Mathematical Programming 2016

Dr. Valentin Popov

`vmp@st-andrews.ac.uk`

General information

- **Lecturer and Module Coordinator:** Dr. Valentin Popov
- **e-mail address:** vmp@st-andrews.ac.uk
- **Office:** CREEM
- **Prerequisite**: MT2001 or MT2501 (or MT1002 and MN2002)
- **Lectures**: At 12 noon on Wednesdays, Fridays and odd Mondays in Theatre B of the Maths Institute
- **Tutorials/Feedback sessions**: Fortnightly, either 9 a.m. on Thursday in Theatre A, or at 3 p.m. on Thursday in 1A starting in week 3
- **Assessment**: 2 Hour Examination = 100%

Syllabus

1. **Linear Programs (LPs) [2]**
   Introduction, graphical solution, slack and surplus variables.
2. **The simplex algorithm [7]**
   Basic and non-basic variables, setting up an initial tableau, applying the simplex algorithm, finding an initial vertex, special cases, sensitivity analysis, matrix representation of the simplex procedure.
3. **Duality [3]**
   Definition of the dual problem, duals of non-canonical problems, relationships between the primal and the dual, complementary slackness.
4. **Integer Programming [1]**
   Introduction, the branch-and-bound algorithm.

5. **Solving a LP using a spreadsheet [1]**
   Using the Solver in Excel, using Simplextab.

6. **Transportation Problems (TPs) [5]**
   LP formulation, the dual of a TP, finding an initial vertex, cell evaluations, the general step of the transportation algorithm, unbalanced problems, sensitivity analysis.

7. **Transshipment Problems (TsPs) [1.5]**
   Buffer stocks approach, cheapest route approach, shortest route problems as TsPs.

8. **Network Flow Problems [1.5]**
   A more graphical approach, the network flow algorithm.

9. **Assignment Problems [2]**
   LP formulation, the Hungarian assignment algorithm.

References

- Wayne L Winston : *Operations Research - applications and algorithms* (4th ed.)
- Hamdy A Taha : *Operations Research - an introduction* (9th ed.)

Outline

Example A (Canvassing)

During an election, an agent for a particular candidate has to decide how best to deploy his volunteer canvassers for the rest of the campaign. He calculates that they will be able to provide 180 hours of work in total. He reckons that a telephone canvasser can contact 20 households an hour, whilst a canvasser going door-to-door will speak to only 15 households each hour. Some of his volunteers are elderly and infirm and can only canvass by telephone, so the amount of door-to-door work will not be able to exceed 150 hours. The candidate, however, wishes to create an impact on the streets and has decreed that at least twice as many hours should be spent on door-to-door work as on the telephone.

How should the canvassers be deployed to maximise the number of households contacted?

Stating the problem

Define

x_1 = number of hours devoted to telephone canvassing;
x_2 = number of hours devoted to door-to-door canvassing;
Z = total number of households contacted.

The agent's task is thus to:

$$\begin{array}{llrcll}
\textbf{Maximise} & & Z = 20x_1 & + & 15x_2 & & (1) \\
\text{subject to} & & x_1 + x_2 & \leq & 180 & \text{(total no. of hours available)} & (2) \\
& & x_2 & \leq & 150 & \text{(max. for door-to-door work)} & (3) \\
& & 2x_1 - x_2 & \leq & 0 & \text{(ratio of telephone to door-to-door)} & (4)
\end{array}$$

We must also have

$$x_1 \geq 0, \;\; x_2 \geq 0, \qquad (5)$$

since it clearly would not make sense for either x_1 or x_2 to be negative.

Basic concepts

- The above canvassing problem is an example of a **Linear Program** (LP).
- This particular problem has two **decision variables,** namely x_1 and x_2.
- Expression (1) is called the **objective function** (OF). In this case it is the quantity that we wish to maximise: in other LPs we may, for instance, be looking at costs and so wish to minimise the OF.
- Inequalities (2) to (5) are referred to as **constraints**. The name Linear Program is appropriate for the canvassing problem as the OF and all the constraints are linear functions of the decision variables x_1 and x_2.

The general class of LPs with *n* decision variables and *m* constraints

$$\begin{array}{rrcl} \textbf{Maximise} & Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n & & \\ & & & \\ \text{subject to} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & \leq & b_1, \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & \leq & b_2, \\ & & \vdots & \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & \leq & b_m, \\ & & & \\ & x_1 \geq 0, \;\; x_2 \geq 0, \; \ldots \; , x_n \geq 0. & & \end{array} \tag{6}$$

Define column vectors

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

and a matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$

Matrix form of a LP

$$\begin{array}{ll} \textbf{Maximise} & \mathbf{Z} = \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{A}\,\mathbf{x} \leq \mathbf{b} \text{ and } \mathbf{x} \geq 0. \end{array} \tag{7}$$

Example A

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 20 \\ 15 \end{pmatrix},$$

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 2 & -1 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} 180 \\ 150 \\ 0 \end{pmatrix}$$

Consequently

$$\mathbf{c}^T \mathbf{x} = (20 \quad 15)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 20x_1 + 15x_2, \quad \text{and}$$

$$\mathbf{A}\,\mathbf{x} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 2 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_2 \\ 2x_1 - x_2 \end{pmatrix}.$$

Assumptions

- **Certainty**

  It is implicit in the above description that the parameters of the problem, the elements of **A**, **b** and **c**, are known with certainty.

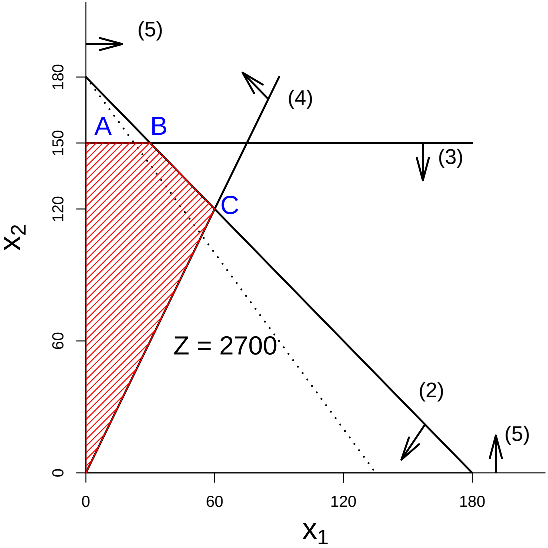
- **Divisibility**

  We also assume until further notice that each decision variable can assume fractional values. We will look later at integer linear programs in which some or all of the decision variables must be non-negative integers.

1. Linear Programs (LPs)
    - Introduction
    - Graphical solution
    - Slack and surplus variables
2. The Simplex algorithm
    - Basic and non-basic variables
    - Setting up an initial tableau
    - Applying the simplex algorithm
    - Finding an initial vertex
    - Special cases
    - Sensitivity Analysis
    - Matrix representation of the simplex procedure

- The set of values of $\mathbf{x}$ which satisfy all the constraints is usually called the **feasible region (FR)**. Taha refers to it as the **feasible solution space**.
- In LPs with only two decision variables, the constraints correspond to straight lines in the $(x_1,\ x_2)$ plane, implying that the FR is usually an area of the plane bounded by straight lines (but could be just a line segment or a point).
- These two-dimensional problems can be solved graphically

Example A



- Each constraint is numbered according to its corresponding inequality. The points that satisfy the constraint lie on the side of the line to which the arrow points.

- Thus the FR is the quadrilateral OABC.

- The broken line gives the contour $Z = k$ of the OF for the case $k = 2700$.

- As k is increased, the contour moves away from the origin, retaining its gradient.

- The optimal solution could be found by comparing the gradient of the OF with the gradients of the boundary of the FR, to find which point(s) lie on the highest contour of Z.

- Given the linearity of the edges of the FR, it is, however, obvious that the optimal value of Z must occur either at a vertex of the FR, or at all points on a line segment joining two vertices.

- Thus the easiest way to find the optimal solution is usually to evaluate Z at each vertex.
- In the present case we obtain:

$(x_1,\ x_2)$	$O = (0,0)$	$A = (0,150)$	$B = (30,150)$	$C = (60,120)$
Z	0	2250	2850	3000

Definition 1 (Optimal solution)

In a LP of $\left\{\begin{array}{c}\text{maximisation}\\ \text{minimisation}\end{array}\right\}$ type, an **optimal solution** is a point in the FR with the $\left\{\begin{array}{c}\text{largest}\\ \text{smallest}\end{array}\right\}$ value of the OF.

$\Rightarrow$ Hence the optimal solution in the canvassing example is given by $(x_1, x_2) = (60, 120)$, resulting in a total of 3000 households being contacted.

Number of solutions

- It is clear from the canvassing example that, if a LP problem with two decision variables has an optimal solution, there will always be *a vertex* of the FR which yields the optimal value of Z.
- Sometimes the optimal solution is non-unique and *two adjacent vertices* are both optimal.
- When this arises the contours of Z must be **parallel** to the edge joining the two vertices, and in this case any point on that edge is also an optimal solution.

General case

- When there are n decision variables, the constraints and the contours of the objective function represent hyperplanes in $\mathbb{R}^n$.
- It can be shown that any (non-empty) FR will be a *convex set*.

Definition 2 (Convex set)

A set S is a **convex set** if the line segment joining any pair of points in S is wholly contained in S.

Remarks

- It remains true that, if a solution exists, there will always be *a vertex* of the FR that yields the optimal value of Z
- the solution need not be unique.
- e.g. in three dimensions, there may be
  - no optimal solution,
  - a unique optimum at a vertex of the FR,
  - an infinite number of optimal solutions along an edge of the FR,
  - or an infinite number of optimal solutions located on a plane bounding the FR.

1. Linear Programs (LPs)
    - Introduction
    - Graphical solution
    - Slack and surplus variables
2. The Simplex algorithm
    - Basic and non-basic variables
    - Setting up an initial tableau
    - Applying the simplex algorithm
    - Finding an initial vertex
    - Special cases
    - Sensitivity Analysis
    - Matrix representation of the simplex procedure

- Inequality constraints in a LP can be turned into equalities by introducing **slack** or **surplus** variables.
- An LP with m inequality constraints (ignoring non-negativity ones) will need m such variables.

Example A (Canvassing)

Here the inequalities were of $\leq$ type, so we convert each of the constraints (2), (3) and (4) into an equality by introducing a slack variable into each one. So we may write

$$\begin{aligned} x_1 + x_2 + s_1 &= 180, \\ x_2 + s_2 &= 150, \\ 2x_1 - x_2 + s_3 &= 0 \end{aligned}$$

where we now require that $s_1 \geq 0,\ s_2 \geq 0,\ s_3 \geq 0.$

A minimisation problem

A similar approach works in minimisation problems:

Example B (Diet)

A dairy company was investigating diets, based on only bread and cheese, that would yield the daily needs for protein, fat and carbohydrates. The nutritional details were as follows:

	Grams per 250g			Calories
Food	Protein	Fat	Carbohydrates	per 250g of food
Wholemeal bread	2.0	0.5	10.0	40
Cheese	6.0	8.0	0.0	100

The required daily amounts were 72g of protein, 68g of fat and 240g of carbohydrates. The company wished to determine the lowest daily calorie intake that would produce a healthy diet.

Formal statement of the problem

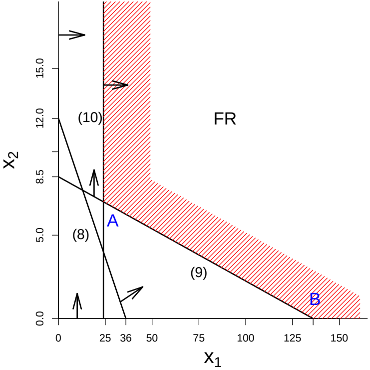
Let

x_1 = amount of bread required, in units of 250g;
x_2 = amount of cheese required, in units of 250g;
Z = total intake in calories.

The company wish to:

$$\begin{array}{llrcrr} \textbf{Minimise} & Z = 40x_1 + 100x_2 & & & & \\ \text{subject to} & & 2x_1 + 6x_2 & \geq & 72 & (8) \\ & & 0.5x_1 + 8x_2 & \geq & 68 & (9) \\ & & 10x_1 & \geq & 240 & (10) \\ & & x_1 \geq 0,\ x_2 & \geq & 0 & \end{array}$$

Example B: Plot of the FR



Observe that constraint (8) does not touch the FR, and is effectively irrelevant. Such a constraint is called **redundant**. In this case the non-negativity constraint $x_1 \geq 0$ is also redundant.

Here we can convert the constraint inequalities to equalities using non-negative surplus variables $s_1,\ s_2$ and s_3 such that

$$\begin{aligned} 2x_1 + 6x_2 - s_1 &= 72, \\ 0.5x_1 + 8x_2 - s_2 &= 68, \\ 10x_1 - s_3 &= 240 \end{aligned}$$

Observe that the on each edge of the FR one of the decision or surplus variables is zero:

Edge	$\{(24,\ x_2)\ :\ x_2\ \geq\ 7\}$	AB	$\{(x_1,\ 0)\ :\ x_1\ \geq\ 136\}$
Zero variable	s_3	s_2	x_2

Hence, at each of the two vertices A and B of the FR, there are two variables equal to zero.

Example A (Canvassing)

Similarly here, on each edge of the FR, either a decision variable or one of the slack variables (introduced on slide 24) is zero:

Edge	OA	AB	BC	CO
Zero variable	x_1	s_2	s_1	s_3

Note

At the vertices A, B and C, there are again two variables equal to zero, but at O the three variables x_1, x_2 and s_3 are all zero. (The edge on which $x_2 = 0$ is degenerate, corresponding to a single point.)

References for Chapter 1

- *Winston* :
    - §3.2 Graphical solution
    - §4.1 Slack and surplus variables
- *Taha* :
    - §2.2 Graphical solution
    - §3.1.1 Slack and surplus variables

Outline

1. Linear Programs (LPs)
    - Introduction
    - Graphical solution
    - Slack and surplus variables
2. The Simplex algorithm
    - Basic and non-basic variables
    - Setting up an initial tableau
    - Applying the simplex algorithm
    - Finding an initial vertex
    - Special cases
    - Sensitivity Analysis
    - Matrix representation of the simplex procedure

1. Linear Programs (LPs)
    - Introduction
    - Graphical solution
    - Slack and surplus variables

2. The Simplex algorithm
    - Basic and non-basic variables
    - Setting up an initial tableau
    - Applying the simplex algorithm
    - Finding an initial vertex
    - Special cases
    - Sensitivity Analysis
    - Matrix representation of the simplex procedure

Basic solution

- In general, introducing slack or surplus variables into the m constraints yields m equations in $m + n$ unknowns (n decision variables plus m slack or surplus ones).
- Suppose we set n of these variables to zero.
- Assuming that the resulting m linear equations in m unknowns are consistent, and that there are no redundant equations, we can solve them to find what is called a **basic solution**.

Example A (Canvassing)

- Setting $x_2 = s_1 = 0$, the constraint equations on slide 24 now read

$$x_1 = 180, \quad s_2 = 150, \quad 2x_1 + s_3 = 0 \quad \Rightarrow \quad s_3 = -360$$

- So this is a basic solution, but it is not a vertex of the feasible region.

- It does not give a feasible solution of the LP, since $s_3 < 0$ and hence constraint (4) is not satisfied.

Definition 3 (Feasible basic solution, basic / non-basic variables)

A non-negative basic solution is called a **feasible basic solution** (fbs), and the solution variables are **basic variables**. The variables set equal to zero are **non-basic variables**.

Example C

Consider the LP:

$$\begin{aligned} \textbf{Maximise } Z &= 4x_1 + 5x_2 \\ \text{subject to} \quad 2x_1 + x_2 &\leq 15 \\ x_1 + x_2 &\leq 8 \\ x_1 + 2x_2 &\leq 14 \\ x_1 \geq 0, \; x_2 &\geq 0 \end{aligned}$$

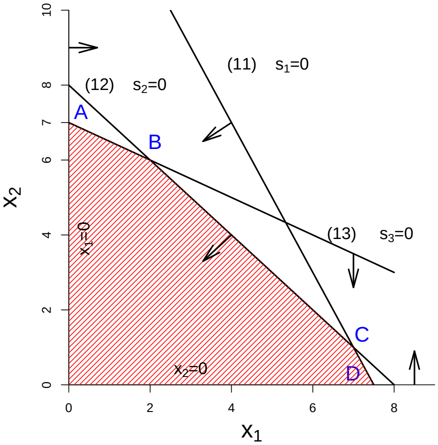
or, equivalently, using slack variables,

$$2x_1 + x_2 + s_1 = 15 \tag{11}$$

$$x_1 + x_2 + s_2 = 8 \tag{12}$$

$$x_1 + 2x_2 + s_3 = 14 \tag{13}$$

Example C: Feasible region



Example C (cont.)

The basic and non-basic variables at each vertex are as follows:

Vertex	O	A	B	C	D
Non-b. var.	x_1, x_2	x_1, s_3	s_2, s_3	s_1, s_2	x_2, s_1
B. var.	s_1, s_2, s_3	x_2, s_1, s_2	x_1, x_2, s_1	x_1, x_2, s_3	x_1, s_2, s_3

where “Non-b. var.” stands for Non-basic variables and respectively “B. var.” stands for Basic variables

Observe that, on each occasion, a move from one vertex to an adjacent one involves changing one of the basic variables with a non-basic one.

The simplex algorithm: introduction

This is how the simplex algorithm operates.

- Starting from an initial vertex, we move, in each iteration, between adjacent vertices by replacing one of the basic variables with a non-basic one until we reach the optimal solution.
- In a maximisation problem the choice of non-basic variable is made by selecting the direction in which the objective function increases most steeply.
- In many maximisation problems, the origin serves as an initial vertex.

Example C

- In order to check whether optimality has been attained, or, if not, in which direction to move, both the basic variables and the objective function must be expressed in terms of non-basic variables.

- If we start at the origin, we already have these quantities in the required form, and clearly at O we have $Z = 0$. Consider the effect of a small increase in one of the non-basic variables, x_1 and x_2.

- Since $Z = 4x_1 + 5x_2$, the more rapid increase is obtained by making x_2 non-zero (i.e. making it a basic variable).

- By how much can we increase it?

Example C (cont.)

- Recalling that the slack variables must be non-negative, and keeping x_1 equal to zero

$$\left\{\begin{array}{llllll} \text{constraint} & (11) & \Rightarrow & s_1 = 15 - 2x_1 - x_2 & \Rightarrow & x_2 \leq 15 \\ \text{constraint} & (12) & \Rightarrow & s_2 = 8 - x_1 - x_2 & \Rightarrow & x_2 \leq 8 \\ \text{constraint} & (13) & \Rightarrow & s_3 = 14 - x_1 - 2x_2 & \Rightarrow & x_2 \leq 7 \end{array}\right.$$

- So the largest that x_2 can be is 7.

- So we set $(x_1,\ x_2,\ s_1,\ s_2,\ s_3) = (0, 7, 8, 1, 0)$ i.e we move to vertex A.

Example C (cont.)

- We must now express Z and the constraints in terms of the new non-basic variables.
- Constraint (13) gives $x_2 \;=\; 0.5(14 - x_1 - s_3)$

$$\Rightarrow Z = 4x_1 + 2.5(14 - x_1 - s_3) = 35 + 1.5x_1 - 2.5s_3 \qquad (14)$$

- So to increase Z further we must increase x_1.
- To determine by how much we can do this, we again use the constraints, noting that we are keeping s_3 equal to zero.

Example C (cont.)

- As above, constraint (13) gives

$$x_2 = 0.5(14 - x_1 - s_3) \quad \Rightarrow \quad x_1 \leq 14.$$

- Constraint (11) gives

$$s_1 = 15 - 2x_1 - 0.5(14 - x_1 - s_3) = 8 - 1.5x_1 + 0.5s_3$$

$$\Rightarrow x_1 \leq 5.333.$$

- Constraint (12) gives

$$s_2 = 8 - x_1 - 0.5(14 - x_1 - s_3) = 1 - 0.5x_1 + 0.5s_3 \tag{15}$$

$$\Rightarrow x_1 \leq 2.$$

- So the largest that x_1 can be is 2.
- So we set $(x_1, \ x_2, \ s_1, \ s_2, \ s_3) = (2, 6, 5, 0, 0)$ i.e we move to vertex B.

Example C (cont.)

- Again we must express Z in terms of the new non-basic variables, s_2 and s_3.
- We note from (15) that $x_1 = 2 - 2s_2 + s_3$.
- Hence, from (14), we get

$$Z = 35 + 1.5(2 - 2s_2 + s_3) - 2.5s_3 = 38 - 3s_2 - s_3$$

- As the coefficients of s_2 and s_3 here are both negative, no further increase in the value of the objective function is possible. i.e vertex B is optimal.

The methodology of the above example illustrates that used by the simplex algorithm. As we shall see, however, the required calculations are more conveniently set out in tableaux.

Standard form of the LP

In order to apply the version of the algorithm given in the next section, the LP must first be cast in a *standard form*, in which

(i) all constraints (except for non-negativity ones) are expressed as equations;

(ii) all variables are restricted to be non-negative;

(iii) the objective function is to be maximised.

We have already seen how to meet condition (i) using slack and surplus variables. If we have a unrestricted variable x_i which can take any real value, condition (ii) may be met by defining two new variables x_i^+ and x_i^- such that $x_i = x_i^+ - x_i^-$, where $x_i^+ \geq 0$ and $x_i^- \geq 0$. Condition (iii) is not hard to meet, since minimising Z is equivalent to maximising $-Z$.

- When applying the simplex algorithm manually, the standard way of performing the calculations is to use tableaux.

- In the main body of each tableau, the rows correspond to the constraints and the columns to the variables.

- At each iteration of the algorithm, the entries in a particular row in the body of the tableau are just the coefficients of the variables in the version of the relevant constraint equation appropriate for that iteration.

- To start the algorithm, it is necessary to set up an initial tableau. For this we need an initial vertex i.e. an initial fbs. Equivalently, we wish to put the constraint equations into an appropriate form.

The initial tableau

Required form of the equations for the initial tableau

(i) Write the m constraint equations so that each equation contains exactly one of the m basic variables of the fbs, with this variable appearing in no other equation.

(ii) Scale each equation to ensure that the basic variable has a coefficient of 1. For a fbs, the right-hand side of each constraint equation must be non-negative, and clearly it gives the value of the corresponding basic variable at that vertex.

This is easily done in a maximisation problem in which the origin provides a fbs.

Example C

- Using slack variables, as in equations (11) to (13), we can write this LP as

$$\begin{array}{lrcl}
\textbf{Maximise } Z & = & 4x_1 + 5x_2 & \\
\text{subject to} & 2x_1 + x_2 + s_1 & = & 15 \\
 & x_1 + x_2 + \quad s_2 & = & 8 \\
 & x_1 + 2x_2 \qquad + s_3 & = & 14,
\end{array}$$

  with all the variables being non-negative.

- Note that the three requirements of the standard form are all met.
- Moreover, looking at the basic variables at the origin, we see that s_i appears only in the ith constraint ($i = 1, 2, 3$) and with coefficient 1.
- The values of s_1, s_2 and s_3 at the origin are 15, 8 and 14 respectively, as given on the RHS.

Example C Initial tableau

Check	Basic	x_1	x_2	s_1	s_2	s_3	RHS	ratio	
19	s_1	2	1	1	0	0	15	15	
11	s_2	1	1	0	1	0	8	8	
18	s_3	1	②	0	0	1	14	7	←
-9	Z	-4	-5	0	0	0	0		
			↑						

Explanations to the initial tableau

- In the second column, in the body of the table, we list the basic variables for the vertex. The entries on each of these rows, in the variable and the RHS columns, are just the coefficients from the above equations.
- The final row is the **objective function row** in which the coefficients in the expression for Z appear with their signs reversed (Equivalently, put all terms on the LHS of the equation.)
- The first column is an optional check on later computations, and simply gives the sum, for each row, of the entries in the variable and the RHS columns.

1. Linear Programs (LPs)
    - Introduction
    - Graphical solution
    - Slack and surplus variables
2. The Simplex algorithm
    - Basic and non-basic variables
    - Setting up an initial tableau
    - Applying the simplex algorithm
    - Finding an initial vertex
    - Special cases
    - Sensitivity Analysis
    - Matrix representation of the simplex procedure

Example C

- Having set up the initial tableau, we move to step (ii) of the Simplex Algorithm (see slide 52).
- So here the entering variable is x_2.
- Observe that the ratios obtained by following step (iii) are the values that we obtained in §2.1 when considering by how much we could increase x_2.
- Thus the leaving variable here is s_3.
- Having identified the pivot element, we proceed to create the tableau for the next vertex, which is vertex A, at which the basic variables are x_2, s_1 and s_2.

The Simplex Algorithm

(i) Find an initial fbs, and set up the initial tableau for this vertex.

(ii) Mark the variable with the most negative entry in the objective function row. (Break any tie arbitrarily.) This identifies the **pivot column** and the **entering variable** that will become basic.

(iii) Ignoring any row in which the pivot column entry is negative, calculate, for each *constraint* row, the ratio of the RHS divided by the corresponding entry in the pivot column.

(iv) Mark the row with the smallest such ratio. (Again break any tie arbitrarily.) This is the **pivot row** and identifies the **leaving variable** which is to become non-basic.

The Simplex Algorithm (cont.)

(v) Ring the entry that lies in both the pivot row and the pivot column. This is the **pivot element**.

(vi) Create a new tableau by

  (a) replacing the leaving variable by the entering variable in the "Basic" column;

  (b) dividing the elements of the pivot row by the pivot element;

  (c) taking each of the other rows (including the objective function row) in turn and adding to it (or subtracting from it) a suitable multiple of the pivot row to create a zero in the pivot column.

(vii) Generate further tableaux, by repeating steps (ii) to (vi) until all the variables have entries in the objective function row which are non-negative. An optimal solution has then been found, and the values of the basic variables are as in the RHS column of the tableau.

Example C: tableau for vertex A

Check	Basic	x_1	x_2	s_1	s_2	s_3	RHS	ratio	
10	s_1	1.5	0	1	0	-0.5	8	5.333	
2	s_2	$\boxed{0.5}$	0	0	1	-0.5	1	2	←
9	x_2	0.5	1	0	0	0.5	7	14	
36	Z	-1.5	0	0	0	2.5	35		
		↑							

Observe that the entries in the objective function row tally with the coefficients of the variables in expression (14) for Z, and the value of Z at vertex A is 35 as shown in the RHS column. Note that the ratios are now the values that we obtained when considering by how much we could increase x_1, and that the entries in the body of the tableau tally with the equations used to calculate those ratios.

- This time the entering variable is x_1 and the leaving variable is s_2.
- We move to vertex B, at which the basic variables are x_1, x_2 and s_1.
- The tableau for vertex B reads:

Check	Basic	x_1	x_2	s_1	s_2	s_3	RHS	ratio
4	s_1	0	0	1	-3	1	5	
4	x_1	1	0	0	2	-1	2	
7	x_2	0	1	0	-1	1	6	
42	Z	0	0	0	3	1	38	

- The entries in the objective function row now tally with the final expression for Z obtained in §2.1.
- All the variables have entries in that row which are non-negative, so we have an optimal solution. As shown in the RHS column, this is given by
  $x_1 = 2$, $x_2 = 6$.

Objective functions

- The objective function rows of the above three tableaux can be read as saying:

Initial tableau $Z - 4x_1 - 5x_2 = 0$

(Obvious from original statement of problem)

Second tableau $Z - 1.5\,x_1 + 2.5\,s_3 = 35$

(c.f. (14))

Final tableau $Z + 3\,s_2 + s_3 = 38$

$\Rightarrow Z = 38 - 3\,s_2 - s_3$

- So, as we saw on slide 42, if either s_2 or s_3 is made positive, it will reduce the value of Z. This implies that we have a unique optimum.

Uniqueness and Non-Uniqueness

- More generally, if once an optimal solution has been found, the entries in the objective function row in the columns for the non-basic variables are positive, the optimal solution is unique.
- It can happen, however, that one or more of these entries is zero.
- If a non-basic variable has a zero coefficient, that variable can be increased whilst leaving the value of Z unaltered.
- It is then possible to move to another vertex of the FR which has the same optimal value of Z.

Uniqueness and Non-Uniqueness (cont.)

- So, in this case, there would be alternative optimal solutions.
- In a commercial LP problem the existence of an alternative optimum may offer a company a choice of strategies offering the best return.

Note

Manipulation of the tableau is equivalent to solving the simultaneous linear equations by the Gauss-Jordan method. We express the new basic variables in terms of the non-basic ones, so that only one basic variable appears in each equation and with coefficient one (cf. slide 46)

Obtaining an initial vertex by inspection

- In some problems the origin is not a vertex of the FR.
- One possible approach is to *obtain an initial vertex by inspection*

Example B (Diet)

After the introduction of surplus variables, the constraint equations on slide 27 read as follows:

$$\begin{array}{rcrcl} 2x_1 + 6x_2 - s_1 & & & = & 72, \\ 0.5x_1 + 8x_2 & - s_2 & & = & 68, \\ 10x_1 & & - s_3 & = & 240 \end{array}$$

Obtaining an initial vertex by inspection (cont.)

- The origin $(x_1,\ x_2)\ =\ (0,\ 0)$ does not give a fbs since all the surplus variables would be negative.
- Suppose, however, that we put $s_2 = s_3 = 0$

$$\Rightarrow x_1 = 24, \quad x_2 = 7, \quad s_1 = 18.$$

- This gives a fbs. To apply the simplex algorithm we need to express the constraints in the required form (See box on slide 46):

$$\begin{aligned}
x_1 &= 24 + \frac{1}{10}s_3 \\
x_2 &= \frac{17}{2} + \frac{1}{8}s_2 - \frac{1}{16}\left(24 + \frac{1}{10}s_3\right) = 7 + \frac{1}{8}s_2 - \frac{1}{160}s_3 \\
s_1 &= 2\left(24 + \frac{1}{10}s_3\right) + 6\left(7 + \frac{1}{8}s_2 - \frac{1}{160}s_3\right) - 72 = 18 + \frac{3}{4}s_2 + \frac{13}{80}s_3
\end{aligned}$$

Obtaining an initial vertex by inspection (cont.)

- These equations can be re-arranged, putting all the variables on the LHS.
- We wish to minimise the objective function, which, in terms of the non-basic variables, is:

$$\begin{aligned} Z = 40x_1 + 100x_2 &= 40\left(24 + \frac{1}{10}s_3\right) + 100\left(7 + \frac{1}{8}s_2 - \frac{1}{160}s_3\right) \\ &= 1660 + \frac{25}{2}s_2 + \frac{27}{8}s_3 \end{aligned}$$

- Equivalently, we want to

$$\textbf{Max} \quad -Z = -1660 - \frac{25}{2}s_2 - \frac{27}{8}s_3.$$

Obtaining an initial vertex by inspection (cont.)

$$Z = 1660 + \frac{25}{2}s_2 + \frac{27}{8}s_3.$$

- We could use this information to construct an initial tableau.
- In fact this is unnecessary, since in the expression for Z, the coefficients for s_2 and s_3 are positive.

  $\Rightarrow$ Can't reduce Z further by making either s_2 or s_3 positive.

  i.e. the fbs found by inspection is in fact optimal.

The Big *M* Method

- In other problems it can be much harder to spot an initial vertex.
- A more systematic approach is to introduce new non-negative variables, called **artificial variables**, into equations without slack variables.
- We also insert each artificial variable into the objective function Z, giving it a large positive coefficient M in a minimisation problem (or a large negative coefficient -M in a maximisation one).
- If we then solve this modified problem, the algorithm will seek to make the artificial variables zero in order to avoid huge penalties.

The Big *M* Method (cont.)

- If this is achieved in the optimal solution of the modified problem, the optimal values of the basic variables will then provide an optimal fbs to the original problem.
- If an artificial variable remains positive, there is no feasible solution to the original problem.

Example D

$$
\begin{array}{lrcl}
\textbf{Maximise} & Z & = & x_1 + 5x_2 \\
\text{subject to} & x_1 & \leq & 7 \\
 & x_2 & \leq & 10 \\
 & x_1 + x_2 & \geq & 5 \\
 & x_1 \geq 0,\ x_2 & \geq & 0.
\end{array}
$$

The Big *M* Method (cont.)

- Although the origin is not in the FR, it is again easy here to find an initial vertex by inspection.
- We shall, however, use this example to show how the Big *M* method can be applied.
- We convert each constraint to an equality by introducing a non-negative slack or surplus variable, as appropriate, and also insert an artificial variable r_1 in the equation with the surplus variable:

$$\left.\begin{array}{rcrcrcrcrcrcr} x_1 & & & + & s_1 & & & & & & & = & 7 \\ & & x_2 & & & + & s_2 & & & & & = & 10 \\ x_1 & + & x_2 & & & & & - & s_3 & + & r_1 & = & 5 \end{array}\right\} \tag{16}$$

The Big M Method (cont.)

- Observe that the origin $(x_1, x_2) = (0, 0)$ does give a vertex of the FR in the modified problem, as we can take the basic variables to be s_1, s_2 and r_1.
- In this modified problem, we take the objective function to be

$$Z = x_1 + 5x_2 - M\,r_1.$$

- Expressing Z in terms of the non-basic variables gives

$$\begin{aligned} Z &= x_1 + 5x_2 - M(5 - x_1 - x_2 + s_3) = \\ &= (M+1)x_1 + (M+5)x_2 - M\,s_3 - 5M. \end{aligned}$$

- Note that the constraints are already in the required form (see box on slide 46)

The Big *M* Method (cont.)

Initial simplex tableau:

Check	Basic	x_1	x_2	s_1	s_2	s_3	r_1	RHS	Ratio
9	s_1	1	0	1	0	0	0	7	
12	s_2	0	1	0	1	0	0	10	
7	r_1	1	1	0	0	-1	1	5	
-6M-6	Z	-M-1	-M-5	0	0	M	0	-5M	

See Tutorial 2 for the derivation of the optimal solution using the simplex algorithm.

Note: If, in any tableau, an artificial variable becomes non-basic, the column corresponding to that variable can be deleted (modifying the check column suitably), as there will never be a case for giving it a non-zero value again.

The Two-Phase method

- For computational purposes, the Big M method has the drawback that the original coefficients of the decision variables are **dwarfed** by the big M, and so are liable to be lost in rounding errors.
- An alternative approach is to tackle the problem in two phases.
- In *Phase I*, take the constraints in the same form as in the Big M method. If these constraints contain the artificial variables $r_1, r_2, \ldots, r_k$, take the objective function to be

$$Z_2 = \begin{cases} r_1 + r_2 + \ldots + r_k & \text{in a minimisation problem;} \\ -r_1 - r_2 - \ldots - r_k & \text{in a maximisation problem.} \end{cases}$$

The Two-Phase method (cont.)

$$Z_2 = \begin{cases} r_1 + r_2 + \ldots + r_k & \text{in a minimisation problem;} \\ -r_1 - r_2 - \ldots - r_k & \text{in a maximisation problem.} \end{cases}$$

- In solving this problem, the simplex algorithm will attempt to make each of these artificial variables equal to zero.
- If it succeeds, it will in doing so find a fbs to the original problem. If it does not succeed, there is no feasible solution to the original problem.
- Artificial variables may again be discarded when they become non-basic.
- In *Phase II*, use the optimal fbs to the modified problem in Phase I as an initial fbs for the original problem. Apply the simplex algorithm in the usual way.

The Two-Phase method (cont.)

Example D

- Here we seek to

$$\textbf{Maximise} \quad Z = x_1 + 5x_2$$

- First, however, in Phase I we seek to

$$\textbf{Maximise} \quad Z_2 = -r_1$$

subject to constraints (16).

The Two-Phase method (cont.)

- We set up an initial tableau for the simplex algorithm in the usual way, except that, in addition to the row for the new objective function Z_2 at the bottom, we also include a row for Z at the top.
- Note that, as in the Big M method, we can take the initial basic variables to be s_1, s_2 and r_1.
- It is easy to express Z_2 in terms of the non-basic variables, since the third of the constraint equations (16) expresses the artificial variable r_1 in such terms. We obtain

$$Z_2 = -r_1 = x_1 + x_2 - s_3 - 5.$$

The Two-Phase method (cont.)

The initial tableau reads:

Check	Basic	x_1	x_2	s_1	s_2	s_3	r_1	RHS	Ratio	
-6	Z	-1	-5	0	0	0	0	0		
9	s_1	1	0	1	0	0	0	7	7	
12	s_2	0	1	0	1	0	0	10	∞	
7	r_1	①	1	0	0	-1	1	5	5	←
-6	Z_2	-1	-1	0	0	1	0	-5		
		↑								

We apply the simplex algorithm in the usual way to the constraint and Z_2 rows, but also, at each iteration, add an appropriate multiple of the pivot row to the Z row, in order to achieve a zero in the pivot column entry for that row.

The Two-Phase method (cont.)

Result:

Check	Basic	x_1	x_2	s_1	s_2	s_3	r_1	RHS	Ratio
1	Z	0	-4	0	0	-1	1	5	
2	s_1	0	-1	1	0	1	-1	2	
12	s_2	0	1	0	1	0	0	10	
7	x_1	1	1	0	0	-1	1	5	
1	Z_2	0	0	0	0	0	1	0	

This is an optimal solution to the Phase I problem, in which we note that r_1 is zero. So this solution effectively provides a fbs for the original problem.

The Two-Phase method (cont.)

- If we had not included the Z row, we would have said that we needed to express Z in terms of the non-basic variables in this fbs, which are x_2, s_3 and r_1.

- The third constraint row of the last tableau gives

$$x_1 = 5 - x_2 + s_3 - r_1$$

- So, an expression for Z in terms of the non-basic variables is

$$Z = (5 - x_2 + s_3 - r_1) + 5x_2 = 5 + 4x_2 + s_3 - r_1$$

- In fact, we have avoided the need to do this, as this is precisely the content of the Z row.

The Two-Phase method (cont.)

Discarding the artificial variable and revising the check column, the initial tableau for Phase II is:

Check	Basic	x_1	x_2	s_1	s_2	s_3	RHS	Ratio
3	s_1	0	-1	1	0	1	2	
12	s_2	0	1	0	1	0	10	
6	x_1	1	1	0	0	-1	5	
0	Z	0	-4	0	0	-1	5	

This tableau can be used to derive the optimal solution to the original problem. See Tutorial 2.

1. Linear Programs (LPs)
   - Introduction
   - Graphical solution
   - Slack and surplus variables

2. The Simplex algorithm
   - Basic and non-basic variables
   - Setting up an initial tableau
   - Applying the simplex algorithm
   - Finding an initial vertex
   - Special cases
   - Sensitivity Analysis
   - Matrix representation of the simplex procedure

Alternative optima

- We noted on slide 57 that if, at the point where an optimal solution has been found, a non-basic variable has a zero coefficient in the OF row, that variable can become positive without altering the value of Z, indicating the existence of another vertex of the FR which has the same optimal value of Z.
- As we have seen, this situation arises when the optimal vertices are on an edge or a face of the FR parallel to the OF and the number of optimal solutions is in fact infinite.

No feasible solution

- By definition, there is always a feasible solution in problems in which the origin serves as an initial vertex.

- So the problem of the non-existence of a feasible solution is one that is only liable to arise in situations in which we are resorting to the use of artificial variables.

- As we have seen, if an artificial variable takes a positive value in the optimal solution under the Big M method or after Phase I of the Two-Phase method, this indicates that there is no feasible solution.

Degeneracy

- Normally in a LP with n decision variables, a vertex occurs at the intersection of n constraints.
- If more than n constraints pass through a vertex it is called **degenerate** or over-determined.
- Hence one (or more) constraints is redundant.

Example A (Canvassing)

In this example, we had 2 decision variables, so usually 2 constraints determine a vertex. The origin, however, was degenerate as it lay on the constraints $2x_1 - x_2 \le 0, \quad x_1 \ge 0 \quad \text{and} \quad x_2 \ge 0$. So the redundant constraint here is $x_2 \ge 0$. We saw on slide 35 that one variable was zero on each constraint line. So at the origin we have not only $n = 2$ non-basic variables being zero but also a basic variable taking the value zero. (See the RHS column in the first two tableaux in the solution to Tutorial 1, question 3.)

Degeneracy (cont.)

- In general, degeneracy can be recognised in the simplex tableaux by a basic variable taking the value zero.
- Degeneracy can cause the phenomenon of **cycling** where the simplex iterations get into a repeated loop in which the objective function does not increase.
- In other cases, as in the canvassing example, Z may remain unchanged for an iteration, but increase thereafter when the simplex procedure leads us to another vertex.
- In fact, it is extremely rare for degeneracy to cause cycling to occur.
- Modified forms of the simplex algorithm which avoid the problem have nonetheless been developed.

Unbounded solution

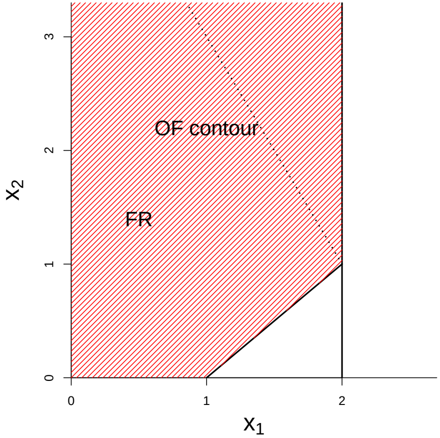
- If, in a maximisation problem, all entries in the column of a simplex tableau corresponding to a particular decision variable are non-positive, there is no finite optimal solution.

- **Example E**

  Consider the simplex tableau:

Basic	x_1	x_2	s_1	s_2	RHS
s_1	1	-1	1	0	1
s_2	1	0	0	1	2
Z	-2	-1	0	0	0

Unbounded solution (cont.)



Unbounded solution (cont.)

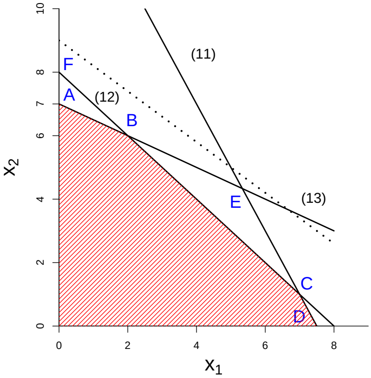
- At the origin, the variable x_2 is non-basic.
- As its entries in the constraint rows are non-positive, there is no danger of infringing a constraint by increasing it - however great the increase may be.
- Moreover, as x_2 has a negative entry in the OF row, clearly Z will increase indefinitely as x_2 is increased.
- Thus in this case the FR is unbounded and the value of the OF is not bounded above.
- In other cases, despite the FR being unbounded, an optimal value of the OF may nonetheless exist.
- e.g. If in example E, the OF is changed to $Z = 2x_1 - x_2$, the LP has a finite optimal solution at $(x_1, x_2) = (2,1)$.
- In practice, finding that a LP is unbounded usually indicates that the model is mis-specified e.g. by omission of a constraint.

Definition 4 (Binding constraints, scarce/abundant resources)

A constraint is **binding** if it passes through the optimal solution. When a constraint corresponds to a resource, the resource is said to be $\left\{ \begin{array}{c} \textbf{scarce} \\ \textbf{abundant} \end{array} \right\}$ if its constraint is $\left\{ \begin{array}{c} \text{binding} \\ \text{non}-\text{binding} \end{array} \right\}$. If the optimal solution is adopted, a scarce resource is thus completely used.

- In some problems, reducing the stock of an abundant resource may cut costs
- Relaxing a binding constraint can also be advantegeous
- We consider relaxing binding constraints one at a time

Example C



Since the optimal solution is vertex B, we see from the diagram that the binding constraints are (12) and (13), whilst constraint (11) is non-binding.

The contour of the objective function given by

$$4x_1 + 5x_2 = 45$$

is shown as a broken line.

It is evident that the value of Z could be increased by relaxing the binding constraints i.e. moving them away from the origin whilst retaining their gradients.

- If we do this with the edge BC, there is clearly nothing to be gained by going past point E, since, once past E, this constraint would be redundant.
- Note that $E = \left(\frac{16}{3}, \frac{13}{3}\right)$
- Thus the RHS of constraint (12) on slide 34, which originally equalled 8, has been increased at E to $29/3$.
- The value of the objective function Z at E is

$$4\left(\frac{16}{3}\right) + 5\left(\frac{13}{3}\right) = \frac{129}{3} = 43.$$

- Thus an increase of 5 in the value of Z has been achieved by relaxing the RHS by $5/3$.

- Hence relaxing constraint (12) by a unit amount increases Z by $5 \cdot \frac{3}{5} = 3$.
- Similarly if we relax constraint (13), whilst retaining its gradient, this constraint would be redundant if we went past point $F = (0, 8)$. At this point the RHS of constraint (13) which originally equalled 14, has been increased to 16.
- The value of the objective function Z at F is 40.
- Thus an increase of 2 in the value of Z has been achieved by relaxing the RHS by 2.
- Hence relaxing constraint (13) by a unit amount increases Z by 1.

Shadow price

- The increase in the value of the OF produced by relaxing a constraint by one unit is called the **shadow price**.
- When a constraint represents a resource, the shadow price gives the unit worth of the resource in terms of its benefit to the OF, suggesting that the constraint corresponding to the resource with the highest shadow price is the most important one to try to relax.
- The shadow price does not, however, take into account the cost of acquiring the resource.

Example C

- It is also of interest to examine the ways in which the OF rows of the second and third tableaux in §2.3 (slides 54 and 55) were constructed.

- Let $R_i^{(n)}$ denote the i^{th} row of the n^{th} tableau. So $R_4^{(2)}$ is the OF row of the second tableau.

- Noting the ways in which the relevant rows were created, we have

$$\begin{aligned} R_4^{(2)} &= R_4^{(1)} + 5R_3^{(2)} \\ &= R_4^{(1)} + 5\left(0.5R_3^{(1)}\right) \\ &= R_4^{(1)} + (0)R_1^{(1)} + (0)R_2^{(1)} + 2.5R_3^{(1)} \end{aligned}$$

Simplex multipliers

Thus

$$R_4^{(2)} = R_4^{(1)} + (0)R_1^{(1)} + (0)R_2^{(1)} + 2.5R_3^{(1)}$$

- Observe that the coefficients of $R_1^{(1)}$, $R_2^{(1)}$ and $R_3^{(1)}$ appear as the elements of the OF row of the second tableau corresponding to the initial basic variables s_1, s_2 and s_3 (slide 54).
- So these elements give the multiples of the rows of the *initial* tableau that have been added to the OF row.
- These elements are therefore called **simplex multipliers.**

- More generally, under the simplex algorithm, any row $R_i^{(n)}$ $(n > 1)$ is produced by adding to $R_i^{(n-1)}$ some multiple (positive or negative) of a constraint row of the $n - 1$th tableau.

- By induction, $R_i^{(n)}$ equals $R_i^{(1)}$ plus some linear sum of the constraint rows of the initial tableau.

- Moreover, in the initial tableau, the elements of the OF row in the initial basic variable columns are all zero.

Example C

- In the third tableau, the elements of the OF row in the columns corresponding to the initial basic variables s_1, s_2 and s_3 are 0, 3 and 1.
- The vector (0, 3, 1) must therefore be a linear sum of the row vectors (1, 0, 0), (0, 1, 0) and (0, 0, 1) that occupied these columns in the constraint rows of the initial tableau.
- We decide that

$$R_4^{(3)} = R_4^{(1)} + (0)R_1^{(1)} + (3)R_2^{(1)} + (1)R_3^{(1)}$$

- In general, when applying the simplex algorithm, it is clear that, for any iteration, the elements of the OF row corresponding to the initial basic variables may be viewed as simplex multipliers.
- We calculated above the shadow prices associated with constraints (12) and (13).
- The slack variables associated with these constraints are s_2 and s_3.
- Observe that the simplex multipliers appearing in the final tableau in the s_2 and s_3 columns are simply the shadow prices.

Initial tableau in matrix form

- Let **A** be an $m \times q$ matrix, **b** an $m \times 1$ vector and **I** an $m \times m$ identity matrix.
- If we take the constraints in the form required for the initial tableau of the simplex algorithm, these equations can be written as

$$[\mathbf{A},\ \mathbf{I}]\,\mathbf{x} \;=\; \mathbf{b}, \qquad (\mathbf{x} \;\geq\; \mathbf{0}) \tag{17}$$

where **x** is defined to be a vector consisting of *all* the variables (decision, slack, surplus or artificial) *in some appropriate order*.

Example D

$$\begin{array}{ccccccccccccc} x_1 & & & + & s_1 & & & & & & & = & 7 \\ & & x_2 & & & + & s_2 & & & & & = & 10 \\ x_1 & + & x_2 & & & & & - & s_3 & + & r_1 & = & 5 \end{array}$$

If we set $\mathbf{x} = (x_1, x_2, s_3, s_1, s_2, r_1)^T$, the above constraints can be expressed in the form (17):

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} x_1 \\ x_2 \\ s_3 \\ s_1 \\ s_2 \\ r_1 \end{pmatrix} = \begin{pmatrix} 7 \\ 10 \\ 5 \end{pmatrix}$$

Basic variables

- Now let $\mathbf{x}_B$ be any $m \times 1$ vector of basic variables, and $\mathbf{x}_N$ a $q \times 1$ vector of the corresponding non-basic variables.
- If, by selecting appropriate columns of $[\mathbf{A},\ \mathbf{I}]$, we create a new $m \times m$ matrix $\mathbf{B}$ and a new $m \times q$ matrix $\mathbf{N}$, we may write equation (17) as

$$\mathbf{B}\,\mathbf{x}_B + \mathbf{N}\,\mathbf{x}_N = \mathbf{b} \qquad \text{or} \qquad [\mathbf{B},\ \mathbf{N}] \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = \mathbf{b}. \qquad (18)$$

- It can be shown that the columns of $\mathbf{B}$ are linearly independent, and hence that $\mathbf{B}$ is of rank m and is invertible.

Basic variables (cont.)

- Multiplying through (18) by the inverse matrix $\mathbf{B}^{-1}$ gives

$$\mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\,\mathbf{x}_N = \mathbf{B}^{-1}\,\mathbf{b} \tag{19}$$

$$[\mathbf{I},\ \mathbf{B}^{-1}\mathbf{N}] \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = \mathbf{B}^{-1}\,\mathbf{b} \tag{20}$$

- Equation (19) gives an

Expression for the basic variables in terms of the non-basic ones

$$\mathbf{x}_B = \mathbf{B}^{-1}\,\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\,\mathbf{x}_N \tag{21}$$

Example C

Taking the constraints given on sl. 34, we may write them in the form (17) as follows:

$$\left(\begin{array}{cc|ccc} 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{array}\right)\begin{pmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} 15 \\ 8 \\ 14 \end{pmatrix}$$

If we let $\mathbf{x}_B = (s_1,\ x_1,\ x_2)^T$ and $\mathbf{x}_N = (s_2,\ s_3)^T$, then $\mathbf{x}_B$ contains the basic variables for the final tableau on sl 55. So here we have

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \Rightarrow \mathbf{B}^{-1} = \begin{pmatrix} 1 & -3 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \text{ and } \mathbf{N} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Thus $[\mathbf{B},\ \mathbf{N}]$ is obtained from $[\mathbf{A},\ \mathbf{I}]$ by permutation of columns.
- Since, at the vertex where $\mathbf{x}_B$ provides the basic variables, $\mathbf{x}_N = \mathbf{0}$, it follows from (21) that

$$\mathbf{x}_B = \mathbf{B}^{-1}\,\mathbf{b} = \begin{pmatrix} 1 & -3 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 15 \\ 8 \\ 14 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 6 \end{pmatrix}$$

- So, in this case, equation (20)

$$[\mathbf{I}, \quad \mathbf{B}^{-1}\mathbf{N}] \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = \mathbf{B}^{-1}\,\mathbf{b}$$

reads

$$\left(\begin{array}{ccc|cc} 1 & 0 & 0 & -3 & 1 \\ 0 & 1 & 0 & 2 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{array}\right) \left(\begin{array}{c} s_1 \\ x_1 \\ x_2 \\ \hline s_2 \\ s_3 \end{array}\right) = \begin{pmatrix} 5 \\ 2 \\ 6 \end{pmatrix}$$

- Observe that these equations are the constraint equations of the final tableau given on slide 55.

- The objective function Z is a linear function of the decision variables.

- Suppose that, at the initial vertex, the vectors of basic and non-basic variables are $\mathbf{x}_\beta$ and $\mathbf{x}_\alpha$ respectively.

- We may write Z either in terms of $\mathbf{x}_\alpha$ or (by just rearranging the order of the terms) in terms of the basic and non-basic variables, $\mathbf{x}_B$ and $\mathbf{x}_N$ respectively, at some other vertex.

- We obtain

$$Z \ = \ \mathbf{c}_\alpha^T \, \mathbf{x}_\alpha \ + \ c_0 \ = \ \mathbf{c}_B^T \, \mathbf{x}_B \ + \ \mathbf{c}_N^T \, \mathbf{x}_N \ + \ c_0,$$

  where c_0 is a constant, $\mathbf{c}_B$ is an $m \times 1$ vector, and $\mathbf{c}_\alpha$ and $\mathbf{c}_N$ are $q \times 1$ vectors.

- Substituting for $\mathbf{x}_B$ using (21) gives

Z as a function only of $\mathbf{x}_N$

$$\begin{aligned} Z &= \mathbf{c}_B^T(\mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{N}\,\mathbf{x}_N) + \mathbf{c}_N^T\mathbf{x}_N + c_0 \\ &= (\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N})\mathbf{x}_N + \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} + c_0 \end{aligned} \tag{22}$$

- At the vertex where $\mathbf{x}_B$ gives the basic variables, we have $\mathbf{x}_N = \mathbf{0}$, and hence from (22) the value of Z is

$$\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} + c_0.$$

- This value of Z is the entry in the RHS column of the objective function row.

Example C

- Here we took the origin as the initial vertex, and we had $\mathbf{x}_\beta = (s_1,\ s_2,\ s_3)^T$ and $\mathbf{x}_\alpha = (x_1,\ x_2)^T$, whereas at the optimal vertex we have $\mathbf{x}_B = (s_1,\ x_1,\ x_2)^T$ and $\mathbf{x}_N = (s_2,\ s_3)^T$.

Example C (cont.)

- The expression at the origin for the objective function was

$$Z = \begin{pmatrix} 4 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 4 & 5 \end{pmatrix} \begin{pmatrix} s_1 \\ x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \end{pmatrix} \begin{pmatrix} s_2 \\ s_3 \end{pmatrix}$$

$$\text{i.e.} \quad c_0 = 0, \quad \mathbf{c}_B^T = \begin{pmatrix} 0 & 4 & 5 \end{pmatrix}$$

- Using the matrix $\mathbf{B}^{-1}\,\mathbf{b}$ obtained above (slide 102), we can confirm that, at the optimal vertex,

$$Z = \mathbf{c}_B^T\,\mathbf{B}^{-1}\,\mathbf{b} = \begin{pmatrix} 0 & 4 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ 6 \end{pmatrix} = 38$$

- In fact, the updating of the complete simplex tableau can be presented in matrix terms.
- Incorporating equation (17) in a larger matrix equation and letting $\mathbf{0}$ be the $m \times 1$ vector $(0,\ 0,\ \dots\ ,\ 0)^T$, we can write the corresponding tableau as

$$\begin{pmatrix} \mathbf{A} & \mathbf{I} & \mathbf{0} \\ -\mathbf{c}_\alpha^T & \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_\alpha \\ \mathbf{x}_\beta \\ Z \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ c_0 \end{pmatrix} \tag{23}$$

- Pre-multiplying this equation by $\begin{pmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{c}_B^T\,\mathbf{B}^{-1} & 1 \end{pmatrix}$ yields

Matrix representation of a simplex tableau

$$\begin{pmatrix} \mathbf{B}^{-1}\mathbf{A} & \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{A} - \mathbf{c}_\alpha^T & \mathbf{c}_B^T\,\mathbf{B}^{-1} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_\alpha \\ \mathbf{x}_\beta \\ Z \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} + c_0 \end{pmatrix} \tag{24}$$

- Observe that the elements of the vector $\mathbf{c}_B^T\,\mathbf{B}^{-1}$ are the simplex multipliers.

Example C

- At the initial vertex, we had $\mathbf{x}_\beta = (s_1,\ s_2,\ s_3)^T$ and $\mathbf{x}_\alpha = (x_1,\ x_2)^T$.
- The initial tableau can be written in the form (23) as:

$$\left(\begin{array}{cc|ccc|c} 2 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ \hline -4 & -5 & 0 & 0 & 0 & 1 \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ \hline s_1 \\ s_2 \\ s_3 \\ \hline Z \end{array}\right) = \left(\begin{array}{c} 15 \\ 8 \\ 14 \\ 0 \end{array}\right) \qquad (25)$$

- As we saw on slide 54, after one iteration of the simplex algorithm, the basic variables at vertex A were s_1, s_2 and x_2.
- So set $\mathbf{X_B} = (s_1, s_2, x_2)^T$

- For the first iteration, moving from the origin to vertex A, the relevant matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \quad \Rightarrow \quad \mathbf{B}^{-1} = \frac{1}{2}\begin{pmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

- At the origin, $Z = 4x_1 + 5x_2$.
- Hence $\mathbf{c}_B^T = \begin{pmatrix} 0 & 0 & 5 \end{pmatrix}$ and so $\mathbf{c}_B^T B^{-1} = \begin{pmatrix} 0 & 0 & 2.5 \end{pmatrix}$

$$\Rightarrow \quad \begin{pmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{c}_B^T\mathbf{B}^{-1} & 1 \end{pmatrix} = \left(\begin{array}{ccc|c} 1 & 0 & -0.5 & 0 \\ 0 & 1 & -0.5 & 0 \\ 0 & 0 & 0.5 & 0 \\ \hline 0 & 0 & 2.5 & 1 \end{array}\right).$$

- Pre-multiplying equation (25) by this matrix yields

$$\left(\begin{array}{cc|ccc|c} 1.5 & 0 & 1 & 0 & -0.5 & 0 \\ 0.5 & 0 & 0 & 1 & -0.5 & 0 \\ 0.5 & 1 & 0 & 0 & 0.5 & 0 \\ \hline -1.5 & 0 & 0 & 0 & 2.5 & 1 \end{array}\right) \left(\begin{array}{c} x_1 \\ x_2 \\ \hline s_1 \\ s_2 \\ s_3 \\ \hline Z \end{array}\right) = \left(\begin{array}{c} 8 \\ 1 \\ 7 \\ 35 \end{array}\right)$$

- The correspondence with the tableau for vertex *A* is clear.

- We now consider in more detail the effect of simplex iterations on the OF row, which is shown in (24) partitioned according to the initial basic variables.

- Suppose that, ordering the variables as in (17), the form of the OF for the initial tableau is

$$Z = c_0 + \sum_{j=1}^{m+q} c_j x_j.$$

- Thus, for a maximisation problem, the entry in the OF row in the x_j column of the initial tableau will be $-c_j$.

- Now let $\mathbf{P}_j$ $(j = 1,\ 2,\ \dots\ ,\ m+q)$ denote the vector of entries in the constraint rows of the x_j column of **the initial tableau**.

- As the elements of the vector $\mathbf{c}_B^T\,\mathbf{B}^{-1}$ are the simplex multipliers, it follows that the entry in the OF row for the x_j column of the resulting tableau will be $\mathbf{c}_B^T B^{-1}\mathbf{P}_j - c_j$.

- Clearly, by the rules for manipulating tableaux, if x_j is a basic variable, this entry must be zero.

- This can also be confirmed algebraically.

- Suppose $\mathbf{P}_j$ is the k^{th} column of $\mathbf{B}$. Let $\mathbf{e}_k^T$ be a 1 x m row vector with the k^{th} element equal to 1 and the rest equal to zero. Then $\mathbf{B}^{-1}\mathbf{P}_j = \mathbf{e}_k$ by definition of the inverse matrix.

- Hence $\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{P}_j = \mathbf{c}_B^T \mathbf{e}_k = k^{\text{th}}$ element of $\mathbf{c}_B^T = c_j$ (as the k^{th} column of $\mathbf{B}$ is the x_j column of the tableau.) i.e.

$$\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{P}_j - c_j = 0$$

NB

In a maximisation problem, in terms of the matrix $\mathbf{B}$ corresponding to the vector $\mathbf{x}_B$ of basic variables, the entry for x_j in the OF row is $\mathbf{c}_B^T \mathbf{B}^{-1} \mathbf{P}_j - c_j$, which is zero if x_j is a basic variable.

References (for Chapter 2)

	Taha O.R. (9th ed.)	Winston O.R. (4th ed.)
Simplex tableaux	3.3.2	4.5
Big M	3.4.1	4.12
Two-Phase	3.4.2	4.13
Special cases	3.5	4.7,4.8,4.11
Sensitivity	3.6.1 (2 dim.), (4.5)	5.1 (2 dim.), (6.3)
Matrix representation	4.2.2, 4.2.4, 7.1.2	6.2

Note

These texts contain much useful material, but beware differences from the lectures, including:

(i) Both authors routinely put the objective function row at the top of their tableaux.

(ii) Definitions of the standard form differ in these two books, and both differ from the lectures.

(iii) Taha shows that, rather than converting minimisation problems to maximisation ones, the entering variable may be chosen as the one having the most *positive* entry in the objective function row. Winston discusses both approaches.

Outline

3. Duality
   - Definition of the dual problem
   - Duals of non-canonical problems
   - Relationships between the primal and the dual
   - Complementary slackness

4. Integer Programming
   - Introduction
   - The Branch-and-Bound (B&B) algorithm

5. Solving a LP using a spreadsheet

3 Duality
- Definition of the dual problem
- Duals of non-canonical problems
- Relationships between the primal and the dual
- Complementary slackness

4 Integer Programming
- Introduction
- The Branch-and-Bound (B&B) algorithm

5 Solving a LP using a spreadsheet

- Every original or **primal** LP problem has associated with it a companion or **dual** problem.
- In some cases, the dual problem is easier to solve than the primal one, particularly if the job is being done manually.
- Investigation of duality offers further insight into the primal LP.
- We will also see below that the standard algorithm for solving transportation problems is derived via the dual of the LP for the transportation problem.

Definition 5 (Canonical form of a LP)

A LP is in **canonical form** if all variables are non-negative and all the constraints are inequalities of

$$\begin{cases} \leq \text{ type when the objective is maximisation;} \\ \geq \text{ type when the objective is minimisation.} \end{cases}$$

- So the LP (7) on slide 12 was a maximisation problem in canonical form.
- It was specified in terms of $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, $\mathbf{b} = (b_1, b_2, \ldots, b_m)^T$, $\mathbf{c} = (c_1, c_2, \ldots, c_n)^T$, and an $m \times n$ matrix $\mathbf{A}$.

Definition 6 (Dual LP)

The **dual LP** corresponding to the primal problem

$$\begin{array}{ll} \textbf{Maximise} & Z = \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge \mathbf{0}. \end{array}$$

is given by

$$\begin{array}{ll} \textbf{Minimise} & W = \mathbf{b}^T \mathbf{y} \\ \text{subject to} & \mathbf{A}^T\mathbf{y} \ge \mathbf{c} \text{ and } \mathbf{y} \ge \mathbf{0}. \end{array}$$

So here $\mathbf{y}$ is a $m \times 1$ column vector and $\mathbf{A}^T$ is a $n \times m$ matrix. Thus

Number of **decision variables** in the dual =
Number of **constraints** in the primal.

Number of **constraints** in the dual =
Number of **decision variables** in the primal.

Example C

$$\begin{array}{lrcl} \textbf{Maximise} & Z = 4x_1 + 5x_2 & & \\ \text{subject to} & 2x_1 + x_2 & \leq & 15 \\ & x_1 + x_2 & \leq & 8 \\ & x_1 + 2x_2 & \leq & 14 \\ & x_1 \geq 0, \; x_2 & \geq & 0 \end{array}$$

Example C

For the LP in **Example C** we have

$$\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} 15 \\ 8 \\ 14 \end{pmatrix} \text{ and } \mathbf{c} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}.$$

Hence the corresponding dual LP is

$$\textbf{Minimise} \quad W = 15y_1 + 8y_2 + 14y_3$$

$$\text{subject to} \quad \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \geq \begin{pmatrix} 4 \\ 5 \end{pmatrix} \quad \left(\text{i.e.} \quad \begin{array}{rcrcrcl} 2y_1 & + & y_2 & + & y_3 & \geq & 4 \\ y_1 & + & y_2 & + & 2y_3 & \geq & 5 \end{array} \right)$$

and $\mathbf{y} \geq \mathbf{0}$.

Dual of the dual

Theorem 1

The dual of the dual is the primal.

Proof

The dual LP **Minimise** $\mathbf{b}^T\,\mathbf{y}$

subject to $\mathbf{A}^T\mathbf{y} \geq \mathbf{c}$ and $\mathbf{y} \geq \mathbf{0}$

can be written as a maximisation problem in canonical form as

Maximise $-\mathbf{b}^T\,\mathbf{y}$

subject to $-\mathbf{A}^T\mathbf{y} \leq -\mathbf{c}$ and $\mathbf{y} \geq \mathbf{0}$

Proof of **Theorem 1** (cont.)

If we denote the vector of decision variables of the dual of the dual by **x**, we may write the dual of the dual as

$$\textbf{Minimise} \quad -\mathbf{c}^T\,\mathbf{x}$$

$$\text{subject to} \quad (-\mathbf{A}^T)^T\mathbf{x} \geq -\mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}$$

This is equivalent to

$$\textbf{Maximise} \quad \mathbf{c}^T\,\mathbf{x}$$

$$\text{subject to} \quad (\mathbf{A}^T)^T\mathbf{x} \leq \mathbf{b} \quad \text{and} \quad \mathbf{x} \geq \mathbf{0}$$

Noting that $(\mathbf{A}^T)^T = \mathbf{A}$, we see that this is simply the primal.

Thus duality is a mutually reciprocal property: either problem can be regarded as the primal one.

Non-canonical LP

For future use, we look next at the duals of some non-canonical LPs. Consider the LP

Example F

$$\begin{array}{llrcl} \textbf{Maximise} & Z = x_1 + x_2 & & & \\ \textbf{subject to} & & x_1 + 2x_2 & \leq & 5 \\ & & 3x_1 + x_2 & \geq & 4 \\ & & x_1 \geq 0, \; x_2 & \geq & 0. \end{array} \tag{26}$$

In canonical form the constraints read

$$\begin{array}{rcl} x_1 + 2x_2 & \leq & 5, \\ -3x_1 - x_2 & \leq & -4 \end{array}$$

Dual of the LP in **Example F**

$$\begin{array}{lrcl} \textbf{Minimise} & W & = & 5y_1 - 4y_2 \\ \text{subject to} & y_1 - 3y_2 & \geq & 1, \\ & 2y_1 - y_2 & \geq & 1 \\ & y_1 \geq 0,\ y_2 & \geq & 0 \end{array}$$

Putting $u_2 = -y_2$, we can write the dual so that all the coefficients have the same sign as in the primal:

$$\begin{array}{lrcl} \textbf{Minimise} & W & = & 5y_1 + 4u_2 \\ \text{subject to} & y_1 + 3u_2 & \geq & 1, \\ & 2y_1 + u_2 & \geq & 1, \\ & y_1 \geq 0,\ u_2 & \leq & 0. \end{array}$$

Inequality deviating from canonical form

In general, if a primal constraint has an inequality deviating from canonical form (i.e. a $\geq$ constraint in a maximisation problem or a $\leq$ constraint in a minimisation problem), then the associated variable in the dual is restricted to be non-positive.

Example G

We now change constraint (26) in example F to an equality i.e.

$$3x_1 + x_2 = 4.$$

So the LP could now be written as:

$$\begin{array}{ll} \textbf{Maximise} & Z = x_1 + x_2 \\ \text{subject to} & x_1 + 2x_2 \leq 5, \\ & 3x_1 + x_2 \leq 4, \\ & 3x_1 + x_2 \geq 4, \\ & x_1 \geq 0, \; x_2 \geq 0. \end{array}$$

So the dual becomes:

$$\begin{array}{llrcrcrcl} \textbf{Minimise} & W = & 5y_1 & + & 4y_2 & + & 4y_3 & & \\ \text{subject to} & & y_1 & + & 3y_2 & + & 3y_3 & \geq & 1, \\ & & 2y_1 & + & y_2 & + & y_3 & \geq & 1, \\ & & \multicolumn{7}{r}{y_1 \geq 0,\ y_2 \geq 0,\ y_3 \leq 0.} \end{array}$$

Alternatively, if we set $u_2 = y_2 + y_3$, we can write the dual more simply as:

$$\begin{array}{llrcrcl} \textbf{Minimise} & W = & 5y_1 & + & 4u_2 & & \\ \text{subject to} & & y_1 & + & 3u_2 & \geq & 1, \\ & & 2y_1 & + & u_2 & \geq & 1, \\ & & \multicolumn{5}{r}{y_1 \geq 0,\ u_2 \text{ unrestricted in sign.}} \end{array}$$

In general, if a primal constraint is a strict equality, then the associated variable in the dual is unrestricted in sign.

Apart from this change, the usual rules for forming the dual continue to apply.

Notation

In a dual problem with decision variables $y_1,\ y_2,\ \ldots\ ,\ y_m$, we will denote surplus or slack variables by $t_1,\ t_2,\ \ldots\ ,\ t_n$.

Example C

To solve the dual LP, derived on slide 120, we turn the constraints into equalities using surplus variables:

$$\begin{array}{rcrcrrrcr} 2y_1 & + & y_2 & + & y_3 & -t_1 & & = & 4 \\ y_1 & + & y_2 & + & 2y_3 & & -t_2 & = & 5 \end{array}$$

We could then spot the initial vertex given by $y_2 = 5,\ t_1 = 1$, $y_1 = y_3 = t_2 = 0$. Next we express W and the constraints in the required form and write the LP as a maximisation problem.

Initial tableau

Check	Basic	y_1	y_2	y_3	t_1	t_2	RHS	Ratio
8	y_2	1	1	2	0	-1	5	2.5
1	t_1	-1	0	1	1	-1	1	1
-30	$-W$	7	0	-2	0	8	-40	

Second tableau

Check	Basic	y_1	y_2	y_3	t_1	t_2	RHS	Ratio
6	y_2	3	1	0	-2	1	3	
1	y_3	-1	0	1	1	-1	1	
-25	$-W$	5	0	0	2	6	-38	

The optimal solution is $y_1 = 0$, $y_2 = 3$ and $y_3 = 1$, yielding an optimal value of $-W = -38$, i.e. $W = 38$.

Similarities with the optimal tableau for the primal (see slide 55):

1. Optimal value for the dual = Optimal value for the primal
   i.e. Minimum value of W = Maximum value of Z.

2. Recall that the number of decision variables in the dual equalled the number of constraints in the primal. Moreover the variable y_i is associated with the i^{th} primal constraint (the coefficients of y_i in the dual come from the i^{th} primal constraint). We now see that:

   Optimal value of dual decision variable y_i = shadow price associated with i^{th} primal constraint ($i = 1, 2, 3$).

3. Similarly: Optimal value of primal variable x_j = shadow price associated with j^{th} dual constraint ($j = 1, 2$).

   Since the roles of the primal and the dual can be reversed, this is unsurprising.

Are these properties true in general?

Theorem 2 (Weak Duality Property)

If $\mathbf{x}_f$ *and* $\mathbf{y}_f$ *are feasible solutions to the primal and dual problems respectively, and we set* $Z_f = \mathbf{c}^T\mathbf{x}_f$ *and* $W_f = \mathbf{b}^T\mathbf{y}_f$*, then* $Z_f \leq W_f$.

Proof:

As $\mathbf{x}_f$ is feasible, we have $\mathbf{A}\mathbf{x}_f \leq \mathbf{b}$. Since $(MN)^T = N^T M^T$ for matrices M and N of appropriate dimensions, we have

$$\mathbf{x}_f^T \mathbf{A}^T = (\mathbf{A}\,\mathbf{x}_f)^T \leq \mathbf{b}^T \quad \Rightarrow \quad \mathbf{x}_f^T \mathbf{A}^T \mathbf{y}_f \leq \mathbf{b}^T\,\mathbf{y}_f \quad (\text{as } \mathbf{y}_f \geq \mathbf{0}) \qquad (27)$$

Similarly, as $\mathbf{y}_f$ is feasible, we have $\mathbf{A}^T\,\mathbf{y}_f \geq \mathbf{c}$

$$\Rightarrow \quad \mathbf{x}_f^T \mathbf{A}^T \mathbf{y}_f \geq \mathbf{x}_f^T\,\mathbf{c} = \mathbf{c}^T\,\mathbf{x}_f \quad (\text{as } \mathbf{x}_f \geq \mathbf{0} \text{ and } \mathbf{x}_f^T\,\mathbf{c} \text{ is scalar}) \qquad (28)$$

Hence, from (27) and (28), $\mathbf{c}^T\,\mathbf{x}_f \leq \mathbf{x}_f^T \mathbf{A}^T \mathbf{y}_f \leq \mathbf{b}^T\,\mathbf{y}_f \quad \Rightarrow \quad Z_f \leq W_f.$

Corollaries of Theorem 2

Theorem 2 shows that, at feasible points, the objective function for the dual is never less than the objective function for the primal.

Corollary 1

If $\mathbf{x}_f$ *and* $\mathbf{y}_f$ *are feasible solutions to the primal and dual problems respectively, such that* $\mathbf{c}^T\mathbf{x}_f = \mathbf{b}^T\mathbf{y}_f$, *then* $\mathbf{x}_f$ *and* $\mathbf{y}_f$ *are optimal solutions to their respective problems.*

Corollary 2

If there are feasible solutions to the primal problem, but the objective function is unbounded, there is no feasible solution to the dual problem.

(See Tutorial 3 for proofs of corollaries.)

Note

In proving the Duality Theorem below we assume the primal problem is in the canonical form given in section 3.1 and that the origin will serve as an initial vertex. In the notation of section 2.7, we then have

(a) the initial vector $\mathbf{x}_\beta$ of basic variables consists of slack variables;

(b) the initial vector $\mathbf{x}_\alpha$ of non-basic variables is a vector $\mathbf{x}$ of decision variables;

(c) the coefficients of the objective function Z are such that $\mathbf{c}_\alpha^T = \mathbf{c}^T$ and $c_0 = 0$.

To generalise to other cases, include any surplus variables in $\mathbf{x}$, and slack or artificial variables in $\mathbf{x}_\beta$.

Theorem 3 (Duality Theorem)

If $\mathbf{x}^*$ *is a feasible solution to the primal problem yielding a finite optimal value* Z^* *of the objective function* Z*, then the dual has a feasible solution* $\mathbf{y}^*$ *yielding the same value of the objective function* W*. Hence* $\mathbf{y}^*$ *is the optimal solution to the dual problem and the optimal value* W^* *of* W *satisfies* $Z^* = \mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^* = W^*$*.*

Proof

Firstly set $\mathbf{y}^* = (\mathbf{B}^{-1})^T\mathbf{c}_B = (\mathbf{c}_B^T\mathbf{B}^{-1})^T$, where $\mathbf{B}$ is the matrix associated with the basic variables giving the optimal solution $\mathbf{x}^*$. [Thus $\mathbf{y}^*$ is the vector of shadow prices (optimal simplex multipliers).]

Then, as $\mathbf{b}^T\mathbf{y}^*$ is a scalar, $W = \mathbf{b}^T\mathbf{y}^* = (\mathbf{y}^*)^T\mathbf{b} = \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} = Z^*$ by (24).

- So this expression for $\mathbf{y}^*$ gives a value of W which equals the optimal value Z^* of Z.
- But is $\mathbf{y}^*$ feasible?
- Noting the substitutions given before the theorem, and again defining $\mathbf{B}$ relative to the optimal basic variables, the matrix version (24) of the optimal tableau reads:

$$\begin{pmatrix} \mathbf{B}^{-1}\mathbf{A} & \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{A} - \mathbf{c}^T & \mathbf{c}_B^T\mathbf{B}^{-1} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{x}_\beta \\ Z \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b} \end{pmatrix}$$

- As we are using the matrix **B** which gives the optimal tableau, we have

$$\begin{pmatrix} \mathbf{c}_B^T \mathbf{B}^{-1}\mathbf{A} - \mathbf{c}^T & \mathbf{c}_B^T \mathbf{B}^{-1} \end{pmatrix} \geq \mathbf{0}^T \quad \Rightarrow \quad \begin{pmatrix} \mathbf{A}^T \left(\mathbf{c}_B^T \mathbf{B}^{-1}\right)^T - \mathbf{c} \\ \left(\mathbf{c}_B^T \mathbf{B}^{-1}\right)^T \end{pmatrix} \geq \mathbf{0}$$

$$\text{i.e.} \begin{pmatrix} \mathbf{A}^T \mathbf{y}^* - \mathbf{c} \\ \mathbf{y}^* \end{pmatrix} \geq \mathbf{0}, \text{ i.e. } \quad \mathbf{A}^T \mathbf{y}^* \geq \mathbf{c} \text{ and } \mathbf{y}^* \geq \mathbf{0}.$$

- Hence $\mathbf{y}^*$ is indeed feasible.
- Since $\mathbf{x}^*$ and $\mathbf{y}^*$ are feasible solutions to the primal and dual problems respectively, such that $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$, then, by Corollary 1 above, $\mathbf{y}^*$ is the optimal solution to the dual problem, and so

$$Z^* = \mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^* = W^*.$$

3 Duality

- Definition of the dual problem
- Duals of non-canonical problems
- Relationships between the primal and the dual
- Complementary slackness

4 Integer Programming

- Introduction
- The Branch-and-Bound (B&B) algorithm

5 Solving a LP using a spreadsheet

Suppose that we introduce slack variables into the canonical form of the primal problem given on slide 117, and surplus variables into the dual. We may write the constraints of these problems as

$$\begin{aligned} \mathbf{A}\,\mathbf{x} \;+\; \mathbf{I}_m\,\mathbf{s} &= \mathbf{b} \\ \mathbf{A}^T\,\mathbf{y} \;-\mathbf{I}_n\,\mathbf{t} &= \mathbf{c}, \end{aligned} \tag{29}$$

where $\mathbf{s} = (s_1,\ s_2,\ldots,\ s_m)^T, \mathbf{t} = (t_1,\ t_2,\ldots,\ t_n)^T$ and I_m is an $m \times m$ identity matrix.

Theorem 4 (Complementary slackness)

If $\boldsymbol{x}$ *and* $\boldsymbol{y}$ *are optimal solutions to the primal and dual problems respectively, then*

$$y_i\ s_i = 0 \qquad i = 1,\ 2,\ldots,\ m \tag{30}$$

and

$$x_j\,t_j = 0 \qquad j = 1,\ 2,\ldots,\ n \tag{31}$$

Proof of Theorem 4

Multiplying (29) by $\mathbf{y}^T$ gives the scalar equation

$$\mathbf{y}^T\,\mathbf{A}\,\mathbf{x} \;+\; \mathbf{y}^T\,\mathbf{s} \;=\; \mathbf{y}^T\,\mathbf{b} \quad \Rightarrow \quad \mathbf{x}^T\,\mathbf{A}^T\,\mathbf{y} \;+\; \mathbf{s}^T\,\mathbf{y} \;=\; \mathbf{b}^T\,\mathbf{y}$$

(by taking transposes)

If $\mathbf{x}$ and $\mathbf{y}$ are optimal, $\mathbf{b}^T\mathbf{y} \;=\; \mathbf{c}^T\mathbf{x}$, and therefore

$$\mathbf{c}^T\mathbf{x} \;=\mathbf{x}^T\,\mathbf{A}^T\,\mathbf{y} \;+\; \mathbf{s}^T\,\mathbf{y} \;\geq\; \mathbf{x}^T\mathbf{c} \;+\; \mathbf{s}^T\mathbf{y}$$

since $\mathbf{y}$ is feasible in the dual problem. But, since it is scalar, $\mathbf{c}^T\mathbf{x} = \mathbf{x}^T\mathbf{c}$, and hence

$$0 \;\geq\; \mathbf{s}^T\,\mathbf{y} \;=\; s_1\,y_1 \;+\; s_2\,y_2 \;+\; \ldots \;+ s_m\,y_m$$

Now $s_i \geq 0$ and $y_i \geq 0$, and therefore the result (30) follows. The proof of equations (31) is similar.

Equations (30) and (31) tell us that

(a) *either* $y_i = 0$ *or* $s_i = 0$ (i.e. the primal constraint associated with y_i is an equality) *or* both; and

(b) *either* $x_j = 0$ *or* $t_j = 0$ (i.e. the dual constraint associated with x_j is an equality) *or* both.

Thus, in each case, if we know that one of the pair of variables is positive, the other must be zero.

N.B. $y_1 = 0$ does NOT imply $s_1 > 0$. Similar comments apply to the other equations in (30) and (31).

Theorem 4 thus establishes the **principle of complementary slackness**, which says:

"At the optimal solutions to the primal and dual problems, either a variable is zero or the constraint associated with that variable is an equality at the optimal solution values."

Example C

The constraint equations of the primal (on slide 34) and the dual (on slide 120) were:

$$\begin{aligned} 2x_1 + x_2 + s_1 &= 15, \\ x_1 + x_2 + s_2 &= 8, \\ x_1 + 2x_2 + s_3 &= 14 \end{aligned} \qquad \text{and} \qquad \begin{aligned} 2y_1 + y_2 + y_3 - t_1 &= 4, \\ y_1 + y_2 + 2y_3 - t_2 &= 5 \end{aligned}$$

- From the plot of the feasible region of the primal in §2.1 we could have shown graphically that the optimal solution is given by $x_1 = 2, x_2 = 6$.
- It follows that $s_1 = 5$, $s_2 = 0$, $s_3 = 0$.

- Now, by complementary slackness

$$s_1 > 0 \quad \Rightarrow y_1 = 0 \qquad \text{and } \; x_1 > 0$$
$$\Rightarrow t_1 = 0 \qquad \text{and } \; x_2 > 0 \quad \Rightarrow t_2 = 0$$

- Hence the dual constraints provide the simultaneous equations

$$\begin{aligned} y_2 + y_3 &= 4 \\ y_2 + 2y_3 &= 5 \end{aligned}$$

- These provide the optimal values 3 and 1 of the dual variables y_2 and y_3 respectively.
- Thus the optimal solution of the dual can also be obtained without use of the simplex algorithm.

References for Chapter 3

- *W. L. Winston* O.R. (4th ed.)
  - 6.5 (defining duals),
  - 6.7 (duality thm.),
  - 6.10 (complementary slackness).
- *H. A. Taha* O.R. (9th e.d.)
  - 4.1 (defining duals)
  - 4.2.3 (optimal dual sln.),
  - 7.4 (matrix representation)

Outline

3. Duality
   - Definition of the dual problem
   - Duals of non-canonical problems
   - Relationships between the primal and the dual
   - Complementary slackness

4. Integer Programming
   - Introduction
   - The Branch-and-Bound (B&B) algorithm

5. Solving a LP using a spreadsheet

Definition 5 (Integer Linear Program)

An **Integer Linear Program** (ILP) is a LP in which some or all of the variables must be (non-negative) integers. If all of the variables must be integers, we have a **pure** ILP. If only some of the variables must be integers, we have a **mixed** ILP.

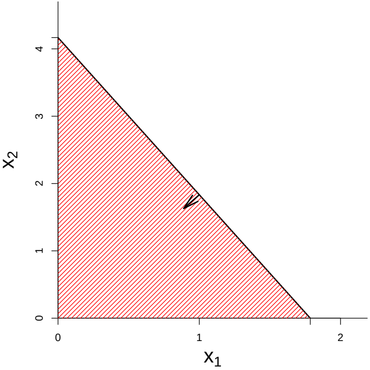
Example H:

Consider the LP:

$$\left.\begin{array}{lrcrcl} \textbf{Maximise} & Z & = & 28x_1 & + & 11x_2 \\ \text{subject to} & 14x_1 & + & 6x_2 & \leq & 25, \\ & x_1 \geq 0, & & x_2 \geq 0. & & \end{array}\right\} \quad (32)$$

The optimal solution is $(x_1, x_2) = \left(\frac{25}{14},\ 0\right)$.

Example H



- Suppose we turn the problem into a pure ILP, and thus demand an integer-valued solution.
- Naive rounding will often not give the optimal solution to the corresponding ILP.
- Here the rounded value (2, 0) is not even feasible.

To avoid complete enumeration of all feasible integer solutions, ILPs can be solved using:

(a) **cutting methods**, which sequentially cut out parts of the FR, without slicing off feasible integer solutions. The process continues until the optimal vertex of the reduced FR is integer-valued.

(b) **search methods**, which define a series of sub-problems, enabling gradual elimination of parts of the FR.

3 Duality
- Definition of the dual problem
- Duals of non-canonical problems
- Relationships between the primal and the dual
- Complementary slackness

4 Integer Programming
- Introduction
- The Branch-and-Bound (B&B) algorithm

5 Solving a LP using a spreadsheet

- Assume that we have a maximisation-type mixed ILP such that x_j $(j \in J)$ must be an integer.
- The B&B algorithm is a search method that defines appropriate sub-problems which, when represented graphically, correspond to branches of a tree.
- The sub-problems to be investigated are taken from a **candidate list** of LPs which are variants on the original ILP.
- At any stage the greatest value so far found of the objective function Z of the ILP is called the **encumbent.** [So, of the solutions so far found for which x_j $(j \in J)$ is an integer, the encumbent is the best.]
- We denote the encumbent by z_L as it gives a lower bound on the optimum value of Z.

The Branch-and-Bound algorithm

Step 1 (Initialisation)

Start the candidate list with the original ILP. Set $z_L = -\infty$.

Step 2 (Branching)

If the candidate list is empty, stop. If not, select a problem (called a **candidate problem** and denoted CP) from the candidate list on a last-in-first-out (LIFO) basis.

Step 3 (Relaxation and Bounding)

Relax any integer restrictions on CP, and solve the resulting LP.
If it has no feasible solution (NFS), or the optimum value is $\leq z_L$, go to Step 2.
If the optimum value is greater than z_L and the solution is such that x_j $(j \in J)$ is an integer, update the encumbent and go to Step 2.
If not, go to Step 4.

The Branch-and-Bound algorithm (cont.)

Step 4 (Separation)

Select any variable x_j ($j \in J$) that is not an integer in the optimal solution to CP. Suppose $x_j = b$ and let $[b]$ denote the integer part of b. Add two new problems to the candidate list, the first being CP with the additional constraint $x_j \leq [b]$ and the second CP with the additional constraint $x_j \geq [b] + 1$.

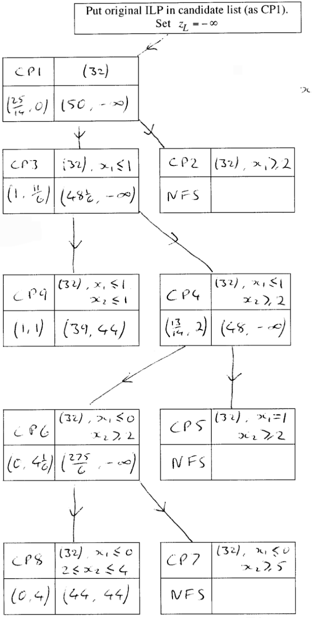
Go to Step 2.

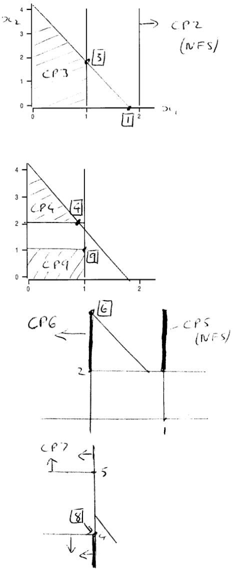
Example H

We apply the B&B algorithm, assuming that both x_1 and x_2 must be integers (i.e. $J = \{1, 2\}$). On the tree on the next slide, each box, except the first, is of the form

Candidate Problem No.	Constraints
Optimal (x_1, x_2) if such exists	(Optimal Z, z_L)

The objective function $Z = 28x_1 + 11x_2$ stays unchanged and is not shown. On the plots, the optimal solution to CPi is shown as $\boxed{i}$.





Thus the solution to CP9,
namely $Z =$ 39
does not improve on
the encumbent, $z_L =$ 44.
So, from CP8,
the optimal solution
to the ILP is
(x_1, x_2) = (0, 4)
yielding $Z =$ 44.

Note

(i) If you stop while $z_L = -\infty$, the ILP has NFS.

(ii) a CP is said to be **fathomed** if

  (a) it has been seen to have NFS *or*
  (b) its optimal value is less than or equal to z_L *or*
  (c) its optimal solution is greater than z_L and such that x_j $(j \in J)$ is an integer.
  So if a CP has been fathomed, investigation of that branch of the tree has been completed, resulting in either an updated encumbent or elimination of the branch. Note that Step 4 of the algorithm is only reached if a CP is not fathomed.

(iii) More sophisticated versions of the algorithm have been devised. These include better systems than LIFO for selecting from the candidate list, and more careful selection of the variable used for separation in Step 4.

References for Chapter 4

- *Taha*: O.R. (9th ed.))
    - 9.2.1
- *Winston*: O.R. (4th ed.)
    - 9.3
    - 9.2

Outline

3 Duality
  - Definition of the dual problem
  - Duals of non-canonical problems
  - Relationships between the primal and the dual
  - Complementary slackness

4 Integer Programming
  - Introduction
  - The Branch-and-Bound (B&B) algorithm

5 Solving a LP using a spreadsheet

Refer to the document *Chapter5.pdf* on the MMS.

Outline

6 Transportation Problems (TPs)
- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

- Transportation Problems (TPs) are an important, and useful, class of LP with a special structure.
- Solution by the simplex algorithm is possible, though usually the number of variables will be relatively large.
- The transportation algorithm gives an easier way of implementing the simplex method.

6. Transportation Problems (TPs)
   - LP formulation
   - The dual of a TP
   - Finding an initial vertex
   - Cell Evaluations
   - The general step in the transportation algorithm
   - Unbalanced problems
   - Sensitivity Analysis

Example J

The manager of a car-hire firm needs to re-deploy some of his stock. At towns D_1, D_2 and D_3, he requires 24, 20 and 16 cars respectively. To meet these shortfalls, he has decided to take 20, 17, 10 and 13 cars from towns S_1, S_2, S_3 and S_4 respectively. The cost (in units of £10) of transporting *one* car from S_i $(i = 1, 2, 3, 4)$ to D_j $(j = 1, 2, 3)$ is given in the body of the following table:

	D_1	D_2	D_3	Supply
S_1	7	3	2	20
S_2	10	7	6	17
S_3	9	4	5	10
S_4	9	5	1	13
Demand	24	20	16	60

How many cars should be sent from S_i $(i = 1, 2, 3, 4)$ to D_j $(j = 1, 2, 3)$ to minimise the total transportation cost?

Definition 6 (Balanced TP)

A TP is **balanced** if total demand = total supply.

Example J (cont.)

Here total demand = 60 = total supply. ⇒ The TP is balanced.
Let x_{ij} = Number of cars to be sent from S_i ($i = 1, 2, 3,\ 4$) to D_j ($j = 1, 2, 3$).
We wish to

$$\textbf{Minimise} \quad Z = 7x_{11} + 3x_{12} + \ldots + 5x_{42} + x_{43}$$

subject to

$$\text{Supply constraints} \left\{ \begin{array}{c} x_{11} + x_{12} + x_{13} \leq 20 \\ \vdots \\ x_{41} + x_{42} + x_{43} \leq 13 \end{array} \right\} \qquad (33)$$

$$\text{Demand constraints} \left\{ \begin{array}{c} x_{11} + x_{21} + x_{31} + x_{41} \geq 24 \\ \vdots \\ x_{13} + x_{23} + x_{33} + x_{43} \geq 16 \end{array} \right\} \qquad (34)$$

Non-negativity constraints $x_{ij} \geq 0 \qquad (i = 1, 2, 3, 4; \quad j = 1, 2, 3)$.

- Constraints (33) imply $\sum_{i=1}^{4} \sum_{j=1}^{3} x_{ij} \leq 60,$ whilst constraints (34) imply $\sum_{i=1}^{4} \sum_{j=1}^{3} x_{ij} \geq 60$

- Hence balance implies inequalities (33) and (34) must all be equalities.

- In the general *balanced* TP, we have

  a supply of a_i units at source S_i $(i = 1, 2, \ldots, m)$,
  a demand for b_j units at destination D_j $(j = 1, 2, \ldots, n)$,
  a cost c_{ij} for taking one unit from S_i $(i = 1, 2, \ldots, m)$ to D_j $(j = 1, 2, \ldots, n)$.

We let $x_{i\,j}$ = Number of units taken from S_i $(i = 1, 2, \ldots, m)$ to D_j $(j = 1, 2, \ldots, n)$. We wish to

$$\textbf{Minimise} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{i\,j}\, x_{i\,j}$$

Supply constraints:

$$\sum_{j=1}^{n} x_{i\,j} = a_i \qquad (i = 1, 2, \ldots, m)$$

Demand constraints:

$$\sum_{i=1}^{m} x_{i\,j} = b_j \qquad (j = 1, 2, \ldots, n)$$

Non-negativity constraints:

$$x_{i\,j} \geq 0 \qquad (i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n)$$

6 Transportation Problems (TPs)

- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

- In §3.2 we considered a rule for finding the dual of a LP with equality constraints.
- We now apply it to TPs.

Example K

Consider the balanced TP with cost table:

	D_1	D_2	D_3	Supply
S_1	7	11	9	4
S_2	8	10	12	6
Demand	2	3	5	10

Primal

$$\textbf{Minimise } Z = 7x_{11} + 11x_{12} + 9x_{13} + 8x_{21} + 10x_{22} + 12x_{23}$$

subject to

$$x_{11} + x_{12} + x_{13} = 4 \qquad (35)$$

$$x_{21} + x_{22} + x_{23} = 6 \qquad (36)$$

$$x_{11} + x_{21} = 2 \qquad (37)$$

$$x_{12} + x_{22} = 3 \qquad (38)$$

$$x_{13} + x_{23} = 5 \qquad (39)$$

$$x_{ij} \geq 0 \quad (i = 1, 2; \quad j = 1, 2, 3).$$

Denote the dual variables associated with the $\left\{ \begin{array}{l} \text{supply constraints by } u_1 \text{ and } u_2; \\ \text{demand constraints by } v_1, v_2 \text{ and } v_3. \end{array} \right.$

Dual

$$
\begin{array}{llllllll}
\textbf{Maximise} & W = & 4u_1 & +6u_2 & +2v_1 & +3v_2 & +5v_3 & \\
\text{subject to} & & & & & & & \\
 & & u_1 & & +v_1 & & & \leq 7 \\
 & & u_1 & & & +v_2 & & \leq 11 \\
 & & u_1 & & & & +v_3 & \leq 9 \\
 & & & u_2 & +v_1 & & & \leq 8 \\
 & & & u_2 & & +v_2 & & \leq 10 \\
 & & & u_2 & & & +v_3 & \leq 12
\end{array}
$$

u_i ($i = 1, 2$) and v_j ($j = 1, 2, 3$) unrestricted in sign.

- In general, each dual constraint of a TP contains exactly two decision variables.
- As u_i is associated with row i of the cost table ($i = 1, 2, \dots, m$), and v_j is associated with column j ($j = 1, 2, \dots, n$), there is one dual constraint associated with each cell (i, j) of the table.

Note

Note Although a TP has a total of $m + n$ supply and demand constraints, only $m + n - 1$ of them are independent : the final one is redundant. e.g. in example K

$$(39) = (35) + (36) - (37) - (38)$$

6 Transportation Problems (TPs)

- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

The North-West corner rule

The main advantage of this method, which ignores the costs c_{ij}, is its simplicity. The resulting vertex is often far from being optimal.

The North-West corner rule

(i) To the cell in the NW corner, make as large an allocation as is compatible with the constraints.

(ii) Revise the marginal entries to reflect this allocation.

(iii) Cover up the satisfied row or column. Unless the complete table is now covered, consider the resulting reduced table and return to step (i).

Note

If both the row and the column constraints are satisfied simultaneously (other than at the end), cover up *either* the row *or* the column (*not both*). The next allocation will then be a zero. This procedure ensures that allocations will be made to exactly $m + n - 1$ of the $m \cdot n$ cells in the table.

Example J

Following the above algorithm gives:

	D_1	D_2	D_3	Supply
S_1	20			~~20~~
S_2	4	13		~~17~~ ~~13~~
S_3		7	3	~~10~~ ~~3~~
S_4			13	13
Demand	~~24~~	~~20~~	~~16~~	60
	~~4~~	~~7~~	13	

Observe that the algorithm makes $4 + 3 - 1 = 6$ allocations. The ensuing total cost is

$$20c_{11} + 4c_{21} + 13c_{22} + 7c_{32} + 3c_{33} + 13c_{43} = 327 \text{ units} = £3270.$$

Vogel's Rule

Often (but not always) this yields a better initial vertex. This results in fewer iterations of the transportation algorithm being required to obtain the optimal solution.

Vogel's Rule

(i) For each row and column, find the difference obtained by subtracting the smallest from the second smallest of the costs c_{ij} in that row or column. (This indicates the size of the penalty that would be incurred by not using the cheapest cell in the row or column.)

(ii) Look for the row or column with the largest of these differences. (If there is a tie, it can be broken arbitrarily - though making the next allocation to whichever of the implied cells has the smallest c_{ij} may prove helpful).

Vogel's Rule (cont.)

(iii) Allocate as large an amount as possible to whichever cell in the chosen row or column has the smallest c_{ij}. (Again break any tie arbitrarily). Adjust the marginal entries appropriately.

(iv) If this allocation *either* exhausts the supply for a row *or* satisfies the demand for a column (*but not both*), delete the row or column as appropriate. If *both* a row and a column are satisfied, allocate a zero to whichever of the other remaining cells in the row or column has the smallest c_{ij} (breaking any tie arbitrarily), and delete *both* the row and the column.

(v) If only a single row or column remains, complete the table and stop. Otherwise, considering the reduced table, return to (i).

Example J

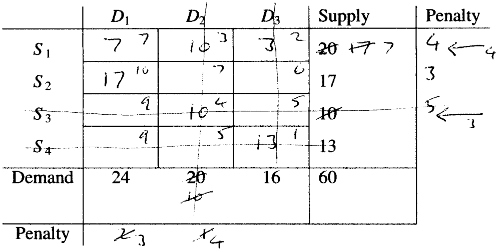
We draw up a transportation tableau as for the NW corner rule, but this time note the cost c_{ij} in the top right-hand corner of each cell. Following the above procedure for Vogel's Rule then gives:

	D_1	D_2	D_3	Supply	Penalty
S_1	7	3	3 2	~~20~~ 17	1
S_2	10	7	6	17	1
S_3	9	4	5	10	1
~~S_4~~	9	5	13 1	~~13~~	4 ← 1
Demand	24	20	~~16~~ ~~3~~	60	
Penalty	2	1	~~1~~ 3 ↑ 2		

The numbered arrows indicate the order of the allocations.

Example J (cont.)

Continuing in this way we get:



	D_1	D_2	D_3	Supply	Penalty
S_1	7 ^{7}	10 ^{3}	3 ^{2}	~~20~~ ~~17~~ 7	4 ← 4
S_2	17 ^{10}	^{7}	^{6}	17	3
~~S_3~~	^{9}	10 ^{4}	^{5}	~~10~~	5 ← 3
~~S_4~~	^{9}	^{5}	13 ^{1}	~~13~~	
Demand	24	~~20~~ ~~10~~	16	60	
Penalty	~~2~~ 3	~~1~~ 4			

Again 6 allocations have been made. The total cost at this vertex is £3080 which is better than the vertex found by the NW corner rule. In practice the complete allocation can be made on a single tableau.

6 Transportation Problems (TPs)

- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

- On slide 169 we found the dual of the particular TP specified there in Example K.
- Consider now the general balanced TP given on slide 165. If, similarly to Example K, we denote the dual variables associated with the supply constraints by u_i ($i = 1, 2, \ldots, m$) and those associated with the demand constraints by v_j ($j = 1, 2, \ldots, n$), the dual of the general balanced TP may be written as:

Maximise $W = \sum_{1}^{m} a_i u_i + \sum_{1}^{n} b_j v_j$

subject to $u_i + v_j \le c_{ij} \quad (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$

u_i and v_j unrestricted in sign.

- On slide 131 we noted that any dual variable is associated with a particular primal constraint.
- Interchanging the roles of the primal and the dual, any primal variable is associated with a particular dual constraint.
- In the above general balanced TP, the dual constraint $u_i + v_j \leq c_{i\,j}$ is associated with the variable $x_{i\,j}$.
- Now suppose, at some iteration, the variable $x_{i\,j}$ is basic. $\Rightarrow x_{i\,j} > 0$ (assuming no degeneracy).
- Hence, by complementary slackness, the dual constraint $u_i + v_j \leq c_{i\,j}$ is an equality. i.e. $u_i + v_j = c_{i\,j}$ when (i, j) is such that x_{ij} is basic.

Definition 7 (Cell evaluation)

For the $(i, \ j)^{th}$ cell $(i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n)$ of the transportation tableau, the **cell evaluation** is given by $\Delta_{ij} = c_{ij} - (u_i + v_j)$ Thus, if $x_{i\,j}$ is basic, $\Delta_{ij} = 0$.

Cell evaluations for cells corresponding to non-basic variables indicate whether or not a feasible solution to a balanced TP can be improved, and, if so, how this can be done. To see why this is so, we first derive another general property of dual problems. Recall that the dual of the LP

$$\begin{aligned} &\text{Maximise} \quad Z = \mathbf{c}^T\,\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} \le \mathbf{b} \text{ and } \mathbf{x} \ge \mathbf{0}. \end{aligned}$$

is given by

$$\begin{aligned} &\text{Minimise} \quad W = \mathbf{b}^T\,\mathbf{y} \\ &\text{subject to} \quad \mathbf{A}^T\mathbf{y} \ge \mathbf{c} \text{ and } \mathbf{y} \ge \mathbf{0}. \end{aligned}$$

- If $\mathbf{P}_j$ $(j = 1, 2, \ldots, n)$ denotes the jth column of $\mathbf{A}$, the jth dual constraint reads $\mathbf{P}_j^T\mathbf{y} \geq c_j$.
- Thus, with the dual written in canonical form, the difference between the LHS and the RHS of the j^{th} dual constraint is $\mathbf{P}_j^T\mathbf{y} - c_j$.
- As this is a scalar quantity, we may take transposes, and write it as $\mathbf{y}^T\mathbf{P}_j - c_j$.
- In particular, if $\mathbf{c}_B^T\mathbf{B}^{-1}$ is the vector of simplex multipliers for a particular iteration and we set $\mathbf{y}^T = \mathbf{c}_B^T\mathbf{B}^{-1}$, the difference between the LHS and the RHS is $\mathbf{c}_B^T\mathbf{B}^{-1}\ \mathbf{P}_j - c_j$.
- As we saw slide 111, this is the entry for x_j in the OF row of the primal tableau for that iteration.

N.B.

When evaluated at the simplex multipliers for a particular iteration, the difference between the LHS and the RHS of the j^{th} dual constraint (expressed in canonical form) is the j^{th} element of the OF row of the primal tableau for that iteration.

- Expressed in the canonical form for duals, the dual of the general balanced TP reads as follows:

$$\text{Minimise} \quad -W = -\sum_{i=1}^{m} a_i\, u_i - \sum_{j=1}^{n} b_j\, v_j$$

$$\text{subject to} \quad -u_i - v_j \geq -c_{i\,j}$$

$$(i = 1, \ldots, m) \quad (j = 1, \ldots, n)$$

u_i and v_j unrestricted in sign.

- The difference between the LHS and the RHS of the $(i,\ j)^{th}$ dual constraint is $c_{i\,j} - (u_i + v_j)$.
- When u_i and v_j are the simplex multipliers for a particular iteration, this difference equals the element in the $x_{i\,j}$ column of the OF row of the primal tableau.
- In particular, if $x_{i,j}$ is a basic variable for that iteration, then, as above $c_{i,j} - (u_i + v_j) = 0$.

- Another way of approaching these results is to consider a modified form of the general balanced TP.

- Let u_i $(i = 1, \ldots , m)$ and v_j $(j = 1, \ldots , n)$ be arbitrary constants and consider a TP which only differs from that given on slide 165 in having the cost c_{ij} replaced by $c_{ij} - (u_i + v_j)$.

- The OF for the modified problem is

$$\tilde{Z} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left\{ c_{ij} - (u_i + v_j) \right\} x_{ij} =$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \, x_{ij} - \sum_{i=1}^{m} u_i \sum_{j=1}^{n} x_{ij} - \sum_{j=1}^{n} v_j \sum_{i=1}^{m} x_{ij}$$

$$= Z - \sum_{i=1}^{m} a_i \, u_i - \sum_{j=1}^{n} b_j \, v_j \qquad \text{(see slide 165)}$$

- So the problem remains essentially unchanged in the sense that the optimum values of the variables will be the same.
- Suppose, furthermore, that for some particular set of basic variables we choose the constants u_i and v_j so that $c_{ij} - (u_i + v_j) = 0$ whenever x_{ij} is basic.
- The above expression for $\tilde{Z}$ then gives the modified OF in terms of variables which are non-basic as is required when producing a simplex tableau.
- Since minimising $\tilde{Z}$ is equivalent to maximising -$\tilde{Z}$, the entry in the x_{ij} column of the (primal) OF row is $c_{ij} - (u_i + v_j)$.

- We saw in the last section that when u_i and v_j are the simplex multipliers for a particular iteration, the cell evaluation $\Delta_{ij} = c_{ij} - (u_i + v_j)$ is zero whenever x_{ij} is basic.

- As there are $m + n - 1$ basic variables, this gives $m + n - 1$ equations in $m + n$ unknowns, namely $u_1, \dots, u_m, v_1, \dots, v_n$.

- To solve them, arbitrarily make one simplex multiplier zero, and solve for the rest. (All sums of the form $u_i + v_j$ are then uniquely determined).

Example J

Using the NW corner solution (cost = 327 units), we generate the simplex multipliers by first arbitrarily setting $v_1 = 0$. We then get (see explanation below):

				u_i
	7 20	4 3 -	5 2 ✓ –	7
	10 4	7 13	8 6 -	10
	7 9 +	4 7	5 3	7
	3 9 +	0 5 +	1 13	3
v_j	0	-3	-2	

Example J (cont.)

The other simplex multipliers were found as follows:

$$\begin{aligned} u_1 + v_1 &= 7 & \Rightarrow \quad u_1 &= 7 \\ u_2 + v_1 &= 10 & \Rightarrow \quad u_2 &= 10 \\ u_2 + v_2 &= 7 & \Rightarrow \quad v_2 &= -3 \\ u_3 + v_2 &= 4 & \Rightarrow \quad u_3 &= 7 \\ u_3 + v_3 &= 5 & \Rightarrow \quad v_3 &= -2 \\ u_4 + v_3 &= 1 & \Rightarrow \quad u_4 &= 3 \end{aligned}$$

For non-basic cells $(i, \ j)$, we evaluate $u_i + v_j$ and enter it in the top left-hand corner of the cell. For example:
$u_1 + v_2 = 7 - 3 = 4, \quad u_1 + v_3 = 7 - 2 = 5, \quad \text{etc.}$

Example J (cont.)

- Next, for the non-basic cells, record in each cell whether the cell evaluation $\Delta_{ij} = c_{ij} - (u_i + v_j)$ is positive, negative or zero.

- Since Δ_{ij} = element in the x_{ij} column of the OF row of the primal tableau, we know that, if $\Delta_{ij} < 0$ for any cell, the solution is not optimal.

- Moreover, the *entering variable* will have the cell evaluation which is most negative (with any tie being broken arbitrarily) and we mark the cell with a tick.

- Here x_{13} becomes basic, and will be given some positive value θ.

Example J (cont.)

Now increasing x_{13} by θ

$\Rightarrow$ reducing x_{11} by θ (to meet the row constraint)
$\Rightarrow$ increasing x_{21} by θ (to meet the column constraint)
$\Rightarrow$ reducing x_{22} by θ (to meet the row constraint)
$\Rightarrow$ increasing x_{32} by θ (to meet the column constraint)
$\Rightarrow$ reducing x_{33} by θ (to meet the row constraint)

In general, consequential effects will be felt around a “loop” of cells, which is always uniquely defined.

Example J (cont.)

In the corresponding simplex tableau, the elements in the *pivot* column (namely the x_{13} column) and the ratio column are as follows:

Row	Pivot column element	Ratio column
x_{43} (basic variable not in the loop)	0	∞
x_{21} and x_{32}	-1	Negative
x_{11}, x_{22} and x_{33}	1	Current value of the variable

Hence, by the usual rule for the ratio column, the *leaving variable* is whichever of x_{11}, x_{22} and x_{33} currently has the smallest allocation.

Example J (cont.)

- So here, x_{33} becomes non-basic. $\Rightarrow$ $\theta = 3.$ (Equivalently, we make θ as large as it can be without any basic variable in the loop going negative.)

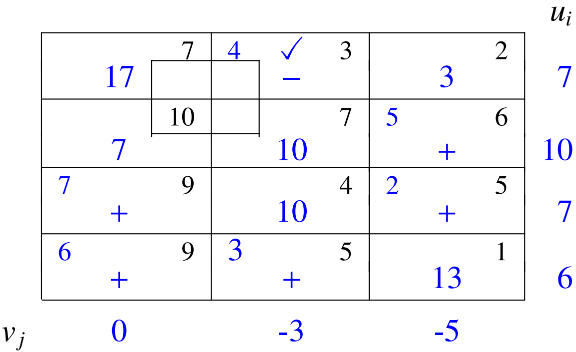
- So the next allocation is:

7 17	3	2 3
10 7	7 10	6
9	4 10	5
9	5	1 13

- New total cost = 318 units
  $[\, = 327 \; + \theta\, \Delta_{13}$, where $\theta \; = \; 3, \; \Delta_{13} \; = \; -3 \,].$

Example J (cont.)

Testing for optimality:



				u_i
	7 17	4 ✓ 3 −	2 3	7
	10 7	7 10	5 6 +	10
	7 9 +	4 10	2 5 +	7
	6 9 +	3 5 +	1 13	6
v_j	0	-3	-5	

So the next entering variable is x_{12}.

Example J (cont.)

We push 10 units around the loop shown, giving the following new allocation, and again test for optimality:

				u_i
	7 7	3 10	2 3	0
	10 17	6 7 +	5 6 +	3
	8 9 +	4 10	3 5 +	1
	6 9 +	2 5 +	1 13	-1
v_j	7	3	2	

Total cost = 308 units [= 318 + $\theta\ \Delta_{12}$, where $\theta = 10$, $\Delta_{12} = -1$].

All cell evaluations are now positive, so this is the unique optimal solution. Observe that, for this problem, Vogel's rule gives the optimal solution immediately.

Notes

(i) The transportation algorithm can also be used when the OF is to be *maximised*, since maximising

$$Z = \sum_i \sum_j c_{ij} x_{ij}$$

is equivalent to minimising

$$-Z = \sum_i \sum_j \left(-c_{ij}\right) x_{ij}.$$

If all the new 'costs' $-c_{ij}$ are negative, it is convenient to add a constant to all of them to make all entries non-negative. Thus we choose to minimise

$$\tilde{Z} = \sum_i \sum_j \left(C - c_{ij}\right) x_{ij}, \quad \text{where } C = \max\left(c_{ij}\right).$$

It follows from the derivation at the end of §6.4 that adding the constant C will leave the solution of the TP unchanged. The ensuing value of the OF will, of course, change.

Notes (cont.)

(ii) When pushing an amount θ around a loop, the allocation may drop to zero for *more than one* cell. In this case, arbitrarily remove the allocation from one of these cells, but leave the rest with zero allocations (to ensure that $m + n - 1$ cells still have allocations.)

(iii) When one or more of the $m + n - 1$ allocations is zero, sometimes only a "zero amount" can be shunted round the loop. This should nonetheless be done, since changing the cell(s) with zero allocations enables the algorithm to proceed.

Notes (cont.)

(iv) If no Δ_{ij} are negative, but one or more is zero, the current allocation is optimal, but **non-unique**. (cf. Non-uniqueness on slides 57-58.)

Making an allocation to a cell for which $\Delta_{ij} = 0$ will yield a different optimal solution. (Although if there is a zero in the optimal solution it may just reallocate it.)

e.g. if in **Example J** we change c_{23} from 6 to 5, the above solution is still clearly optimal, but $\Delta_{2\,3} = 0$. In this case making an allocation to cell (2,3) yields a different optimal solution (see next slide).

Previous:

7 7	3 10	2 3
10 17	6 7 +	5 ✓ 5 0
8 9 +	4 10	3 5 +
6 9 +	2 5 +	1 13

New:

7 10	3 10	2
10 14	7	5 3
9	4 10	5
9	5	1 13

Total cost = 308 units = £3080 (as before)

Notes (cont.)

(v) To see why the check on the total cost works, consider the simplex tableau for the iteration in question. Suppose $(i,\ j)$ is the pivot cell. By the choice of leaving variable, the entry on the RHS of the pivot row is θ and $-\Delta_{ij}$ times the pivot row is added to the OF row to create a zero there in the pivot column. This increases - Z, the variable being maximised, by $-\theta\,\Delta_{ij}$ and hence adds $\theta\,\Delta_{ij}$ to Z (thus reducing Z).

(vi) In a TP, if, for whatever reason, source S_i cannot supply destination D_j, set $c_{ij} = M$, where M is a large positive number.

Summary of the Transportation Problem

Step 1.

Obtain an initial feasible basic solution using, for example, the NW corner rule or Vogel's rule. (The one produced by the latter is likely to be better.)

Step 2.

(a) Determine simplex multipliers u_i and v_j such that $u_i + v_j = c_{ij}$ for all cells (i, j) for which x_{ij} is currently a basic variable.

(b) Evaluate $u_i + v_j$ for all cells (i, j) for which x_{ij} is currently non-basic.

Summary of the Transportation Problem (cont.)

Step 3.

(a) If all cell evaluations are positive, the unique optimal solution has been found, so stop.

(b) If one or more cell evaluations are zero and the rest positive, an optimal solution has been found, but other optimal solutions may exist (see note (iv) above). If the present solution is one that you have reached previously, stop. If not, select an entering variable corresponding to a cell with a cell evaluation of zero, and go to **Step 4**.

(c) If there is a negative cell evaluation, optimality has not yet been reached. The entering variable corresponds to the cell with the most negative cell evaluation (Break any tie arbitrarily).

Summary of the Transportation Problem (cont.)

Step 4.

- Identify the loop of cells corresponding to basic variables whose values will be affected by giving the entering variable a positive value.
- Of the cells at the vertices of the loop, identify those at which the allocation is to be reduced.
- Of these cells, identify the cell with the smallest allocation. (If there is a tie, see note (ii)). The leaving variable corresponds to this cell.
- Denote by θ this smallest allocation.
- Give the entering variable the value θ and make consequential changes around the loop.
- Return to **Step 2** to test the new solution for optimality.

6 Transportation Problems (TPs)

- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

An unbalanced TP must be balanced before the above algorithm can be applied

(i) **Total supply > Total demand**

In this case a dummy destination must be created:

Example J

If the demand at D_2 falls to 15 cars, we create a dummy destination D_4, with a demand for 5 cars and with per unit transportation costs of zero.

We then solve the following balanced TP:

	D_1	D_2	D_3	D_4	Supply
S_1	7	3	2	0	20
S_2	10	7	6	0	17
S_3	9	4	5	0	10
S_4	9	5	1	0	13
Demand	24	15	16	5	60

The cars ‘sent’ to D_4 will just stay where they are.

- In other problems, storing goods at the source may incur storage costs.
- Per unit storage costs, which may differ between sources, should then be used instead of the above zero costs.

(ii) **Total supply < Total demand**

Here a dummy source is created, giving a new row in the cost table, either of zeroes or of per unit penalty costs (which may differ between destinations) for failing to satisfy demand.

6 Transportation Problems (TPs)

- LP formulation
- The dual of a TP
- Finding an initial vertex
- Cell Evaluations
- The general step in the transportation algorithm
- Unbalanced problems
- Sensitivity Analysis

We examine the robustness of the optimal solution to a few of the perturbations which may be of interest. Throughout this section the terms ‘basic’ and ‘non-basic’ are used with respect to this optimal vertex.

(i) **Changing** c_{ij} **to** $c_{ij} - \Delta$**, where** $\Delta > 0$**, for a non-basic variable**

This will not affect

(a) the feasibility of the optimal solution, which depends only on the constraints, and not the OF,
nor
(b) the sums $u_i + v_j$ of the optimal simplex multipliers, which are determined by the basic cells.

So, if $\tilde{\Delta}_{ij}$ is the cell evaluation for the given cell under the previous optimal solution, this solution stays optimal *provided*
the new cell evaluation Δ_{ij} satisfies

$$0 \leq \Delta_{ij} = (c_{ij} - \Delta) - (u_i + v_j) = \tilde{\Delta}_{ij} - \Delta \quad \Leftrightarrow \quad \Delta \leq \tilde{\Delta}_{ij}$$

(ii) Changing c_{ij} to $c_{ij} + \Delta$ for a basic variable

Here the simplex multipliers do alter, so both positive and negative values of Δ should be considered.

Example J

If c_{12} becomes $3 + \Delta$, we may test whether the previous solution is still optimal:

				u_i
	7 7	$3+\Delta$ 10	2 3	0
	10 17	$6+\Delta$ 7 ?	5 6 +	3
	$8-\Delta$ 9 ?	4 10	$3-\Delta$ 5 ?	$1-\Delta$
	6 9 +	$2+\Delta$ 5 ?	1 13	-1
v_j	7	$3+\Delta$	2	

So the solution remains optimal if

$$0 \leq \Delta_{22} = 7 - (6 + \Delta) = 1 - \Delta \quad \Rightarrow \quad \Delta \leq 1$$

and $$0 \leq \Delta_{31} = 9 - (8 - \Delta) = 1 + \Delta \quad \Rightarrow \quad \Delta \geq -1$$

and $$0 \leq \Delta_{33} = 5 - (3 - \Delta) = 2 + \Delta \quad \Rightarrow \quad \Delta \geq -2$$

and $$0 \leq \Delta_{42} = 5 - (2 + \Delta) = 3 - \Delta \quad \Rightarrow \quad \Delta \leq 3$$

i.e. if and only if $-1 \leq \Delta \leq 1$.

(iii) Increasing *both* a_i and b_j by $\Delta > 0$ when x_{ij} is basic

Increasing supply and demand by the same amount implies the TP remains balanced. To find the new optimal solution simply increase x_{ij} by Δ.

Example J

Suppose alternatively that a_1 changes to 25 and b_3 to 21. The new optimal solution is:

	D_1	D_2	D_3	Supply
S_1	7	10	8	25
S_2	17			17
S_3		10		10
S_4			13	13
Demand	24	20	21	65

This is obvious as the solution is clearly feasible and the cell evaluations are all unchanged.

(iv) Increasing *both* a_i and b_j by $\Delta > 0$ when x_{ij} is non-basic

Here a new optimal solution can sometimes be found by finding the loop linking cell $(i,\ j)$ to the basic cells, notionally allocating Δ to cell $(i,\ j)$, and then removing it by shunting it around the loop.

Example J

Suppose alternatively that a_2 changes to $17\ +\ \Delta$ and b_3 to $16\ +\ \Delta$. Allocating Δ to cell (2, 3) gives:

	D_1	D_2	D_3	Supply
S_1	7	10	3	25
S_2	17		Δ	$17\ +\ \Delta$
S_3		10		10
S_4			13	13
Demand	24	20	$16\ +\ \Delta$	$60\ +\ \Delta$

When $\Delta = 6$, redistribution gives:

	D_1	D_2	D_3	Supply
S_1	1	10	9	20
S_2	23			23
S_3		10		10
S_4			13	13
Demand	24	20	22	66

This solution must be optimal as it is feasible and the cell evaluations are unchanged. If $\Delta = 8$, however, the above optimal variables cannot provide a fbs. [The row 2 constraint requires $x_{21} = 25$, breaking the column 1 constraint.]

(v) **Increasing a single supply or demand**

This makes the TP unbalanced, necessitating a dummy source or a dummy destination. To seek a new optimal solution, try a modification of the previous one (e.g. leaving surplus goods at their source).

References

author	title (ed.)	sections
W.L. Winston	O.R. (4th ed.)	7.1-7.4
H.A. Taha	O.R. (9th ed.)	5.1, 5.3

Outline

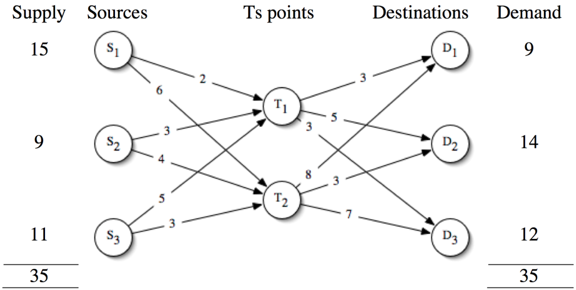
7 Transshipment Problems (TsPs)
- Buffer stocks approach
- Cheapest route approach
- Shortest route problems (SRPs) as TsPs

8 Network Flow Problems
- A more graphical approach
- The network flow algorithm

9 Assignment Problems
- LP Formulation
- The Hungarian Assignment Algorithm (HAA)

- Sometimes goods pass through intermediate transshipment (Ts) points between sources and destinations (e.g. products go from factories to warehouses, and then on to retailers).
- Such problems can be solved by construing them as TPs:

Example L



The numbers on the arcs give the per unit transportation costs.

Observe that this problem is balanced.

- A Ts point adds both a row and a column to the TP tableau as it serves as both a source and a destination.
- In this network, no goods go directly from S_i $(i = 1, 2, 3)$ to D_j $(j = 1, 2, 3)$, nor from T_1 to T_2, nor from T_2 to T_1.
- In the TP tableau, we notionally allow these moves, but assign a very large positive cost M, so that no goods will be allocated to these routes.
- If the warehouses at T_1 and T_2 are sufficiently big, either may handle the total number of units, which is $B = 35$, so we give them demands and supplies of 35.
- If their demand is not satisfied from the sources, any shortfall is accounted for by allowing them to supply themselves, with the routes $T_1 \to T_1$ and $T_2 \to T_2$, both with zero cost, serving as notional internal transactions.

Applying Vogel's rule to the resulting TP tableau:

	T_1	T_2	D_1	D_2	D_3	Supply	Penalty
S_1	2 15	6	M	M	M	~~15~~	4 ← 2
S_2	3 6	4 3	M	M	M	9	1
S_3	5	3 11	M	M	M	~~11~~	2 ← 7
T_1	0 14	M	3 9	5	3 12	~~35~~ ~~26~~ ~~14~~	~~3~~ 5 ← 4
T_2	M	0 21	8	3 14	7	~~35~~ ~~21~~	~~3~~ M ← 6
Demand	~~35~~ ~~20~~ 6	~~35~~ ~~14~~ 3	~~9~~	~~14~~	~~12~~		
Penalty	~~2~~ ~~3~~ 2	~~3~~ 1	5 ↑ 1	~~2~~ M-3 ↑ 5	4 ↑ 3		

- All cell evaluations are positive. (Check!)
- So this is the unique optimal solution.
- i.e.:

  T_1 gets 15 units from S_1, 6 from S_2, and sends 9 to D_1 and 12 to D_3.

  T_2 gets 3 units from S_2, 11 from S_3, and sends all 14 to D_2.

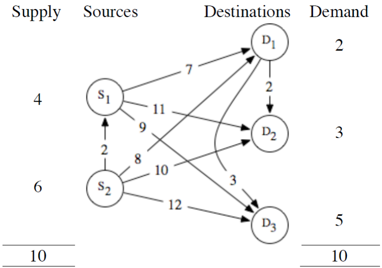
- The 14 units going from T_1 to T_1, and the 21 units going from T_2 to T_2 are purely notional, corresponding to amounts these Ts points might have received but didn't.

- In a more general model, sources and destinations may also serve as Ts points.
- We assume that the problem is balanced (either naturally or by construction), and that

$$\text{Total supply} = B = \text{Total demand.}$$

- Such problems can be solved by construing them as TPs:

Example N



Example N (cont.)

- So S_1 and D_1 must appear as both sources and destinations.
- Again each Ts point can notionally supply goods to itself at zero cost, whilst the route $D_1 \to S_1$ gets the large positive cost M.
- Potentially both S_1 and D_1 may have to handle the total demand $B = 10$, so their demands and supplies must be increased so that
  - (a) the flow of goods is not unnecessarily restricted, and
  - (b) the problem stays balanced.

Example N (cont.)

- So we give S_1 a demand of $B = 10$ units and D_1 a supply of $B = 10$ units.
- We increase the supply at S_1 from 4 to $B + 4$ units and the demand at D_1 from 2 to $B + 2$ units.
- In fact, any sufficiently large **buffer** B suffices.
- It does not matter that $B + 4$ and $B + 2$ exceed the total supply.
- More precise values are unnecessary as any surplus is absorbed at each Ts point by letting notional stock stay there.

Example N (cont.)

Thus an equivalent TP has the per unit costs shown in the following table:

	D_1	D_2	D_3	S_1	Supply
S_1	7	11	9	0	14
S_2	8	10	12	2	6
D_1	0	2	3	M	10
Demand	12	3	5	10	30

N.B.

In general, in the equivalent TP tableau given by the buffer stocks approach:

(i) there is a row for each Ts point and for each source which is not a Ts point;

(ii) there is a column for each Ts point and for each destination which is not a Ts point;

(iii) for each Ts point, the supply is $B+$ actual supply there (if any), and the demand is $B+$ actual demand there (if any);

(iv) each Ts point can supply itself at zero cost;

(v) any cells corresponding to non-existent routes are given per unit cost M.

7. Transshipment Problems (TsPs)
   - Buffer stocks approach
   - Cheapest route approach
   - Shortest route problems (SRPs) as TsPs

8. Network Flow Problems
   - A more graphical approach
   - The network flow algorithm

9. Assignment Problems
   - LP Formulation
   - The Hungarian Assignment Algorithm (HAA)

- It is tacitly assumed in a TP that

  c_{ij} = per unit cost by the *cheapest* route from S_i $(i = 1, 2, \ldots, m)$ to D_j $(j = 1, 2, \ldots, n)$.

- Determining the value of c_{ij} $(i = 1, 2, \ldots, m;\ \ j = 1, 2, \ldots, n)$ may require preliminary work.

- Another way to reduce a TsP to a TP is to calculate first the cheapest route between each source and each destination.

- The Ts points can then be ignored, and the TP solved as usual.

Example N

- The cheapest routes here are easily found by inspection.

- So an equivalent TP is:

	D_1	D_2	D_3	Supply
S_1	7	9	9	4
S_2	8	10	11	6
Demand	2	3	5	10

- In larger problems, the cheapest routes can be found by an application of Floyd's algorithm, or by applications of Dijkstra's algorithm (both outwith the syllabus for this module).

7. Transshipment Problems (TsPs)
   - Buffer stocks approach
   - Cheapest route approach
   - Shortest route problems (SRPs) as TsPs

8. Network Flow Problems
   - A more graphical approach
   - The network flow algorithm

9. Assignment Problems
   - LP Formulation
   - The Hungarian Assignment Algorithm (HAA)

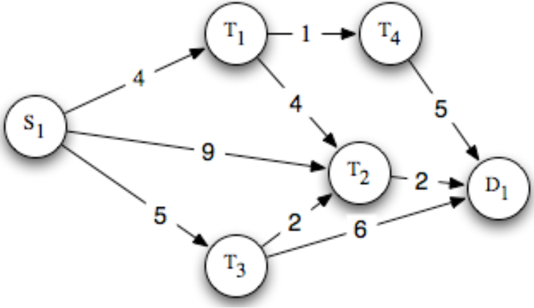
- Shortest route algorithms offer one way of solving TsPs.
- Conversely, the buffer stocks method of solving TsPs can be used to find shortest routes (SRs).

Example P

Consider the TsP:

Supply

1



Demand

1

Under the buffer stock approach, the corresponding TP has cost table:

	T_1	T_2	T_3	T_4	D_1	Supply
S_1	4	9	5	M	M	1
T_1	0	4	M	1	M	1
T_2	M	0	M	M	2	1
T_3	M	2	0	M	6	1
T_4	M	M	M	0	5	1
Demand	1	1	1	1	1	5

- The resulting optimal solution is to send 1 unit along the route

$$S_1 \to T_3 \to T_2 \to D_1$$

  at a cost $(1 \cdot 5) + (1 \cdot 2) + (1 \cdot 2) = 9$.

- So if the per unit transportation costs are proportional to the distance travelled, the TsP simply identifies the SR from S_1 to D_1.
- Clearly any problem of finding a SR between two specified places on a network can be viewed as a TsP by treating distances as per unit transportation costs, and notionally sending 1 unit from the source to the destination.

It follows that we can view a SRP of this type as a LP:

Example P

Relabel $S_1, T_1, \ldots, T_4,\ D_1$ as $L_1, \ldots, L_6$ respectively, and let

x_{ij} = Number of units taken from L_i to L_j $(i, j = 1, 2, \ldots, 6)$;
c_{ij} = per unit transportation cost from L_i to L_j $(i, j = 1, 2, \ldots, 6)$;

Also let $N = \{(i, j)$: there is an arc (i.e. direct route) from L_i to L_j in the network$\}$.

The original formulation then suggests the following LP:

$$\textbf{Minimise} \quad Z = \sum_{(i,j)\in N}\sum c_{ij}x_{ij} \tag{40}$$

$$\text{subject to} \qquad \sum_{j=2}^{4} x_{1j} = 1, \quad \sum_{i=3}^{5} x_{i6} = 1 \tag{41}$$

(i.e. one unit leaves $S_1 \equiv L_1$, and one unit arrives at $D_1 \equiv L_6$.)

$$\sum_{i\,:\,(i,k)\in N} x_{ik} = \sum_{j\,:\,(k,j)\in N} x_{kj} \qquad (k = 2, 3, 4, 5) \tag{42}$$

(i.e. no. of units arriving at L_k = no. of units leaving L_k.)

$$x_{ij} \geq 0 \qquad (i, j) \in N.$$

The equations (42) are called **conservation of flow** equations.

The buffer stocks formulation, given by the table on slide 232, expresses the TsP as a TP. For this the LP is (cf. slide 165):

$$\left.\begin{aligned}
&\text{Minimise } Z = \sum_{i=1}^{5} \sum_{j=2}^{6} c_{ij} x_{ij} \\
&\text{subject to } \quad \sum_{j=2}^{6} x_{ij} = 1 && (i = 1, \ldots, 5) \\
&\qquad\qquad\quad \sum_{i=1}^{5} x_{ij} = 1 && (j = 2, \ldots, 6) \\
&\qquad\qquad\quad x_{ij} \geq 0 \qquad (i = 1, \ldots, 5; && j = 2, \ldots, 6).
\end{aligned}\right\} \qquad (43)$$

Note

The LP (43) does not explicitly require the x_{ij} to be integer-valued (i.e. 0 or 1), but it can be shown that the optimal solution will be of this form. More generally, the TP given on slide 165 has an integer-valued optimal solution provided a_i $(i = 1, 2, \ldots, m)$ and b_j $(j = 1, 2, \ldots, n)$ are integers. For the initial allocations provided by the NW corner rule and Vogel's rule are integer-valued, and reallocations around loops result only in additions and subtractions of these integer values.

Alternatively, if the constraints are written in matrix form (using general LP notation) as $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, it can be shown that $\mathbf{A}$ is **totally unimodular** (i.e. the determinant of every square sub-matrix equals +1, -1 or 0). Use of Cramér's rule then shows that a basic solution is integer-valued.

References for Chapter 7

author	title (ed.)	section
H.A. Taha	O.R. (8th ed.)	5.5
W.L. Winston	O.R. (4 th ed.)	7.6

Outline

7 Transshipment Problems (TsPs)
- Buffer stocks approach
- Cheapest route approach
- Shortest route problems (SRPs) as TsPs

8 Network Flow Problems
- A more graphical approach
- The network flow algorithm

9 Assignment Problems
- LP Formulation
- The Hungarian Assignment Algorithm (HAA)

7 Transshipment Problems (TsPs)
- Buffer stocks approach
- Cheapest route approach
- Shortest route problems (SRPs) as TsPs

8 Network Flow Problems
- A more graphical approach
- The network flow algorithm

9 Assignment Problems
- LP Formulation
- The Hungarian Assignment Algorithm (HAA)

- The simplex algorithm can also be applied to TsPs and other network problems in a more graphical way.
- We have been using graphs to represent networks, with the vertices or nodes of the graphs corresponding to sources, destinations or Ts points.
- The arcs between the vertices have represented possible routes.
- The graphical approach to network problems is based on spanning trees in the graphs (regarding them as undirected):

Definition 8 (Spanning tree)

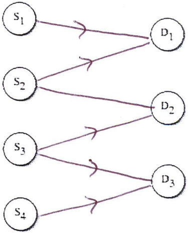
A subset of arcs in a network is a **spanning tree** if it connects all the vertices of the network without creating any loops (i.e. cycles).

- For a network with q vertices, a spanning tree consists of $q - 1$ arcs.

A TP with m sources and n destinations corresponds to a graph with $m+n$ vertices. A fbs uses $m+n-1$ basic variables, or equivalently arcs, and corresponds to a spanning tree.

Example J

Here we had 4 sources and 3 destinations, so the graph has 7 vertices. The spanning tree corresponding to the fbs that results from applying the NW corner rule (see slide 173) uses 6 arcs



- Consider a network on q vertices, $L_1, \ldots, L_q$, and suppose, as above, that N is the set of arcs in the network.
- A Network Flow Problem (NFP) can be defined by the following LP (cf. (41)-(42))

$$\textbf{Minimise} \qquad Z = \sum_{(i,j)\in N} c_{ij}x_{ij} \tag{44}$$

$$\text{subject to} \qquad \sum_{j:(k,j)\in N} x_{kj} - \sum_{i:(i,k)\in N} x_{ik} = d_k, \qquad k = 1, 2, \ldots, q \tag{45}$$

i.e. (flow out of L_k) − (flow into L_k) = d_k

$$x_{ij} \geq 0 \quad (i, j) \in N. \tag{46}$$

- Here d_k is positive, negative or zero depending on whether the vertex L_k is a source, a destination or a Ts point.
- Note that TPs, TsPs and SRPs can all be expressed in this form.
- Clearly the NFP must be balanced if there is a fbs to equations (45).
- A fbs has a corresponding spanning tree, but it is not always possible to find a feasible solution for a given spanning tree.
- Hence *feasible spanning trees* are required.

7. Transshipment Problems (TsPs)
   - Buffer stocks approach
   - Cheapest route approach
   - Shortest route problems (SRPs) as TsPs

8. Network Flow Problems
   - A more graphical approach
   - The network flow algorithm

9. Assignment Problems
   - LP Formulation
   - The Hungarian Assignment Algorithm (HAA)

- The dual of the above LP for the NFP can be written as

$$\begin{array}{lll} \textbf{Maximise} & W = \displaystyle\sum_{k=1}^{q} d_k y_k & \\ \text{subject to} & y_i \; - \; y_j \leq c_{ij} & (i, j) \in N \\ & y_i \text{ unrestricted in sign } (i = 1, 2, \ldots, q) & \end{array}$$

- Clearly there is a strong similarity between this dual and that of the general TP on slide 179.

- It can be shown that a similar approach can be used to examine whether a fbs is optimal, or whether an improved solution exists.

- We have $q \; - 1$ basic variables x_{ij}, each corresponding to an arc in the feasible spanning tree.
- Corresponding to each basic variable, we obtain from the dual constraint the equation

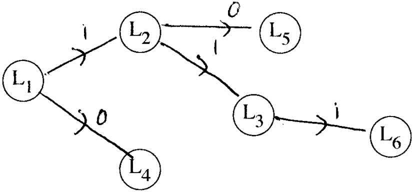
$$c_{ij} - y_i + y_j = 0$$

- This gives $q \; - 1$ equations in q unknowns, so arbitrarily setting $y_1 = 0$, yields a unique solution.
- For arcs which correspond to non-basic variables, we may then calculate the adjusted costs $\tilde{c}_{ij} = c_{ij} - y_i + y_j$.
- If the adjusted costs are all non-negative, the solution is optimal.

- If one or more adjusted costs is negative, the solution can be improved.
- As usual, select (one of) the most negative adjusted cost(s), and make the corresponding variable basic.
- Add to the feasible spanning tree the corresponding arc, thereby creating a loop.
- Assign this arc a flow of θ units in the direction specified on the digraph.
- Then, continuing in this direction, move around the loop making the consequential changes.
- This means adding θ units when you move with the flow and subtracting θ units when you go against the flow.
- Then choose θ to be as large as possible without making any flow negative.
- Discard from the spanning tree an arc for which the flow is reduced to zero.

Example P

- We label the vertices as $L_1, \ldots, L_6$ and define x_{ij} and c_{ij} as on slide 234.
- This SRP is then a NFP as given by (44)-(46), with $d_1 = 1$, $d_2 = d_3 = d_4 = d_5 = 0$ and $d_6 = -1$.
- Feasible spanning trees are easy to find in this problem, and one is shown below



Example P (cont.)

- The number alongside each arc is the number of units sent by that route.
- The basic variables for this fbs are $x_{12}, x_{14}, x_{23}, x_{25}$ and x_{36}.
- Noting the values of c_{ij} given on the digraph on slide 231, the simplex multipliers $y_1, y_2, \ldots, y_6$, satisfy the equations

$$\begin{array}{lll} y_1 - y_2 = 4 & y_2 - y_3 = 4 & y_1 - y_4 = 5 \\ y_2 - y_5 = 1 & y_3 - y_6 = 2 & \end{array}$$

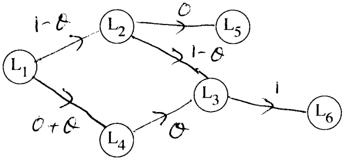
- Setting $y_1 = 0$ gives the solution $(y_1, y_2, \ldots, y_6) = (0,\ -4,\ -8,\ -5,\ -5,\ -10\ ).$

Example P (cont.)

Hence the adjusted costs for the arcs corresponding to non-basic variables are

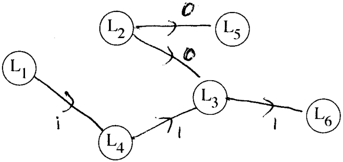
$$\begin{aligned}
\tilde{c}_{13} &= 9 - 0 - 8 = 1 \\
\tilde{c}_{43} &= 2 + 5 - 8 = -1 \\
\tilde{c}_{46} &= 6 + 5 - 10 = 1 \\
\tilde{c}_{56} &= 5 + 5 - 10 = 0
\end{aligned}$$

Hence we choose to make basic the variable x_{43}. The resulting loop is:



Example P (cont.)

- We therefore take $\theta = 1$.
- Arbitrarily choosing x_{12} as the leaving variable, the resulting redistribution gives the feasible spanning tree below.
- This is in fact an optimal solution (cf. slide 233), the substantive part of which is unique.



Notes

(i) In other problems an initial feasible spanning tree may not be obvious. In such cases, notional arcs with prohibitive costs M can be inserted linking each source directly to each destination, if such direct links are not already present. Standard TP methods of finding an initial solution can then be used, with further arcs with zero flows added to make a feasible spanning tree.

(ii) When changing the allocations around a loop, the allocation may drop to zero for *more than one* arc. In this case, arbitrarily remove the allocation from one of these arcs, but leave each of the rest with a notional zero flow (to retain the $q - 1$ arcs needed in the spanning tree).

(iii) A more general form of NFP can be defined in which there are upper and lower bounds on the capacities of arcs.

Summary of the Network Flow Algorithm

Step 1

Find an initial feasible spanning tree.

Step 2

(a) Determine simplex multipliers y_i $(i = 1, 2, \ldots, q)$ such that $y_i - y_j = c_{ij}$ for all arcs (i, j) in the feasible spanning tree.

(b) Evaluate the adjusted costs $\tilde{c}_{ij} = c_{ij} - y_i + y_j$ for all arcs (i, j) for which x_{ij} is currently non-basic (i.e arcs not in the tree).

Summary of the Network Flow Algorithm (cont.)

Step 3

(a) If all adjusted costs are positive, the unique optimal solution has been found, so stop.

(b) If one or more adjusted costs are zero and the rest positive, an optimal solution has been found, but other optimal solutions may exist. If the present solution is one that you have reached previously, stop. If not, select an entering variable corresponding to an arc with an adjusted cost of zero, and go to **Step 4**.

(c) If one or more adjusted costs are negative, optimality has not yet been reached. The entering variable corresponds to the arc with the most negative adjusted cost. (Break any tie arbitrarily).

Summary of the Network Flow Algorithm (cont.)

Step 4

Give the chosen arc a flow of θ units in the specified direction. Continue in this direction around the loop of arcs thus created, adding θ units when you move with the flow and subtracting θ units when you go against the flow. Choose θ to be as large as possible without making any flow negative. The leaving variable then corresponds to an arc in the loop with zero flow. (If two or more flows reduce to zero, see note (ii) above.) Delete this arc and return to **Step 2** to test the new feasible spanning tree for optimality.

References for Chapter 8

author	title (ed.)	section
H.A. Taha	O.R. (8th ed.)	20.1 (CD)
W.L. Winston	O.R. (4 th ed.)	8.5, 8.7

Outline

7. Transshipment Problems (TsPs)
   - Buffer stocks approach
   - Cheapest route approach
   - Shortest route problems (SRPs) as TsPs

8. Network Flow Problems
   - A more graphical approach
   - The network flow algorithm

9. Assignment Problems
   - LP Formulation
   - The Hungarian Assignment Algorithm (HAA)

7 Transshipment Problems (TsPs)
- Buffer stocks approach
- Cheapest route approach
- Shortest route problems (SRPs) as TsPs

8 Network Flow Problems
- A more graphical approach
- The network flow algorithm

9 Assignment Problems
- LP Formulation
- The Hungarian Assignment Algorithm (HAA)

Example Q

The following table gives the time c_{ij} (in minutes) that member i ($i = 1, 2, \ldots, 5$) of a family would take to carry out chore j ($j = 1, 2, \ldots, 5$).

			Chore				
			1	2	3	4	5
			Wash up	Cut grass	Make dinner	Clean house	Wash car
Member	1	Charles	29	22	31	23	21
	2	Camilla	21	28	33	27	25
	3	William	21	22	39	29	24
	4	Kate	24	28	32	28	27
	5	Harry	24	24	31	28	25

If each member of the family is to do exactly one task, how should the jobs be assigned to minimise the total time spent?

Example Q (cont.)

$$\text{Let } x_{ij} = \begin{cases} 1 & \text{if person } i \text{ does job } j \quad (i, j = 1, 2, \ldots, 5) \\ 0 & \text{otherwise.} \end{cases}$$

The LP formulation of the problem is then as follows:

$$\textbf{Minimise } Z = \sum_{i=1}^{5} \sum_{j=1}^{5} c_{ij} x_{ij} \tag{47}$$

$$\text{subject to} \quad \sum_{j=1}^{5} x_{ij} = 1 \qquad (i = 1, 2, \ldots, 5) \tag{48}$$

$$\sum_{i=1}^{5} x_{ij} = 1 \qquad (j = 1, 2, \ldots, 5) \tag{49}$$

$$x_{ij} \geq 0 \qquad (i, j = 1, 2, \ldots, 5).$$

Example Q (cont.)

- Comparison with the equations on slide 165 shows that we have a TP with $m = n = 5, a_i = 1$ $(i = 1, 2, \ldots, 5)$, and $b_j = 1$ $(j = 1, 2, \ldots, 5)$.

- This TP will therefore have an integer-valued solution, and either (48) or (49) implies that

$$x_{ij} = 0 \text{ or } 1 \qquad (i, j = 1, 2, \ldots, 5). \tag{50}$$

- When read in conjunction with equation (50),

  the i^{th} of the constraints (48) then says that person i does exactly one job; and the j^{th} of the constraints (49) then says that chore j is done by exactly one person.

- Thus the equations (50) are the key to ensuring that the LP encapsulates the requirements of this AP, but, as we have seen, it is not necessary to incorporate them as explicit constraints.

- The usual non-negativity constraints of the TP suffice.

- In a general **balanced** AP, we have n people to assign to n jobs, and an $n \times n$ matrix $\mathbf{C} = (c_{ij})$, where $c_{ij} \geq 0$ is the cost of assigning person i ($i = 1, 2, \ldots, n$) to job j ($j = 1, 2, \ldots, n$). (In example Q, the costs are time spent.)

- As the above discussion suggests, it could be viewed as a TP and solved using the transportation algorithm.

- Due to the special structure of the AP, however, the HAA offers an easier alternative.

- **Note**: An unbalanced problem can be balanced by inserting in **C**

  - either row(s) of zeroes, corresponding to dummy personnel;
  - or column(s) of zeroes, corresponding to dummy job(s).

The HAA proceeds as follows:

Step 1

Calculate $\mathbf{C}^{(1)} = (c_{ij}^{(1)})$, where $c_{ij}^{(1)} = c_{ij} - \min_j c_{ij}$.
i.e. subtract the minimum element in the row from every element in the row.

Calculate $\mathbf{C}^{(2)} = (c_{ij}^{(2)})$, where $c_{ij}^{(2)} = c_{ij}^{(1)} - \min_i c_{ij}^{(1)}$.
i.e. subtract the minimum element in the column from every element in the column.

Example Q (cont.)

$$\mathbf{C}^{(1)} = \begin{pmatrix} 8 & 1 & 10 & 2 & 0 \\ 0 & 7 & 12 & 6 & 4 \\ 0 & 1 & 18 & 8 & 3 \\ 0 & 4 & 8 & 4 & 3 \\ 0 & 0 & 7 & 4 & 1 \end{pmatrix} \Rightarrow \mathbf{C}^{(2)} = \begin{pmatrix} 8 & 1 & 3 & 0 & 0 \\ 0 & 7 & 5 & 4 & 4 \\ 0 & 1 & 11 & 6 & 3 \\ 0 & 4 & 1 & 2 & 3 \\ 0 & 0 & 0 & 2 & 1 \end{pmatrix}$$

- On slide 184 we examined the effect on the OF of subtracting arbitrary constants u_i $(i = 1, 2, \ldots, m)$ and v_j $(j = 1, 2, \ldots, n)$ from the rows and columns respectively of the cost table of a TP.

- We saw that the problem was essentially unchanged, with the revised OF $\tilde{Z}$ differing from Z only by a constant.

- In an AP we have $a_i = 1$ $(i = 1, 2, \ldots, n)$ and $b_j = 1$ $(j = 1, 2, \ldots, n)$, so the result on slide 184 now says that $\tilde{Z}$ is given by

$$\tilde{Z} = Z - \sum_{1}^{n} u_i - \sum_{1}^{n} v_j$$

- In example Q, $\sum u_i = 111,$ $\sum v_j = 9,$ $\tilde{Z} = Z - 120$

Step 2

Seek a complete assignment with a total cost of zero under the current revised matrix. If one exists, stop.

- As the revised costs are non-negative, zero is clearly the smallest possible value of the revised OF. Hence **Step 2** describes an optimal assignment.
- Note that if an assignment with a total cost of zero exists, a lone zero in any $\left\{\begin{matrix}\text{row}\\ \text{column}\end{matrix}\right\}$ must be selected (mark it with a 'box'), and any other zeroes in the same $\left\{\begin{matrix}\text{column}\\ \text{row}\end{matrix}\right\}$ cannot then be used (so cross them out). If, at any stage, all remaining zeroes have at least one other zero in both their row and their column, make an arbitrary selection, and then find the best assignment possible.

Example Q (cont.)

Using subscripts to indicate the order in which the allocations are made, we have:

$$\begin{pmatrix} 8 & 1 & 3 & \boxed{0}_3 & \cancel{0} \\ \boxed{0}_1 & 7 & 5 & 4 & 4 \\ \cancel{0} & 1 & 11 & 6 & 3 \\ \cancel{0} & 4 & 1 & 2 & 3 \\ \cancel{0} & \boxed{0}_2 & \cancel{0} & 2 & 1 \end{pmatrix} \tag{51}$$

A complete assignment with total cost of zero is not yet possible. In such cases we have:

Step 3

Draw not more than $n - 1$ lines through some of the rows and columns so that *all* zeroes are covered.

In fact, to avoid unnecessary iterations, the minimum possible number of lines should be used.

Theorem 9 (König, 1916)

Let S consist of those elements of a matrix with a particular property. Then Minimum number of lines, drawn through rows or columns, needed to cover S = Maximum number of elements of S with no two lying in the same row or column.

(Proof omitted.)

Thus in our AP: Minimum number of lines = Number of “boxed” zeroes

The required lines can be found by:

(a) Having “boxed” as many zeroes as possible, mark all rows without assignments.

(b) Mark columns not already marked which have zeroes in marked rows.

(c) Mark rows not already marked which have assignments in marked columns.

(d) Repeat (b) and (c) until you can mark no more.

(e) Draw lines through all unmarked rows and marked columns.

Example Q (cont.)

For the matrix in (51) we get:

$$\begin{array}{cl} \begin{pmatrix} 8 & 1 & 3 & \boxed{0} & \cancel{0} \\ \boxed{0} & 7 & 5 & 4 & 4 \\ \cancel{0} & 1 & 11 & 6 & 3 \\ \cancel{0} & 4 & 1 & 2 & 3 \\ \cancel{0} & \boxed{0} & \cancel{0} & 2 & 1 \end{pmatrix} & \begin{array}{l} \\ \checkmark_3 \\ \checkmark_1 \\ \checkmark_1 \\ \\ \end{array} \\ \begin{array}{l} \quad\checkmark_2 \end{array} & \end{array} \tag{52}$$

For small n, however, it is easy to find the lines by inspection.

Step 4

Subtract the smallest uncovered element h from all uncovered elements.
Add the smallest uncovered element h to each element where 2 lines intersect.
Go to **Step 2**.

For matrix (52), $h = 1$, so we revise the matrix as follows before repeating Steps 2 and 3:

$$\begin{array}{cc}
\left(\begin{array}{ccccc}
9 & 1 & 3 & \boxed{0}_4 & \cancel{0} \\
\boxed{0}_1 & 6 & 4 & 3 & 3 \\
\cancel{0} & \boxed{0}_2 & 10 & 5 & 2 \\
\cancel{0} & 3 & \boxed{0}_3 & 1 & 2 \\
1 & \cancel{0} & \cancel{0} & 2 & 1
\end{array}\right)
&
\begin{array}{c}
\\ \checkmark_5 \\ \checkmark_3 \\ \checkmark_3 \\ \checkmark_1
\end{array}
\\
\begin{array}{ccccc}
\checkmark_4 & \checkmark_2 & \checkmark_2 & &
\end{array}
&
\end{array}$$

Example Q (cont.)

Again, $h = 1$. Revise the matrix as required by Step 4 and go to Step 2 to obtain:

$$\begin{pmatrix} 10 & 2 & 4 & \boxed{0}_{5} & \cancel{0} \\ \boxed{0}_{1} & 6 & 4 & 2 & 2 \\ \cancel{0} & \boxed{0}_{2} & 10 & 4 & 1 \\ \cancel{0} & 3 & \boxed{0}_{3} & \cancel{0} & 1 \\ 1 & \cancel{0} & \cancel{0} & 1 & \boxed{0}_{4} \end{pmatrix} \text{ i.e. } \left\{ \begin{array}{l} \text{Charles cleans house;} \\ \text{Camilla washes up;} \\ \text{William cuts grass;} \\ \text{Kate makes dinner;} \\ \text{Harry washes car} \end{array} \right.$$

is an optimal assignment.

Example Q (cont.)

The third assignment was arbitrarily chosen. Choosing the other remaining zero on row 4 yields an alternative optimal solution, for which the last three columns are:

$$\begin{pmatrix} 4 & 0 & \boxed{0}_5 \\ 4 & 2 & 2 \\ 10 & 4 & 1 \\ 0 & \boxed{0}_3 & 1 \\ \boxed{0}_4 & 1 & 0 \end{pmatrix} \qquad \text{i.e.} \begin{cases} \text{Charles washes car;} \\ \text{Kate cleans house;} \\ \text{Harry makes dinner.} \end{cases}$$

The corresponding minimal time can be found using the original table (slide 260). We obtain

$$\begin{cases} \text{for solution from slide 274:} & 23 + 21 + 22 + 32 + 25 = 123 \\ \text{for solution fromslide 275:} & 21 + 21 + 22 + 28 + 31 = 123 \end{cases}$$

Note: Step 4 is equivalent to “Subtract h from all uncovered rows. Add h to all covered column”.
Since adding h is equivalent to subtracting $-h$, the operations for Step 4 are of the same type as those in Step 1, and thus merely change the OF by a constant.

Example Q

In the 1st pass through Step 4, we subtracted 3 and added 1.

Net amount subtracted = 2

In the 2nd pass through Step 4, we subtracted 4 and added 3.

Net amount subtracted = 1

In Step 1, amount subtracted = $\underline{120}$

Total subtracted = $\underline{123}$

So this agrees with the minimum total time (as above).

Example Q (cont.)

The net amounts subtracted from each row and column are also of interest

	Wash up	Cut grass	Make dinner	Clean house	Wash car	Step 1	Step 4		Total
							1st pass	2nd pass	
Charles	29	22	31	23	21	21	0	0	21
Camilla	21	28	33	27	25	21	1	1	23
William	21	22	39	29	24	21	1	1	23
Kate	24	28	32	28	27	24	1	1	26
Harry	24	24	31	28	25	24	0	1	25
Step 1	0	0	7	2	0				
Step 4, 1st pass	-1	0	0	0	0				
Step 4, 2nd pass	-1	-1	-1	0	0				
Total	-2	-1	6	2	0				

Example Q (cont.)

If cell (i, j) got an assignment under the optimal solution, its entry in the final revised matrix was zero, and so the amounts subtracted for the i^{th} row and the j^{th} column totalled c_{ij}. The following tableau shows an optimal solution to the corresponding TP, matching the solution on slide 274 in its non-trivial parts, together with a viable set of values for the corresponding simplex multipliers.

						u_i
	29	22	31	23 1	21 0	21
	21 1	28	33	27	25	23
	21 0	22 1	39	29	24	23
	24 0	28	32 1	28 0	27	26
	24	24	31	28	25 1	25
v_j	-2	-1	6	2	0	

Example Q (cont.)

- So the optimal simplex multipliers u_i $(i = 1, 2, \ldots, 5)$ and v_j $(j = 1, 2, \ldots, 5)$ in the TP can be chosen to equal the final amounts subtracted in the AP.
- Moreover, as $a_i = 1$ $(i = 1, 2, \ldots, 5)$ and $b_j = 1$ $(j = 1, 2, \ldots, 5)$, the dual of this TP requires the maximisation of $W = \sum_1^5 u_i + \sum_1^5 v_j$ (cf. §6.4)
- Thus:

Optimal value for the primal = Optimal value for the dual

= total amount subtracted in the AP

Notes

(i) In some APs, the aim is to maximise the OF. As with TPs in general, (cf. note (i) of §6.5) the algorithm is then applied to the cost matrix with $(i, j)^{\text{th}}$ element $C - c_{ij}$, where $C = \max(c_{ij})$.

(ii) In an AP, if individual i $(i = 1, 2, \ldots, n)$ cannot do job j $(j = 1, 2, \ldots, n)$, set

$$c_{ij} = \begin{cases} M \text{ in a minimisation problem,} \\ -M \text{ in a maximisation problem,} \end{cases}$$

where M is a large positive number.

References for Chapter 9

author	title (ed.)	section
H.A. Taha	O.R. (8th ed.)	5.4
W.L. Winston	O.R. (4 th ed.)	7.5