $\mathbf{z} \in C$. Taking $\mathbf{x} = \mathbf{0}_n$ we see then that $\mathbf{y} \cdot \mathbf{b} < \mathbf{y} \cdot \mathbf{A0}_n = 0$. The proof is thus concluded by showing that $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{0}_n^{\mathrm{T}}$.

To this end, suppose that $(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i =: \lambda_i < 0$. Then let

$$\mathbf{x} = \frac{1}{\lambda_i}(\mathbf{y} \cdot \mathbf{b})\mathbf{e}_i \geq \mathbf{0}_n$$

where $\mathbf{e}_i$ is the vector in $\mathbf{R}^n$ with zero entries except in the i-th row where it has a unit entry. Now note that

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \mathbf{y} \cdot \mathbf{b}$$

but on account of the definition of $\mathbf{y}$ from the Separating Hyperplane Lemma and the positivity of this particular choice of $\mathbf{x}$, we also have

$$\mathbf{y} \cdot \mathbf{b} < \mathbf{y} \cdot \mathbf{Ax} = \mathbf{y}^{\mathrm{T}}\mathbf{Ax}.$$

This constitutes a contradiction and hence all elements of $\mathbf{y}^{\mathrm{T}}\mathbf{A}$ are positive as required. ■

Lemma 6.8 *Either*

(i) $\mathbf{Ax} \leq \mathbf{b}$ *and* $\mathbf{x} \geq \mathbf{0}_n$ *has a solution, or*

(ii) $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{0}_n^{\mathrm{T}}$, $\mathbf{y} \cdot \mathbf{b} < 0$ *and* $\mathbf{y} \geq \mathbf{0}_m$ *has a solution*

but not both.

Proof. If both (i) and (ii) hold, then

$$0 \leq \mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \mathbf{y} \cdot \mathbf{Ax} \leq \mathbf{y} \cdot \mathbf{b} < 0$$

which is a contradiction. Hence (i) and (ii) cannot hold simultaneously and in particular if (i) is true then (ii) is false. To complete the proof, it again suffices to show that if (i) is false then (ii) is true to complete the proof.

Suppose then that (i) is false. This implies that $\mathbf{Ax} + \mathbf{z} = \mathbf{b}$ has no solution for $\mathbf{x} \geq \mathbf{0}_n, \mathbf{z} \geq \mathbf{0}_m$ as otherwise we would have that $\mathbf{Ax} = \mathbf{b} - \mathbf{z} \leq \mathbf{b}$. Otherwise said, we have that

$$(\mathbf{A}|\mathbf{I}_m)\begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} = \mathbf{b} \text{ and } \begin{pmatrix}\mathbf{x}\\ \mathbf{z}\end{pmatrix} \geq \mathbf{0}_{n+m}$$

has no solution. Hence by Lemma 6.7 it follows that there exists a $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{y}^{\mathrm{T}}(\mathbf{A}|\mathbf{I}_m) \geq \mathbf{0}_{n+m}^{\mathrm{T}}$ and $\mathbf{y} \cdot \mathbf{b} < 0$ as required. Note that the last but one inequality implies that $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{0}_n^{\mathrm{T}}$ and $\mathbf{y}^{\mathrm{T}} \geq \mathbf{0}_m^{\mathrm{T}}$. ■

Now we are ready to prove the Duality Theorem.

Proof of Theorem 6.4. Our strategy for proving this theorem is in two steps. Step 1 is to show that if (P) has no finite optimal feasible solution, then (D) has no finite optimal feasible solution. Since (P) is the dual of (D) it then follows by symmetry that (P) has (no) finite optimal feasible solution if and only if (D) has (no) finite optimal feasible solution. After that, step 2 will show that when both (P) and (D) have finite optimal solutions then their objective values must be equal.

Step 1. There are two ways that (P) may have no finite optimal feasible solution. Either because there is an unbounded optimal solution or because there is no feasible solution. If there is an unbounded optimal solution then by Corollary 6.2 it follows that (D) cannot have a feasible solution. Now suppose that (P) has no feasible solution. Then by Lemma 6.8 there exists a $\mathbf{y}_* \geq \mathbf{0}_m$ such that $\mathbf{y}_*^{\mathrm{T}}\mathbf{A} \geq \mathbf{0}_n^{\mathrm{T}}$ and $\mathbf{y}_* \cdot \mathbf{b} < 0$. Suppose however that there exists a feasible solution to (D), say $\mathbf{y}$, then it must be the case that for any $\lambda \geq 0$ that $\mathbf{y} + \lambda \mathbf{y}_*$ is also a feasible solution. However, as λ is unconstrained, it follows that we can make $(\mathbf{y} + \lambda \mathbf{y}_*) \cdot \mathbf{b}$ arbitrarily large in magnitude showing that (D) has an unbounded feasible solution.

Step 2. From Step 1 it is now apparent that (P) has a bounded optimal solution if and only (D) has a bounded optimal solution. Assume now that both (P) and (D) have bounded optimal solutions. We know already from the Weak Duality Theorem that

$$\max \mathbf{c} \cdot \mathbf{x} \leq \min \mathbf{y} \cdot \mathbf{b}.$$

It would therefore suffice to show that there exist feasible solutions $\mathbf{x}$ and $\mathbf{y}$ to (P) and (D) respectively such that $\mathbf{c} \cdot \mathbf{x} \geq \mathbf{y} \cdot \mathbf{b}$. To this end, let us consider the existence solutions $\mathbf{x} \geq \mathbf{0}_n$, $\mathbf{y} \geq \mathbf{0}_m$ to the linear system

$$\begin{aligned} &\mathbf{A}\mathbf{x} \leq \mathbf{b} \\ &\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{c}^{\mathrm{T}} \\ &\mathbf{c} \cdot \mathbf{x} \geq \mathbf{y} \cdot \mathbf{b} \end{aligned}$$

or, in a more suitable notation for what follows,

$$\begin{pmatrix} \mathbf{A} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{n\times n} & -\mathbf{A}^{\mathrm{T}} \\ -\mathbf{c}^{\mathrm{T}} & \mathbf{b}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \leq \begin{pmatrix} \mathbf{b} \\ -\mathbf{c} \\ 0 \end{pmatrix} \tag{9}$$

Now consider the linear system which requires one to find $\mathbf{z} \geq \mathbf{0}_m$, $\mathbf{w} \geq \mathbf{0}_n$ and $\lambda \geq 0$ such that

$$(\mathbf{z}^{\mathrm{T}} \quad \mathbf{w}^{\mathrm{T}} \quad \lambda) \begin{pmatrix} \mathbf{A} & \mathbf{0}_{m\times m} \\ \mathbf{0}_{n\times n} & -\mathbf{A}^{\mathrm{T}} \\ -\mathbf{c}^{\mathrm{T}} & \mathbf{b}^{\mathrm{T}} \end{pmatrix} \geq \mathbf{0}^{\mathrm{T}}_{n+m} \tag{10}$$

and at the same time

$$\begin{pmatrix} \mathbf{z} \\ \mathbf{w} \\ \lambda \end{pmatrix} \cdot \begin{pmatrix} \mathbf{b} \\ -\mathbf{c} \\ 0 \end{pmatrix} < 0. \tag{11}$$

If we can prove that this system has no solution, then by Lemma 6.8 it follows that (9) must hold with $\mathbf{x} \geq \mathbf{0}_n$, $\mathbf{y} \geq \mathbf{0}_m$. In other words we want to show that

$$\begin{aligned} \mathbf{z}^{\mathbf{T}}\mathbf{A} &\geq \lambda \mathbf{c}^{\mathrm{T}} \\ \lambda \mathbf{b}^{\mathrm{T}} &\geq \mathbf{w}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \\ \mathbf{z} \cdot \mathbf{b} &< \mathbf{w} \cdot \mathbf{c}. \end{aligned} \tag{12}$$

has no solution to complete the proof.

If a solution to (12) were to exist **and** $\lambda = 0$ then let $\mathbf{x}_*$ and $\mathbf{y}_*$ be the optimal solutions to (P) and (D) (which are assumed to exist[20]) and we have

$$\mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{x}_* \geq 0 \text{ and } \mathbf{y}_*^{\mathrm{T}}\mathbf{A}\mathbf{w} \leq 0.$$

But on the other hand, since $\mathbf{A}\mathbf{x}_* \leq \mathbf{b}$ and $\mathbf{y}_*^{\mathrm{T}}\mathbf{A} \geq \mathbf{c}^{\mathrm{T}}$ then

$$\mathbf{c} \cdot \mathbf{w} \leq \mathbf{y}_*^{\mathrm{T}}\mathbf{A}\mathbf{w} \leq 0 \leq \mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{x}_* \leq \mathbf{z} \cdot \mathbf{b}.$$

However, this leads to a contradiction of the third statement of (12).

Suppose instead that a solution to (12) were to exist **and** $\lambda > 0$. In that case we would see that

$$\lambda\mathbf{z} \cdot \mathbf{b} \geq \mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{w} \geq \lambda\mathbf{c} \cdot \mathbf{w}$$

which contradicts again the third statement of (12).

So in short, there can be no solution to (12), in other words no solution to (10) and (11), and hence there must be a solution to (9) and the proof is complete. ■

There is a very nice observation we can also make from the Duality Theorem.

Corollary 6.9 *Suppose that there exists at least one feasible solution to each of (P) and (D) respectively. Then there exist optimal bounded optimal solutions to both (P) and (D) with equal objective values.*

Proof. The result follows as a result of Corollary 6.2 and the Duality Theorem. ■

6.6 Complementary slackness

There is a another very nice relation between the solution of the primal and the dual which can be useful for verifying that a solution to the primal (and hence dual) is optimal.

[20] Remember that we are proving that IF bounded optimal feasible solutions to (P) and (D) exist then their objective values are equal.

Theorem 6.10 (Symmetric complementary Slackness) *Consider the standard linear programming problem:*

$$\begin{array}{l} \textit{maximise } z = \mathbf{c} \cdot \mathbf{x} \\ \textit{subject to:} \\ \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}_n \end{array}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$*,* $\mathbf{b} \in \mathbb{R}^m$*,* $\mathbf{c} \in \mathbb{R}^n$ *are given and* $\mathbf{x} \in \mathbb{R}^n$*.*

Then $\mathbf{x} \in \mathbb{R}^n$ *is an optimal feasible solution for the primal and* $\mathbf{y} \in \mathbb{R}^m$ *is an optimal feasible solution for the dual* **if and only if** *they are feasible solutions and*

$$\begin{array}{l} y_i[(\mathbf{Ax})_i - b_i] = 0 \, \forall i \\ x_i[(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i - c_i] = 0 \, \forall i. \end{array}$$

Note that the last two conditions can also be written equivalently as

$$\begin{array}{l} y_i > 0 \Rightarrow (\mathbf{Ax})_i = b_i \\ x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i. \end{array}$$

or equivalently again as

$$\begin{array}{l} (\mathbf{Ax})_i < b_i \Rightarrow y_i = 0 \\ (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i > c_i \Rightarrow x_i = 0. \end{array}$$

Proof. First we prove the $\Leftarrow$ direction. To this end suppose that $y_i > 0 \Rightarrow (\mathbf{Ax})_i = b_i$ and that $x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i$. In that case

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \sum_i y_i(\mathbf{Ax})_i = \sum_{i:y_i>0} y_i(\mathbf{Ax})_i = \sum_{i:y_i>0} y_i b_i = \sum_i y_i b_i.$$

In other words

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \mathbf{y}^{\mathrm{T}}\mathbf{b} = \mathbf{y} \cdot \mathbf{b}.$$

On the other hand we also have

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \sum_i (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i x_i = \sum_{i:x_i>0} (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i x_i = \sum_i c_i x_i$$

so that we also have

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} = \mathbf{c} \cdot \mathbf{x}$$

Hence $\mathbf{c} \cdot \mathbf{x} = \mathbf{y} \cdot \mathbf{b}$ and thanks to the Weak Duality Theorem this shows that $\mathbf{x}$ and $\mathbf{y}$ are optimal for the primal and the dual respectively.

Now we deal with the $\Rightarrow$ direction. Suppose that $\mathbf{x}$ and $\mathbf{y}$ are optimal for the primal and the dual respectively. Then by the Duality Theorem we have that $\mathbf{c} \cdot \mathbf{x} = \mathbf{y} \cdot \mathbf{b}$. Taking the latter into account together with the fact that $\mathbf{Ax} \leq \mathbf{b}$ and $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{c}^{\mathrm{T}}$ we have

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} \leq \mathbf{y}^{\mathrm{T}}\mathbf{b} = \mathbf{c}^{\mathrm{T}}\mathbf{x} \leq \mathbf{y}^{\mathrm{T}}\mathbf{Ax}. \tag{13}$$

In other words the inequalities in (13) are actually equalities and hence

$$\mathbf{y}^{\mathrm{T}}\mathbf{Ax} - \mathbf{y}^{\mathrm{T}}\mathbf{b} = \sum_i y_i\{(\mathbf{Ax})_i - b_i\} = 0. \tag{14}$$

As $\mathbf{x}$ is feasible it follows that $(\mathbf{Ax})_i - b_i \leq 0$ and as $\mathbf{y}$ is feasible, it also follows that $y_i \geq 0$. In consequence, re-examining (14) we see that

$$y_i\{(\mathbf{Ax})_i - b_i\} = 0$$

for each $i = 1, ..., m$. In other words $y_i > 0 \Rightarrow (\mathbf{Ax}_i) = b_i$ or equivalently $\mathbf{Ax}_i < b_i \Rightarrow y_i = 0$.

To complete the proof note that (13) also implies that

$$\sum_i \{(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i - c_i\}x_i = 0$$

and similar reasoning to above shows that $x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i$ or equivalently $(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i > c_i \Rightarrow x_i = 0$. ■

A similar result can be established for the asymmetric duality.

Theorem 6.11 (Asymmetric complementary slackness) *Consider the canonical linear programming problem:*

$$\begin{aligned}
&\textit{maximise } z = \mathbf{c} \cdot \mathbf{x} \\
&\textit{subject to:} \\
&\mathbf{Ax} = \mathbf{b} \\
&\mathbf{x} \geq \mathbf{0}_n
\end{aligned}$$

where $\mathbf{A} \in \mathbf{R}^{m \times n}$, $\mathbf{b} \in \mathbf{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ *are given and* $\mathbf{x} \in \mathbb{R}^n$.

Then $\mathbf{x} \in \mathbb{R}^n$ *is an optimal feasible solution for the primal and* $\mathbf{y} \in \mathbb{R}^m$ *is an optimal feasible solution for the dual* **if and only if** *they are feasible solutions and*

$$x_i[(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i - c_i] = 0 \,\forall i.$$

Note that the last condition can also be written equivalently as

$$x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i.$$

or equivalently again as

$$(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i > c_i \Rightarrow x_i = 0.$$

Proof. The proof is somewhat simpler than the symmetric case. Suppose that $\mathbf{x}$ and $\mathbf{y}$ are optimal feasible solutions for the primal and dual respectively. Then by the Duality Theorem we have $\mathbf{c} \cdot \mathbf{x} = \mathbf{b} \cdot \mathbf{y}$. It follows that

$$\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{y}^{\mathrm{T}}\mathbf{b} = \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

and hence

$$(\mathbf{y}^{\mathrm{T}}\mathbf{A} - \mathbf{c}^{\mathrm{T}})\mathbf{x} = \sum_i \{(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i - c_i\} x_i = 0. \tag{15}$$

As $\mathbf{x} \geq \mathbf{0}_n$ and $\mathbf{y}^{\mathrm{T}}\mathbf{A} \geq \mathbf{c}^{\mathrm{T}}$ it follows that $x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i$.

Now suppose that $x_i > 0 \Rightarrow (\mathbf{y}^{\mathrm{T}}\mathbf{A})_i = c_i$ (or equivalently that $(\mathbf{y}^{\mathrm{T}}\mathbf{A})_i > c_i \Rightarrow x_i = 0$). In that case it follows that (15) holds. That is to say $\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{c}^{\mathrm{T}}\mathbf{x}$. On the other hand, as $\mathbf{A}\mathbf{x} = \mathbf{b}$ it follows that $\mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{y}^{\mathrm{T}}\mathbf{b}$. In conclusion $\mathbf{c} \cdot \mathbf{x} = \mathbf{b} \cdot \mathbf{y}$ so that by the Weak Duality Theorem $\mathbf{x}$ and $\mathbf{y}$ are optimal. ∎

Note that at the end of the above proof we have used the Weak Duality Theorem for the asymmetric dual. Strictly speaking the Weak Duality Theorem, as presented in this text, is a statement about the symmetric case. None the less it is still true for the asymmetric case. Can you prove it? (See Exercise sheets).

Here is an example of how to use complementary slackness. Consider the linear programming problem

minimise $z = x_1 + 6x_2 + 2x_3 - x_4 + x_5 - 3x_6$
subject to:
$x_1 + 2x_2 + x_3 + 5x_6 = 3$
$-3x_2 + 2x_3 + x_4 + x_6 = 1$
$5x_2 + 3x_3 + x_5 - 2x_6 = 2$
$x_1, ..., x_6 \geq 0.$

Writing $\mathbf{c}^{\mathrm{T}} = (1, 6, 2, -1, 1, -3)$ we can multiply the objective by -1 to convert the primal to a maximisation problem and then write down its (asymmetric) dual,

$$\begin{aligned}&\text{minimise } z = 3z_1 + z_2 + 2z_3\\&\text{subject to:}\\&\mathbf{z}^{\mathrm{T}}\mathbf{A} \geq -\mathbf{c}^{\mathrm{T}}\end{aligned}$$

Note in particular we have $\mathbf{z} \in \mathbb{R}$. It is more convenient to multiply the constraints by -1 and since there is no restriction on the sign of the entries in $\mathbf{z}$ we can set $-\mathbf{y} = \mathbf{z}$. This leads us to the representation of the dual problem,

$$\begin{aligned}&\text{maximise } z = 3y_1 + y_2 + 2y_3\\&\text{subject to:}\\&\mathbf{y}^{\mathrm{T}}\mathbf{A} \leq \mathbf{c}^{\mathrm{T}}\end{aligned}$$

Suppose we are asked to verify that

$$\mathbf{x}^{\mathrm{T}} = (0, \frac{16}{29}, 0, \frac{66}{29}, 0, \frac{11}{29})$$

is optimal for the primal. If this were the case then the Asymmetric Complementary Slackness Theorem would hold. In particular we would have that as x_2, x_4, x_6 are all strictly positive, then $(\mathbf{y}^{\mathrm{T}}\mathbf{A})_2 = c_2, (\mathbf{y}^{\mathrm{T}}\mathbf{A})_4 = c_4$ and $(\mathbf{y}^{\mathrm{T}}\mathbf{A})_6 = c_6$. Having written out carefully the matrix $\mathbf{A}$ we see that this is tantamount to

$$\begin{aligned}&2y_1 - 3y_2 + 5y_3 = 6\\&y_2 = -1\\&5y_1 + y_2 - 2y_3 = -3.\end{aligned}$$

Solving these equations we find that

$$(y_1, y_2, y_3) = (-\frac{4}{29}, -1, \frac{19}{29})$$

and it is a straightforward exercise to check that $\mathbf{y}$ is feasible for the dual and that $\mathbf{c} \cdot \mathbf{x} = \mathbf{b} \cdot \mathbf{y} = -3/29$.

In conclusion, by assuming the given $\mathbf{x}$ is optimal, we have been able to use complementary slackness to deduce what the optimal $\mathbf{y}$ should be. However, having obtained this hypothetical value of $\mathbf{y}$ we discover that in fact $\mathbf{x}$ and $\mathbf{y}$ are necessarily optimal solutions to primal and dual respectively by by virtue of the fact that they have the same objective values and hence the Weak Duality Theorem comes into effect.

7 The transportation problem

7.1 Introduction

We are concerned with moving a commodity from various sources (factories) to various destinations (markets) at minimum cost. The standard problem has m sources of the commodity with respective (strictly positive) supplies $s_i, i = 1, ..., m$ and n destinations for the commodity having respective (strictly positive) demands $d_j, j = 1, ..., n$. The transport cost per unit between supply i and demand j is c_{ij}. We wish to plan the transport from the supply to the demand at minimal total cost. The problem may be formulated as a linear programming problem with special structure as follows.

Let x_{ij} be the amount transported from source i to destination j. The problem is then:

minimise $z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
subject to:
$\sum_{j=1}^{n} x_{ij} \leq s_i \quad i = 1, ..., m$
$\sum_{i=1}^{m} x_{ij} \geq d_j \quad j = 1, ..., n$
$x_{ij} \geq 0 \quad \forall i = 1, ..., m$ and $j = 1, ..., n.$

Remarks

1. Note that the general transportation problem is a linear programming problem with mn variables and $m + n$ constraints.

2. In order for a feasible solution to exist we must necessarily have that total supply is greater or equal to demand. This is indeed the case since

$$\sum_{i=1}^{m} s_i \geq \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \geq \sum_{j=1}^{n} d_j. \tag{16}$$

7.2 The canonical form

If it is the case that supply strictly outstrips demand, then we can introduce an additional 'dump' to the problem where additional supply, $\sum_i^m s_i - \sum_j^n d_j$ is consumed and cost of shipping to this 'dump' state as equal to zero. Note that the zero shipping cost means that objective function for the enlarged problem remains the same as the original one. In that case **we may assume**

without loss of generality that supply is equal to demand. In other words the transportation problem is **balanced** and

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j.$$

In that case, the inequalities in (16) are in fact equalities and the transportation problem takes the **canonical** or **balanced** form as follows:

minimise $z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
subject to:
$\sum_{j=1}^{n} x_{ij} = s_i \quad i = 1, ..., m$
$\sum_{i=1}^{m} x_{ij} = d_j \quad j = 1, ..., n$
$x_{ij} \geq 0 \quad \forall i = 1, ..., m$ and $j = 1, ..., n$

where $c_{ij} \geq 0$, $s_i > 0$, $d_j > 0$ are given with $\sum_i s_i = \sum_j d_j$.

7.3 Dual of the transportation problem

It turns out that providing an effective algorithmic solution to the transportation problem involves making calculations which interplay between the primal and dual. We therefore devote a little time to understanding the relationship of the transportation problem to its dual.

Suppose we introduce the matrix

$$\mathbf{X} = (x_{ij}) = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_m^{\mathrm{T}} \end{pmatrix}$$

where for $i = 1, ..., m$

$$\mathbf{x}_i^{\mathrm{T}} = (x_{i1}, ..., x_{in}).$$

Then let $\mathbf{x}^{\mathrm{T}} = (\mathbf{x}_1^{\mathrm{T}}, ..., \mathbf{x}_m^{\mathrm{T}})$, in other words

$$\mathbf{x}^{\mathrm{T}} = (x_{11}, x_{12}, ..., x_{1n}, x_{21}, ..., x_{2n}, ..., x_{mn}).$$

Similarly write

$$\mathbf{C} = (c_{ij}) = \begin{pmatrix} \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_m^{\mathrm{T}} \end{pmatrix}$$

where for $i = 1, ..., m$

$$\mathbf{c}_i^{\mathrm{T}} = (c_{i1}, ..., c_{in}).$$

Then let $\mathbf{c}^{\mathrm{T}} = (\mathbf{c}_1^{\mathrm{T}}, ..., \mathbf{c}_m^{\mathrm{T}})$, in other words

$$\mathbf{c}^{\mathrm{T}} = (c_{11}, c_{12}, ..., c_{1n}, c_{21}, ..., c_{2n}, ..., c_{mn}).$$

Now let

$$\mathbf{b}^{\mathrm{T}} = (s_1, ..., s_m, d_1, ..., d_n)^{\mathrm{T}}.$$

The primal problem can be written in the form

$$\begin{aligned}
&\text{minimise } z = \mathbf{c} \cdot \mathbf{x}\\
&\text{subject to:}\\
&\mathbf{A}\mathbf{x} = \mathbf{b}\\
&\mathbf{x} \geq \mathbf{0}_{mn}
\end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{(m+n)\times mn}$ and for $1 \leq i \leq m, 1 \leq j \leq mn$ we have[21]

$$A_{ij} = \delta_{j,\{(i-1)n+1,....,in\}}$$

and for $m+1 \leq i \leq m+n, 1 \leq j \leq mn$

$$A_{ij} = \delta_{j,\{(i-m),(i-m)+n,...,(i-m)+(m-1)n\}}.$$

Otherwise said, the main constraints can be seen in the following form

$$\begin{pmatrix}
1 & 1\cdots1 & 1 & & & & & & & \\
 & & & 1 & 1\cdots1 & 1 & & & & \\
 & & & & & & \ddots & & & \\
 & & & & & & & 1 & 1\cdots1 & 1 \\
1 & & & 1 & & & & 1 & & \\
 & 1 & & & 1 & & & & 1 & \\
 & \ddots & & & \ddots & & & & \ddots & \\
 & & 1 & & & 1 & & & & 1
\end{pmatrix}
\begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \\ x_{21} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{m1} \\ \vdots \\ x_{mn} \end{pmatrix}
=
\begin{pmatrix} s_1 \\ \vdots \\ s_m \\ d_1 \\ \vdots \\ d_n \end{pmatrix}. \quad (17)$$

It is important to note that the matrix $\mathbf{A}$ has only entries which are either zero or unity. Further, there are only two non-zero entries in each column.

[21] For scalar j and set S, we use the delta function $\delta_{j,S}$ which is equal to 1 if $j \in S$ and zero otherwise.

This means that the matrix $\mathbf{A}^{\mathrm{T}}$ has only two non-zero entries in each row. Another way of representing $\mathbf{A}$ is to characterise its columns. We know that $\mathbf{A} \in \mathbb{R}^{(m+n)\times mn}$ and hence write

$$\mathbf{A} = (\mathbf{c}_{11}|\mathbf{c}_{12}|\cdots|\mathbf{c}_{1n}|\mathbf{c}_{21}|\cdots|\mathbf{c}_{mn})$$

where

$$\mathbf{c}_{ij} = \begin{pmatrix} \mathbf{e}_i^{(m)} \\ \mathbf{e}_j^{(n)} \end{pmatrix}$$

where $\mathbf{e}_l^{(k)}$ is the vector in $\mathbb{R}^k$ whose entries all zero with the exception of the l-th entry which is unity.

Consider the asymmetric dual[22] which requires one to

$$\begin{gathered} \text{maximise } z = \mathbf{b}\cdot\mathbf{y} \\ \text{subject to:} \\ \mathbf{A}^{\mathrm{T}}\mathbf{y} \leq \mathbf{c} \\ \mathbf{y} \in \mathbb{R}^{m+n}. \end{gathered}$$

For convenience let us write

$$\mathbf{y} = (u_1, ..., u_m, v_1, ..., v_n)^{\mathrm{T}}.$$

Then note that the condition $\mathbf{A}^{\mathrm{T}}\mathbf{y} \leq \mathbf{c}$ can be written as the system of inequalities

$$u_i + v_j \leq c_{ij} \text{ for all } i = 1, ..., m \text{ and } j = 1, ..., n.$$

Further, the objective function $\mathbf{b}\cdot\mathbf{y}$ can be written as

$$\sum_{j=1}^{n} v_j d_j + \sum_{i=1}^{m} u_i s_i.$$

To conclude, the dual problem to the balanced transportation problem can be summarised as the following linear programming problem

$$\begin{aligned} &\text{maximise } z = \textstyle\sum_{j=1}^{n} v_j d_j + \sum_{i=1}^{m} u_i s_i. \\ &\text{subject to:} \\ &u_i + v_j \leq c_{ij} \text{ for all } i = 1, ..., m \text{ and } j = 1, ..., n. \end{aligned}$$

[22] In order to write down the asymmetric dual, it is useful to write the primal in the form: maximise $z = (-\mathbf{c})\cdot\mathbf{x}$ subject to $-\mathbf{A}\mathbf{x} = -\mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}_{mn}$.

with no restriction on the sign of the u_is and v_js.

There is an intuitive way of looking at the formulation of the dual problem. The primal concerns a way of shipping goods from supply to demand at minimal cost. Suppose now there is a competing agent who wishes to offer the same service. This agent charges u_i for picking up goods at the i-th supply and charges v_j for delivering to the j-th demand[23]. The charge for shipping from $i \to j$ is thus $u_i + v_j$. In order to remain competitive, this agent wishes to maximise his income, i.e. $\sum_{j=1}^{n} v_j d_j + \sum_{i=1}^{m} u_i s_i$, but at the same time offering a cheaper alternative to the operator in the primal problem, i.e. $u_i + v_i \leq c_{ij}$ on all routes $i \to j$.

7.4 Properties of the solution to the transportation problem

Let us first examine existence of the solution to the transportation problem. Here The Duality Theorem will play an essential role.

Theorem 7.1 *The transportation problem when in balance has a finite optimal solution.*

Proof. Consider the solution to the primal

$$x_{ij} = \frac{s_i d_j}{\sum_{i=1}^{m} s_i}.$$

It is easy to check that this solution satisfies $\sum_i x_{ij} = d_j$, $\sum_j x_{ij} = s_i$, $x_{ij} \geq 0$. In other words it is a feasible solution.

On the other hand, if we look at the dual to the transportation problem, then it is also clear that $v_j = u_i = 0$ for all $i = 1, ..., m$ and $j = 1, ..., n$ is a feasible solution.

Hence by Corollary 6.9 a bounded optimal solution to both primal and dual must exist. ■

The following result is requires a proof which goes beyond the scope of this course and hence the proof is omitted.

---

[23]There is a slight problem with this interpretation in that the charges u_i and v_j may be negative valued! Can you think why this is not necessarily a problem?

Theorem 7.2 *If the supplies $s_i : i = 1, ..., m$ and the demands $d_j : j = 1, ..., n$ are all integers, then any feasible solution to the transportation problem $\mathbf{X} = (x_{ij})$ in balance is an integer valued matrix. That is to say $\mathbf{X} \in \mathbb{Z}_+^{(m+n)\times mn}$.*

Intuitively speaking, the reason why this result is true is due to the fact that the entries of the matrix $\mathbf{A}$ are either zero or unity.

Finally we conclude this section with a result concerning the rank of the matrix $\mathbf{A}$ which will be of use in the next section when we look at an algorithm for solving the transportation problem.

Theorem 7.3 *Rank*$(\mathbf{A}) = m + n - 1$.

Proof. Recall that $\mathbf{A} \in \mathbb{R}^{(n+m)\times mn}$. Suppose that $\mathbf{r}_1^{\mathrm{T}}, ..., \mathbf{r}_{m+n}^{\mathrm{T}}$ are the rows of $\mathbf{A}$. Partitioning $\mathbf{A}$ between the m-th and $(m+1)$-th row, in any one column we have just one unit entry on either side and the remaining terms zero. Hence

$$\sum_{i=1}^{m} \mathbf{r}_i = \sum_{i=m+1}^{n} \mathbf{r}_i = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{mn}.$$

It follows that the rows of $\mathbf{A}$ are linearly dependent. The main aim of this proof is to deduce from this fact that this means that rank of the matrix $\mathbf{A}$ is the number of rows minus one.

To this end, consider solutions $(\lambda_1, \cdots, \lambda_{m+n-1})$ to the equation

$$\mathbf{u}(\lambda_1, ..., \lambda_{m+n-1}) = 0$$

where

$$\mathbf{u}(\lambda_1, ..., \lambda_{m+n-1}) := \sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i.$$

As we do not include the bottom row of the matrix $\mathbf{A}$ it must be the case that

$$\begin{array}{c} \left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_n = \lambda_1 = 0 \\ \left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_{2n} = \lambda_2 = 0 \\ \vdots \\ \left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_{mn} = \lambda_m = 0. \end{array}$$

Note, it is worth looking at (17) to see why the above holds.

As a consequence of the above we also have

$$\begin{aligned}
\left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_1 &= \lambda_1 + \lambda_{m+1} = \lambda_{m+1} = 0 \\
\left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_2 &= \lambda_1 + \lambda_{m+2} = \lambda_{m+2} = 0 \\
&\vdots \\
\left(\sum_{i=1}^{m+n-1} \lambda_i \mathbf{r}_i\right)_{n-1} &= \lambda_1 + \lambda_{m+n-1} = \lambda_{m+n-1} = 0.
\end{aligned}$$

In conclusion, the unique solution to the equation $\mathbf{u}(\lambda_1, ..., \lambda_{m+n-1}) = 0$ is $\lambda_1 = \cdots \lambda_{m+n-1} = 0$ showing that all but the last row of $\mathbf{A}$ are linearly independent and hence $\text{rank}(\mathbf{A}) = m + n - 1$. ■

To some extent the conclusion of the last theorem is intuitively obvious since we have m equations of the form $\sum_j x_{ij} = s_i$, n equations of the form $\sum_i x_{ij} = d_j$ with one constraint $\sum_i s_i = \sum_j d_j$.

7.5 Solving the transportation problem

The transportation problem is nothing more than a linear programming problem and hence there is no reason why not to solve it using the simplex algorithm. However, by virtue of the large number of variables involved this not the most practical option. Below we describe an algorithm for solving the transportation problem which is essentially based on the Asymmetric Complementary Slackness Theorem. Looking back to this theorem and putting it into the current context we have that, if x_{ij} and u_i, v_j are feasible, then they satisfy

$$x_{ij} > 0 \Rightarrow u_i + v_j = c_{ij}$$

and

$$u_i + v_j < c_{ij} \Rightarrow x_{ij} = 0$$

where $i = 1, ..., m$ and $j = 1, ..., n$.

Here-now is the promised algorithm.

(a) Find a basic feasible solution to the primal. Since $\text{rank}(\mathbf{A}) = m + n - 1$ it follows that that there should be at most $m+n-1$ of the entries x_{ij} are non-zero.[24] Assume that this solution is optimal.

[24] When there are less than $m + n - 1$ non-zero entries, we have a degenerate basic feasible solution.

(b) Write down the $m+n-1$ equations which are of the form

$$u_i + v_j = c_{ij} \text{ for each } i,j \text{ that } x_{ij} > 0.$$

This will give the solution to the $m+n$ variables $u_1, ..., u_m, v_1, ..., v_n$ in terms of an unknown parameter. We can fix this parameter by choosing (for example) $u_1 = 0$.

(c) If with this set values for $u_i, v_j,\ i = 1, ..., m,\ j = 1, ..., n$ we also have that

$$u_i + v_j \leq c_{ij} \text{ for each } i,j \text{ that } x_{ij} = 0$$

then by the the Asymmetric Complementary Slackness Theorem it follows that the feasible solution (x_{ij}) must be optimal as assumed.

If however for some pair i,j we have that

$$u_i + v_j > c_{ij} \text{ and } x_{ij} = 0$$

then we have violated the conditions of the Asymmetric Complementary Slackness Theorem (specifically the dual solution is not feasible) and the assumption that our feasible solution is optimal must be false. We can improve things by increasing the value of this particular x_{ij} (where i,j are the offending pair of indicies) to a non-zero value, adjusting the values of the other x_{ij}-s so that there are still $m+n-1$ non-zero entries[25] and returning to step (a).

This algorithm is not complete as at least the following two questions remain to be answered when utilising the suggested routine.

How does one choose an initial basic feasible solution?

How does one re-adjust the feasible solution in step (c)?

We address these two points in the next two sections.

[25] This is akin to moving to another basic feasible solution in the Simplex Method.

7.6 Initial feasible solutions

There are several methods, but here are two of the more popular.

The 'North-West corner' method: Start by putting $x_{11} = \min\{s_1, d_1\}$ and then move along the rows of x_{ij} using up either the supplies or filling the demands until all supplied are distributed and all demands are fulfilled. Below is an example.

	7	11	15	$\leftarrow d_j$
14	7	7	0	
13	0	4	9	
6	0	0	6	
$s_i \uparrow$				

The 'matrix' (or 'cheapest route') method: We find $\min_{i,j} c_{ij}$ and then allocate as large an x_{ij} value as possible to route $i \to j$ (in other words satisfying either supply or demand, whichever is the smaller). We repeat with the next smallest[26] c_{ij} and on until all factories are empty and all markets are full. This method is lengthier but gives a better starting point. Consider the previous example but with the transport costs given by the matrix

$$(c_{ij}) = \begin{pmatrix} 14 & 13 & 6 \\ 15 & 14 & 8 \\ 9 & 11 & 2 \end{pmatrix}.$$

In this case we establish an initial feasible solution as

	7	11	15	
14	0	5	9	
13	7	6	0	
6	0	0	6	

Note if the problem has been set up with an additional 'dump' then obviously it would not be advisable to allocate to the dump.

[26] If there is a tie between two values of c_{ij} then one picks one of them arbitrarily.

7.7 A worked example

We are given supply/demand/cost data in the following format

	55	70	35	40	$\leftarrow d_j$
80	13	11	18	17	
100	2	14	10	1	
20	5	8	18	11	
$s_i \uparrow$					

Step 1: We start by constructing a feasible solution using the North-West corner method and obtain the following matrix for (x_{ij}) with the cost matrix (c_{ij}) embedded in.

	55	70	35	40	$u_i \downarrow$
80	55 [13]	25 [11]	0 [18]	0 [17]	-3
100	0 [2]	45 [14]	35 [10]	20 [1]	0
20	0 [5]	0 [8]	0 [18]	20 [11]	10
$v_j \rightarrow$	16	14	10	1	$z = 2210$

On the right hand column and bottom row, we have also included the values of $(u_1, u_2, u_3, v_1, v_2.v_3)$ by solving the complementary slackness equations. Specifically

$$\begin{aligned}
x_{11} > 0 &\Rightarrow u_1 + v_1 = 13\\
x_{12} > 0 &\Rightarrow u_1 + v_2 = 11\\
x_{22} > 0 &\Rightarrow u_2 + v_2 = 14\\
x_{23} > 0 &\Rightarrow u_2 + v_3 = 10\\
x_{24} > 0 &\Rightarrow u_2 + v_4 = 1\\
x_{34} > 0 &\Rightarrow u_3 + v_4 = 11.
\end{aligned}$$

We have solved this system by choosing arbitrarily $u_2 = 0$.[27] By inspection we can see that this solution is not optimal because

$$x_{31} = 0 \text{ and yet } u_3 + v_1 = 26 > 5 = c_{31}.$$

in fact the entry in x_{31} is the 'worst offender' in this class so to speak in the sense that

$$\min_{ij:x_{ij}=0} \{c_{ij} - u_i - v_j\} = c_{31} - u_3 - v_1 = -21$$

[27]Note the algorithm we have given suggests to take $u_1 = 0$, however we have chosen $u_2 = 0$ to show that it does not matter.

(which is necessarily a negative number when the solution is not optimal).

Step 2: We can improve our solution by introducing a non-zero entry to x_{31} by so that there are the same number of non-zero entries. Intuitively speaking, our initial solution was a basic solution and we now wish to change the basis to introduce x_{31}.

We introduce x_{31} with the following perturbation loop.

	·	·	·	·	
·	$55-\eta$	$25+\eta$	·	·	·
·	·	$45-\eta$	·	$20+\eta$	·
·	$0+\eta$	·	·	$20-\eta$	·
	·	·	·	·	·

In this perturbation loop we add and subtract η from a loop of nodes in such a way that in any row or any column, for every time η has been added on to an entry, it has also be subtracted off from another entry. This has the effect that the perturbed solution is still feasible. Note that the maximum value of η we can work with is 20 in order to keep the maximum number of non-zero entries equal to $m+n-1=6$. Reallocating resources with $\eta=20$ and computing the u_i-s and v_j-s again using complementary slackness we have

	55	70	35	40	
80	35 [13]	45 [11]	0 [18]	0 [17]	-3
100	0 [2]	25 [14]	35 [10]	40 [1]	0
20	20 [5]	0 [8]	0 [18]	0 [11]	-11
	16	14	10	1	$z=1790$

where we can see that again we have arbitrarily chosen $u_2=0$.

Step 3: Note that

$$\min_{ij:x_{ij}=0}\{c_{ij}-u_i-v_j\}=c_{21}-u_2-v_1=-14$$

and so we propose to introduce x_{21} with the following perturbation loop

	·	·	·	·	
·	$35-\eta$	$45+\eta$	·	·	·
·	$0+\eta$	$25-\eta$	·	·	·
·	·	·	·	·	·
	·	·	·	·	·

The maximum value of η permitted is 25 and hence we obtain the new table

	55	70	35	40	
80	10 [13]	70 [11]	0 [18]	0 [17]	11
100	25 [2]	0 [14]	35 [10]	40 [1]	0
20	20 [5]	0 [8]	0 [18]	0 [11]	3
	2	0	10	1	$z = 1440$

where the new values of u_i and v_j have been filled in with the arbitrary choice $u_2 = 0$.

Step 4: We compute as usual

$$\min_{ij:x_{ij}=0} \{c_{ij} - u_i - v_j\} = c_{13} - u_1 - v_3 = -3.$$

and work with the following perturbation loop

	·	·	·	·	
·	$10-\eta$	·	$0+\eta$	·	·
·	$25+\eta$	·	$35+\eta$	·	·
·	·	·	·	·	·
	·	·	·	·	·

which permits $\eta = 10$ as its largest value. This results in the following table

	55	70	35	40	
80	0 [13]	70 [11]	10 [18]	0 [17]	8
100	35 [2]	0 [14]	25 [10]	40 [1]	0
20	20 [5]	0 [8]	0 [18]	0 [11]	3
	2	3	10	1	$z = 1410$

By construction we have that $x_{ij} > 0 \Rightarrow u_i + v_j = c_{ij}$ and a quick inspection shows that for the remaining entries satisfy $u_i + v_j \leq c_{ij}$ and $x_{ij} = 0$. Hence the conditions of the Asymmetric Complementary Slackness Theorem are fulfilled and this solution must be optimal.

7.8 Improvement at each iteration

Why did we work with the 'worst offender' $\min_{ij:x_{ij}=0}\{c_{ij} - u_i - v_j\}$ at each stage of the iteration in the example above?

If x_{kl} is the variable to be made basic, then we find a 'pertubation loop' of **basic** variables of the form

$$x_{kl} \leftrightarrow x_{kp} \updownarrow x_{rp} \leftrightarrow x_{r\cdot}, \cdots, x_{s\cdot} \leftrightarrow x_{sl} \updownarrow x_{kl}$$

These are alternately decreased and increased by the value η. The difference between the new value of the objective minus old value of the objective is therefore

$$\begin{aligned}
&\eta(c_{kl} - c_{kp} + c_{rp} - c_{r\cdot} + \cdots + c_{s\cdot} - c_{sl}) \\
&= \eta[c_{kl} - (u_k + v_p) + (u_r + v_p) - (u_r + v_\cdot) + \cdots + (u_s + v_\cdot) - (u_s + v_l)] \\
&= \eta(c_{kl} - u_k - v_l) \\
&< 0
\end{aligned}$$

whenever $c_{kl} - u_k - v_l < 0$. Note that the first equality follows becasue, apart from the pair k, l all other entries x_{ij} in the pertubation loop are positive and hence for such pairs i, j we have $c_{i,j} = u_i + v_j$. The second equality is the consequence of a large cancellation of terms. For a non-degenerate solution we will always have $\eta > 0$ and so there will be a definite reduction in the total transport cost. Choosing the 'worst offender', ie chooising x_{kl} such that

$$c_{kl} - u_k - v_l = \min_{ij:x_{ij}=0} \{c_{ij} - u_i - v_j\} < 0,$$

is a good way of trying to maximise the reduction at each step[28]. Of course, if one does not choose the 'worst offender' to include in the basis, it is not detrimental to the ultimate aim of minimising the objective, so long as one chooses an x_{kl} for which $c_{kl} - u_k - v_l < 0$.

7.9 Degeneracy

Recall that a basic feasible solution to a linear programming problem is degenerate if one of the variables in the basis is zero-valued. In the context of a transportation problem this would correspond to finding a solution x_{ij} for which there are strictly less than $m + n - 1$ non-zero entries. (Recall that $\text{rank}(A) = m + n - 1$).

[28]If one considers two different perturbation loops, then there is not guarantee that the quantity η is the same for each one and hence strictly speaking we cannot compare the relative reduction between loops at this level of abstraction even if the quantity $c_{kl} - u_k - v_l$ is the smallest negative value possible.

It is not uncommon to obtain degenerate solutions to transportation problems (this can even happen by using, for example, the matrix method to construct an initial solution). One needs to be a little careful in the case that one has a degenerate solution. Consider for example the following example (taken from the 2004 exam).

In a land levelling project, soil must be taken from four zones and taken to five other zones. The amounts (in thousands of tons) to be removed from zones A,B,C and D are 25, 50, 30 and 40 respectively whilst the amounts needed at zones K, L, M, N and P are 30, 60, 20, 15 and 20. The average distances travelled (in meters) in moving soil between the various zones are shown below.

	K	L	M	N	P
A	120	70	200	60	100
B	80	130	100	130	140
C	190	130	70	110	160
D	130	90	100	180	150

Solve the problem of moving the soil as required whilst minimising the effort required in terms of total tons times meters travelled.

This problem is a standard balanced transportation problem where the table above can be taken directly as the cost matrix. Applying the matrix method we can immediately produce the following initial solution.

	30	60	20	15	20	
25	0 [120]	10 [70]	0 [200]	15 [60]	0 [100]	·
50	30 [80]	**0** [130]	0 [100]	0 [130]	20 [140]	·
30	0 [190]	10 [130]	20 [70]	0 [110]	0 [160]	·
40	0 [130]	40 [90]	0 [100]	0 [180]	0 [150]	·
	·	·	·	·	·	·

Note that $m = 4, n = 5$ and hence $m + n - 1 = 8$ and yet the solution above only has 7 non-zero entries. It is degenerate. This poses a problem if we try to compute the dual values u_i, v_j as, even choosing $u_1 = 0$, there are not enough equations to obtain exact dual values. One way around this is to flag one of the zeros, say $0 = x_{22}$ as the degenerate part of the basic solution (we have done this by putting it in bold) and then including the equality $u_2 + v_2 = c_{22}$ in the set of linear equations used to solve for the dual variables. We can now proceed to solve for the dual values (choosing $u_1 = 0$) and obtain.

	30	60	20	15	20	
25	0 [120]	$10+\eta$ [70]	0 [200]	$15-\eta$ [60]	0 [100]	0
50	30 [80]	**0** [130]	0 [100]	0 [130]	20 [140]	60
30	0 [190]	$10-\eta$ [130]	20 [70]	$0+\eta$ [110]	0 [160]	60
40	0 [130]	40 [90]	0 [100]	0 [180]	0 [150]	20
	20	70	10	60	80	·

On a practical note, there is a quick way of filling out the final column and row. Start by setting $u_1 = 0$, move across the first row to non-zero entries, do a quick mental calculation and fill in the entries for those columns in the final row. Next by looking at the second row use these entries in the final row to obtain as many entries as possible in the final column and so on until all values of u_i, v_j have been filled out. Having filled out the final column and final row one may now quickly check for optimality by looking at all the zero entries and seeing if the corresponding values in the terminal row and column add up to something less than or equal to the corresponding cost in square brackets.

We have also included in the table above a perturbation loop around the 'worst offender' x_{34}.

Following through with the calculations (bearing in mind that we are still thinking of $0 = x_{22}$ as part of the basic solution), the next table takes the form

	30	60	20	15	20	
25	0 [120]	20 [70]	0 [200]	5 [60]	0 [100]	0
50	30 [80]	**0** [130]	0 [100]	0 [130]	20 [140]	60
30	0 [190]	0 [130]	20 [70]	10 [110]	0 [160]	50
40	0 [130]	40 [90]	0 [100]	0 [180]	0 [150]	20
	20	70	20	60	80	$z = 13,000$

and the reader can easily check that we now have that $u_i + v_j \leq c_{ij}$ for all i, j such that $x_{ij} = 0$. Hence we have found the optimal solution.

7.10 Pricing out

Suppose that we wish to prevent that the solution to the transportation problem makes use of the route $i \to j$. We can do this by setting $c_{ij} = M$ where

M is some arbitrarily large value (or indeed $M = \infty$). The transportation algorithm will naturally move away from solutions which use this route as it is a minimisation algorithm. We have thus 'priced-out' the route $i \to j$.

Consider the example in the previous section. Suppose in addition, we want to solve this transportation problem with the additional constraint that at most 20,000 tons are taken from D to L. The way to handle this additional constraint is to spilt the zone D into two parts D_1, which corresponds to the first 20,000 tons of soil at D, and D_2, which corresponds to the remaining 20,000 tons of soil at D. We thus formulate the problem with the data in the table below.

	30	60	20	15	20	
25	120	70	200	60	100	
50	80	130	100	130	140	
30	190	130	70	110	160	
D_1 : 20	130	90	100	180	150	
D_2 : 20	130	M	100	180	150	

This table shows in the last two rows that shipping any soil over and above 20,000 tons from D has identical costs to shipping any of the first 20,000 tons except to L which incurs an arbitrary large cost (and hence implicitly will be avoided by the algorithm). One may now proceed to solve this transportation problem in the usual way (note that the matrix method will produce a degenerate initial solution again).[29] At any stage of the algorithm one should avoid introducing the variable x_{52} and when computing the values of the dual variables u_i, v_j note that it will always be the case that (when checking the zero $0 = x_{52}$) $u_5 + v_2 \leq M$ since M is assumed to be arbitrarily large.

Below is another example of how to use pricing out (taken from the 2005 exam). Unlike the above example we use pricing out to build in a temporal feature of the given problem.

A factory makes a product for which there is a fluctuation but predictable demand. The following table shows the predicted demand for each of the next five months together with the production capacity and unit production costs for each month.

[29] Please refer to the model solutions provided on my webpage.

Month	1	2	3	4	5
Demand (‘000)	60	80	85	100	70
Production Capacity (‘000)	90	95	100	100	90
Unit production cost (pence)	50	55	60	60	65

Items produced in a month are available to meet demand in the same month but they may also be stored to meet demand 1,2,3 or 4 months later at a cost of 4,7,9 and 10 pence per item respectively. It is required to schedule production to meet the anticipated demand at minimum total cost.

Formulating this as a transportation problem is rather tricky as there is a temporal aspect that we need to build into the solution. The idea is to note that the supply at month n can in principle feed the demand at month n as well as all subsequent months. However the supply at month n **cannot** feed the demands at previous months.[30] We may thus think of x_{ij} for $i = 1, ..., 5$ and $j = i, .., 5$ as the amount produced at time i which feeds the demand at time j. Further when computing c_{ij} we need to take account of the production cost at time i plus the storage costs for the additional months after the production month before the item is sold on. We think of the supply nodes as the production in the individual months (the second row of the above table) and the markets as the demand in each month (the first row of the above table). In that case we see that our transportation problem is not balanced and we need to introduce a ‘dump’. Hence we present the given data for use in a transportation problem as follows.

	60	80	85	100	70	80	
90	50	54	57	59	60	0	
95	M	55	59	62	64	0	
100	M	M	60	64	67	0	
100	M	M	M	60	64	0	
90	M	M	M	M	65	0	

Using the matrix method we can introduce an initial solution given below. Note that it is degenerate and as before we have indicated in bold a zero which we wish to think of as an element of the basis and for which we shall force tightness in the corresponding dual inequality. We have also set as usual

[30] We are making the assumption that time travel is not possible.

$u_1 = 0$.

	60	80	85	100	70	80	
90	60 [50]	$30-\eta$ [54]	0 [57]	0 [59]	$0+\eta$ [60]	0 [0]	0
95	0 [M]	$50+\eta$ [55]	$45-\eta$ [59]	0 [62]	0 [64]	0 [0]	1
100	0 [M]	0 [M]	$40+\eta$ [60]	0 [64]	0 [67]	$60-\eta$ [0]	2
100	0 [M]	0 [M]	0 [M]	100 [60]	**0** [64]	0 [0]	1
90	0 [M]	0 [M]	0 [M]	0 [M]	$70-\eta$ [65]	$20+\eta$ [0]	2
	50	54	58	59	63	−2	·

We have also indicated a perturbation loop and this yields the solution (setting again $u_1 = 0$).

	60	80	85	100	70	80	
90	60 [50]	0 [54]	0 [57]	0 [59]	30 [60]	0 [0]	0
95	0 [M]	80 [55]	15 [59]	0 [62]	0 [64]	0 [0]	4
100	0 [M]	0 [M]	70 [60]	0 [64]	0 [67]	30 [0]	5
100	0 [M]	0 [M]	0 [M]	100 [60]	**0** [64]	0 [0]	4
90	0 [M]	0 [M]	0 [M]	0 [M]	40 [65]	50 [0]	5
	50	51	55	56	60	−5	$z = 22,885$

A quick scan reveals that $u_i + v_j \leq c_{ij}$ for all i, j and hence we have reached an optimal solution.

8 Optimisation over Networks

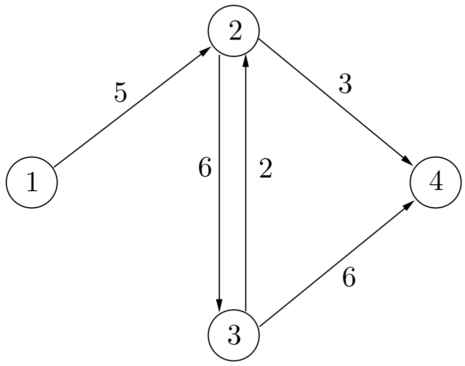
8.1 Capacitated Networks

A capacitated network (N, γ) consists of a set of nodes N and a set of **directed arcs**[31], which can be regarded as a subset of of $N \times N$. Associated with each arc is a non-negative real number γ which can be interpreted as the capacity of the arc. Formally γ is a a mapping $\gamma : N \times N \to [0, \infty)$ which satisfies

$$\gamma(i,j) \geq 0 \text{ for all } i,j \in N$$
$$\gamma(i,i) = 0 \text{ for all } i \in N.$$

A network may be represented either

diagramatically, for example,



or equivalently by a matrix of values of γ

$$\begin{pmatrix} 0 & 5 & 0 & 0 \\ 0 & 0 & 6 & 3 \\ 0 & 2 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

[31]To be clear, a directed arc simply means the direct path $i \to j$ where $i, j \in \{1, ..., N\}$.

8.2 Flows in Networks

Many problems are concerned with flows in networks where γ is interpreted as a capacity limiting the possible flow along an arc. One class of problems we shall be concerned with is finding the maximal flow through a network. In this we shall take some nodes, called sources, to be the origin of the flows and others, called sinks, to be where the flow is taken away. In general there can be many sources and sinks.

We need to introduce some new notation at this point.

1. Node $s \in N$ is a **source** iff

$$\gamma(i,s) = 0 \ \forall i \in N \quad \text{and} \quad \exists i \in N \ \ s.t. \ \ \gamma(s,i) > 0$$

2. Node d is a **sink** iff

$$\gamma(d,i) = 0 \ \forall i \in N \quad \text{and} \quad \exists i \in N \ \ s.t. \ \ \gamma(i,d) > 0$$

3. A **path** is a sequence of arcs of the form:

$$(i_1, i_2), (i_2, i_3), ..., (i_{k-1}, i_k)$$

   where $i_1, i_2, ..., i_k$ are distinct nodes.

4. A **flow** in a network is a function f defined on $N \times N$ satisfying:

   (a) $f(i,j) \leq \gamma(i,j) \ \ \forall i,j \in N$

   (b) $f(i,j) = -f(j,i)$

   (c) $\sum_{j \in N} f(i,j) = \sum_{j \in N} f(j,i) \ \ \forall i \in N \backslash (S \cup D)$

   where S and D are the sets of sources and sinks respectively. Condition (c) simply requires conservation of flow at all nodes other than sources or sinks. Actually, both sides of the equation in (c) are both zero as a result of (b).[32] The latter says that a flow in one direction is treated as equivalent to a negative flow in the opposite direction. This convention makes later computations somewhat more more convenient.

5. A path is **saturated** if it contains an arc (i,j) with $f(i,j) = \gamma(i,j)$.

[32] The point being that from (b) we have that $\sum_{j \in N} f(i,j) = -\sum_{j \in N} f(j,i) = \sum_{j \in N} f(j,i)$. Since the only number which is the negative of itself is zero, the claim follows.

8.3 Maximal flow problem and labelling algorithm

Henceforth we shall consider only networks with just one source and one sink.

For a network with single source and sink, **the maximal flow problem** is to determine arc flows $f(i,j)$ in order to maximise the total flow from the source, $A(f)$, which is defined by

$$A(f) = \sum_{i \in N} f(s,i).$$

In this section we shall describe the **labelling algorithm**.[33] In outline, the algorithm is as follows. Suppose that a flow through the network has been established.

Step 1: Find an unsaturated path from source s to sink d. If none exists, the flow is maximal.

Step 2: Augment the existing flows to saturate the path. Return to Step1.

This looks easy, but the complicating feature is that the path may involve traversing arcs in the opposite direction to the present flow i.e. augmenting the flow may involve **reducing** the existing flow along one or more arcs. The most reliable way to find an unsaturated path is to adopt the labelling procedure now described.

Step 1: Suppose we have been able to assign a flow[34] through the network, say $g(i,j)$, then calculate the **residual capacities**, $\gamma'(i,j)$, by the following

$$\gamma'(i,j) = \gamma(i,j) - g(i,j).$$

(a) Find all nodes i with the property

$$\gamma'(s,i) > 0$$

i.e all those nodes with an unsaturated arc from the source s. **Label** node i with (d_i, s) where $d_i = \gamma'(s,i)$.

[33] Also known as the Ford-Fulkerson labelling algorithm.

[34] We mean specifically which respects the definition of a **flow**.

(b) Find nodes j with the property

$$\gamma'(i,j) > 0 \text{ where node } i \text{ is } \textbf{labelled}.$$

Label node j with (d_j, i) where

$$d_j = min(\gamma'(i,j), d_i)$$

(c) Keep returning to Step 1 (b) until all nodes that can be labelled have been labelled.

Step 2: If the sink d is not labelled then all paths to the sink are saturated and the flow must be optimal. The optimal flow by the simple relation

$$g(i,j) = \gamma(i,j) - \gamma'(i,j)$$

where γ' are the residual capacities in the current round of the algorithm.

If on the other hand the sink d is labelled then there is an **unsaturated path** from source to sink and the labels will show the nodes which make up the path as well as the minimal residual capacity along the path.

We now create a new set of residual capacities for the network, $\gamma''(ij)$ as follows

$$\gamma''(i,j) = \gamma'(i,j) \text{ if } i,j \text{ do not belong to the unsaturated path}$$

and

$$\gamma''(i,j) = \gamma'(i,j) - c \text{ if } i,j \text{ belong to the unsaturated path}$$

where c is the minimal residual flow along the unsaturated path.

Now return to Step 1 (a) replacing γ by γ''.

This algorithm is to some extent incomplete as it does not tell us how to find an initial solution. Here is a rough description of how one may do this. Start at the source and find a path of arcs which connects the source to the sink. The maximum flow that can be allowed along this path is the minimum capacity of all the arcs along the path. Suppose this minimal capacity is c. Subtract this capacity from all the capacities along the the given path so that at least one arc has zero capacity. To get the best possible initial flow one can return to the source and proceed along another path with positive capacity at each arc and repeat the procedure and then continue in this way until there are no more paths possible from source to sink. Essentially this is going through the whole labelling procedure in one go and so it does not matter if not all paths are saturated on the initial flow as subsequent steps of the labelling algorithm will address these deficiencies.

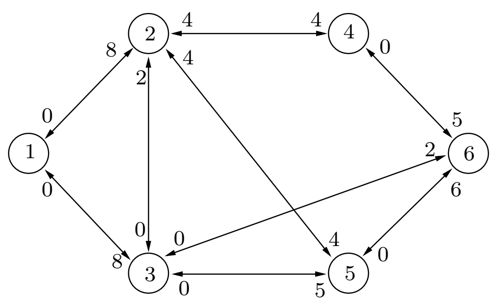
Things are best visualised in an example!

8.4 A worked example

Consider the network whose structure and capacities are described by the matrix

$$\begin{pmatrix} 0 & 8 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 & 0 \\ 0 & 2 & 0 & 0 & 5 & 2 \\ 0 & 4 & 0 & 0 & 0 & 5 \\ 0 & 4 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

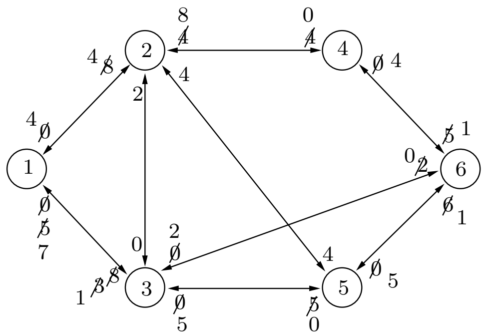
Which can otherwise be represented in diagrammatic form as follows.



We now construct an initial flow by looking at paths from the source to the sink and adding the minimal flow along these paths. In the diagram below we have inserted the residual capacities which follow from each included flow from source to sink. Note that final figures give us the initial values of $\gamma'(i,j)$ to work with in step 1 (a) of the algorithm.

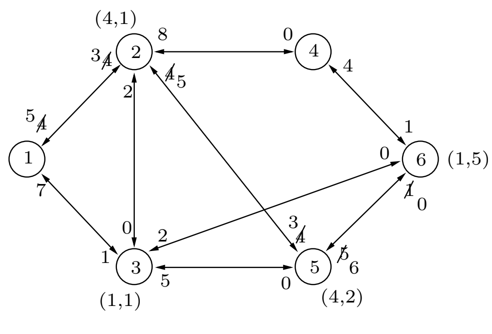
The paths we work with and their respective flows are

$$1 \to 2 \to 4 \to 6 \text{ flow } 4$$
$$1 \to 3 \to 5 \to 6 \text{ flow } 5$$
$$1 \to 3 \to 6 \text{ flow } 2$$



The reader will note that strictly speaking we could in principle add more flow by adding a unit flow to the path $1 \to 2 \to 5 \to 6$. However, as we shall see below this is essentially what the first step of the labelling algorithm will do anyway.

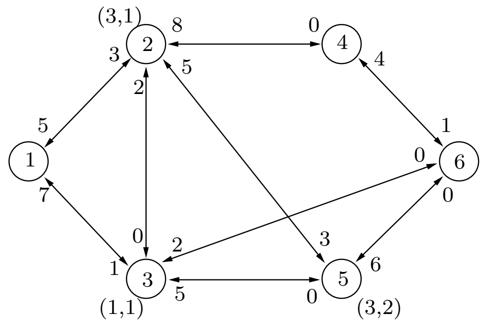
Now following Step 1, (b) and (c) we insert the labels on the nodes where possible and observe that the sink has been labelled which means that saturation has not been reached. On the same diagram we have computed the new set of residual capacities.



From the labels we see that a flow $1 \to 2 \to 5 \to 6$ of unit strength is still possible. We include this flow by augmenting the capacities again. With the new residual capacities we label the network as in the following diagram. Note that the sink is no longer labelled which means that the network is saturated. Using the relation that $g(i,j) = \gamma(i,j) - \gamma'(i,j)$ where γ' is the current residual capacities we find that

$$g(s,1) = 8 - 3 = 5 \text{ and } g(s,2) = 8 - 1 = 7$$

making the maximal flow $\sum_j g(s,j) = 12$.



Finally note that had we included the path $1 \to 2 \to 5 \to 6$ in the initial solution, the labelling algorithm would have concluded immediately that we have found the optimal solution.

8.5 Maximal flow and minimal cuts

The procedure described above is based upon ideas developed by Ford and Fulkerson in 1957. Central to the underlying theory is the idea of a **cut**, which is now described. This description again is limited to the case of netwoks having a single source and sink.

We need to introduce yet more notation.

1. A **cut** in a network is an ordered pair (A, B) of subsets of N such that

   (a) $s \in A$

   (b) $d \in B$

(c) $A \cup B = N$

(d) $A \cap B = \emptyset$

2. The **capacity of a cut** $\gamma(A, B)$ is defined as

$$\gamma(A, B) = \sum_{i \in A} \sum_{j \in B} \gamma(i, j)$$

This same notation will be used whenever A and B are subsets of N without necessarily being a cut.

3. If A and B are subsets of N, then the **flow from** A **to** B is defined as

$$f(A, B) = \sum_{i \in A} \sum_{j \in B} f(i, j)$$

Since for any $i, j \in N, f(i, j) \leq \gamma(i, j)$, double summation gives

$$f(A, B) \leq \gamma(A, B) \tag{18}$$

for any cut (A, B).

Lemma 8.1 *For any cut* (A, B) *and any flow* f*, the amount of flow in the network equals the flow across* (A, B)*. That is to say,*

$$A(f) = f(A, B).$$

Proof. Note that by definition of a flow we $\sum_{j \in N} f(i, j) = 0$ for any $i \in N \backslash \{s, d\}$. Now note that

$$\begin{aligned} A(f) &= \sum_{j \in N} f(s, j) + 0 \\ &= \sum_{j \in N} f(s, j) + \sum_{i \in A, i \neq s} \sum_{j \in N} f(i, j) \\ &= \sum_{i \in A} \sum_{j \in N} f(i, j) \\ &= \sum_{i \in A} \sum_{j \in A} f(i, j) + \sum_{i \in A} \sum_{j \in B} f(i, j) \\ &= 0 + \sum_{i \in A} \sum_{j \in B} f(i, j) \\ &= f(A, B) \end{aligned}$$

where in the penultimate equality, the terms $f(i,j)$ and $f(j,i) = -f(j,i)$ appear in the double sum for $i,j \in A$ which cancel one another out. ■

There is a nice intuition behind the above lemma. Think of arcs as pipes. Imagine that we are pumping fluid into the network at the source and the capacities represent the maximal flow of liquid that can pass through each pipe. A cut is a way of severing the network so that all liquid pumped in at the source leaks out of the system. Naturally, if this is the case then the total flow of the 'leakage' must match the flow pumped in at the source. This is what the lemma says.

Combining the conclusion of this lemma with this with the inequality (18), we see that for any cut (A,B),

$$A(f) \leq \gamma(A,B). \tag{19}$$

In other words the amount of flow cannot exceed the capacity of any cut. This inequality allows us to deduce that if for some flow f we can find a cut (A',B') such that

$$A(f) = \gamma(A',B')$$

then it must be the case that f is an optimal flow. Indeed in that case then by (19) we necessarily have that

$$\gamma(A',B') = A(f) \leq \max_g A(g) \leq \min_{(A,B)} \gamma(A,B) \leq \gamma(A',B')$$

where the maximum is taken over all flows g and the minimum is taken over call cuts (A,B). Hence the inequalities above must be equalities and we see that $A(f)$ is optimal, (A',B') is a minimal cut in the sense that

$$\gamma(A',B') = \min_{(A,B)} \gamma(A,B)$$

and $A(f) = \min_{(A,B)} \gamma(A,B)$.

Below, the celebrated maximal flow/minimal cut theorem, provides the converse to this conclusion.

Theorem 8.2 (Maximal flow/minimal cut theorem) *Let f be a maximal flow, then there is a cut (A',B') such that*

$$A(f) = \gamma(A',B') = \min_{(A,B)} \gamma(A,B).$$

Thus the maximal flow at the source equals the minimal capacity across all cuts.

Proof. Let A' be the set of nodes which have been labelled in the final step of the algorithm together with the source and let B' be the set of unlabelled notes. Clearly $A' \cup B' = N$ and $A' \cap B' = \emptyset$. We also claim that (A', B') is a cut. To verify this claim we need to show that $d \in B'$. However this is evident by virtue of the fact that the algorithm stops when the sink is not labelled.

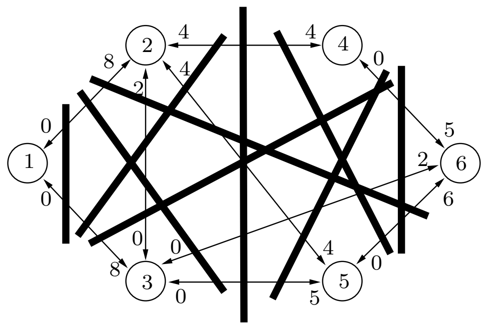
Now recall from Lemma 8.1 that we must necessarily have $A(f) = f(A', B')$. However, by virtue of the fact that the capacity of arcs from labeled nodes in A' to unlabeled nodes in B' must be saturated (this follows by definition of the terminal labeling), it follows that

$$f(A', B') = \sum_{i \in A'} \sum_{j \in B'} f(i,j) = \sum_{i \in A'} \sum_{j \in B'} \gamma(i,j) = \gamma(A', B').$$

Hence the discussion preceding this theorem implies the required result. ■

8.6 Example revisited

In the previously discussed example we computed the maximal flow to be equal to 12. In the diagram below we see all the possible cuts of the network and the associated capacities of the cuts.



Note that all the cuts are described as follows (we leave the reader to compute their capacites and verity that 12 is indeed the minimal flow across a cut).

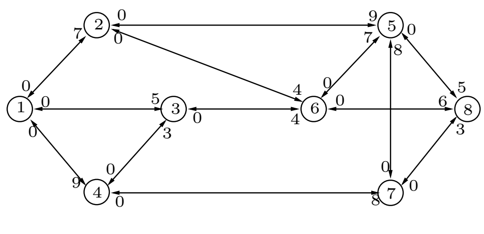
$$\{1\}\{2,3,4,5,6\}$$
$$\{1,2\}\{3,4,5,6\}$$
$$\{1,3\}\{2,4,5,6\}$$
$$\{1,2,3\}\{4,5,6\}$$
$$\{1,2,4\}\{3,5,6\}$$
$$\{1,3,5\}\{2,4,6\}$$
$$\{1,2,3,4\}\{5,6\}$$
$$\{1,2,3,5\}\{4,6\}*$$
$$\{1,2,3,4,5\}\{6\}$$

The cut marked with a $*$ is a minimal cut.

It is interesting to note that what is essentially going on here is Duality in disguise. Theorem 8.2 essentially says that if we think of the optimisation of flow over our given network as a linear programming problem, then when the the primal problem has a solution then so does the dual and their objectives are equal. The dual problem clearly manifests itself in cuts over the network. We leave the reader to puzzle over what **precisely** the dual problem is!

8.7 A second worked example

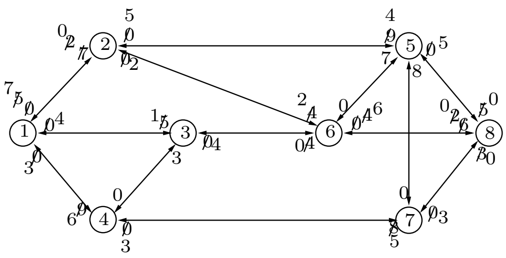
We are required to establish the optimal flow in the the following network



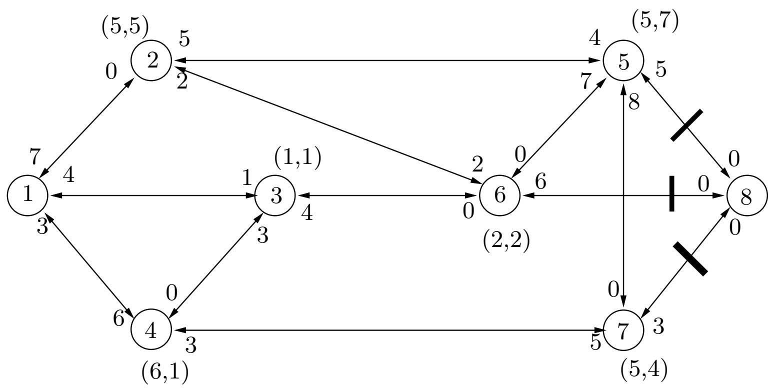
Note that this network has a source at node 1 and a sink at note 8. We identify the following paths and flows.

$$1 \to 2 \to 5 \to 8 \text{ flow } 5$$
$$1 \to 3 \to 6 \to 8 \text{ flow } 4$$
$$1 \to 4 \to 7 \to 8 \text{ flow } 3$$
$$1 \to 2 \to 6 \to 8 \text{ flow } 2$$

which results in the following residual capacities.



With a sharp eye, it should now be clear that we have saturated the network. However, to be sure, we can implement the labelling procedure to obtain the diagram below.



Since the sink at 8 is not labelled the network must be saturated. The proof of the maximum flow/minimal cut theorem tells us that the minimal cut corresponds to the the case that A is the source together with all labelled nodes and B is the remaining nodes. We have indicated this cut too in the diagram which corresponds to $\{1, 3, 4, 5, 6, 7\}\{8\}$. Note that the flow from the source is equal to $7 + 4 + 3 = 14$. On the other hand the sum of the capacities across the indicated cut are equal to $5 + 6 + 3 = 14$.