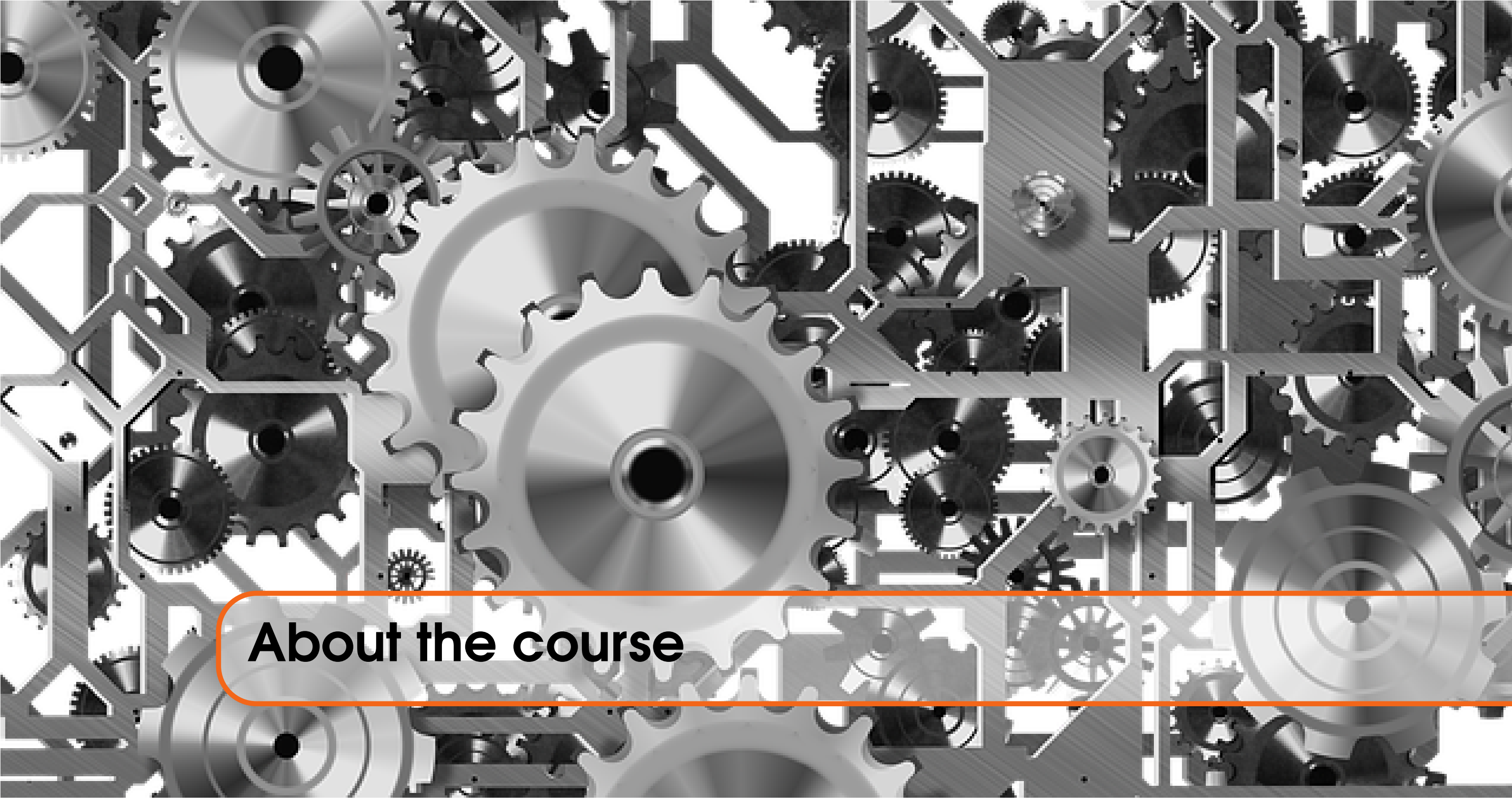
MATH3017: Mathematical Programming

"*Nothing happens in the universe that does not have a sense of either certain maximum or minimum*"– Leonhard Euler, Swiss Mathematician and Physicist, 1707–1783.



About the course

Unlike the title might suggest, this module is not about "programming" (many recent publications instead use the expression *Mathematical Optimization* to refer to the topic). But instead, we will be discussing mathematical problems that can be modeled as a minimization or maximization of a given function under some constraints. As basic example of problem, imagine a company which wants to minimize its cost or maximize its profit under a number of constraints, including budget limitations and salary costs for its employees.

You have probably had some taste of the topic in your first year calculus modules while studying the Fermat rule, as an application of the notion of *derivative*. However, the Fermat rule will not be used in this course as we will be working on optimization problems with constraints, something which is not taken into account in the Fermat rule. Additionally, we will be dealing with problems involving only affine linear functions; to be precise, our focus will be on *linear programming* or *linear optimization*. Problems involving nonlinear functions are studied in MATH3016 (Optimization). Students who have taken MATH2013 (Introduction to Operational Research) will already be familiar with some of the topics that will be covered in this module. In fact some of the concepts and methods from MATH2013 will be recalled and extended.

Having taken MATH2014 (Algorithms) could also be an advantage, especially for Chapter 4 (Network problems) and Chapter 5 (Integer programming). A combination of this module (MATH3017) and the aforementioned ones will allow you to build a strong body of knowledge in Operational Research, which is one the most attractive areas of mathematics for companies interested in analytical skills for optimal decision making. The course will also enable you to build critical knowledge which can be useful in understanding important topics in new and challenging applications of optimization, including emerging areas such as data science.

Focus here will be on the practical aspects of the concepts and methods, providing a broad taste of the underlying mathematics. Precisely, in Chapter 1, we will introduce the linear programming problem and its different forms. Attention will then be given to the graphical approach for solving problems with two variables. In Chapter 2, we will focus on one of the most powerful methods to solve linear programming problems, i.e., the simplex method. Next, we will discuss the duality theory of linear programming and its applications to some specific issues. Also, we will study the variation of linear programs under some specific types of perturbations on the problem data. Chapter 4 will be concerned with the applications of linear programming to problems that can be modeled using networks. There, we will discuss the implementation of a version of the simplex method tailored to this class of problem. Finally, in Chapter 5, we will study integer programming problems, i.e., linear programs where the variables are integers. There, most of the attention will be on applications, links with linear programs and one of the key techniques adapted to this class of problem, i.e., the branch and bound method.

The students taking this module will also be introduced to a software to solve optimization problems, namely `Xpress-IVE`. Two lectures will be devoted to this on March 10th and 17th. We will also have a tutorial each week (Fridays at 9:00AM) to discuss the exercises that will be given in the problem sheets. This will also be the ideal opportunity to ask any question on the material to be discussed in the lectures, though this can also be done during my office hour that will be on Wednesdays from 13:00 to 14:00. The assessment of the module will be 80% by the final exam, while 20% will be dedicated to a coursework that you will get in week 5. Further details on key dates is given on the next page.

The topics covered in the course have been studied in a large number of books and an important number of them can be found in the university library. As useful reference, I will suggest *F.S. Hillier & G.J. Lieberman, Introduction to Operations Research, 9th Ed. McGraw-Hill, 2010* (available in the Hartley library) and *R. Vandrbei, Linear Programming: Foundation and Extension, 4th Ed., Springer 2014* (can be accessed freely online).

How you are going to get feedback: There are various ways you could get feedback on your progress on the module. As key component, I strongly recommend you to work on the weekly problems and submit them so that they are marked and help us assess where you are struggling so that more emphasis is put on that during the tutorials. You will get the problem sheet one week before the tutorial - they will not be assessed as part of your final mark. But they will help you to practice all the components of the lectures and also prepare you towards your final exam. You are also encouraged to come to my office hours to discuss any particular aspect of lectures/material you might be struggling to understand. No appointment is needed to come to my office hour. But you can also contact me by email to arrange different meeting times to discuss your difficulties. Another key element of individual feedback will be based on your coursework assessment. Details on the submission and feedback release dates are also given in the table on the next page. Finally, I will also be able to provide some brief element of feedback by email (efforts will be made to reply to emails by the next working day after reception) if you have any quick questions.

Acknowledgements: Dr Houduo Qi is gratefully acknowledged for the development of previous drafts of the first three chapters and related material.

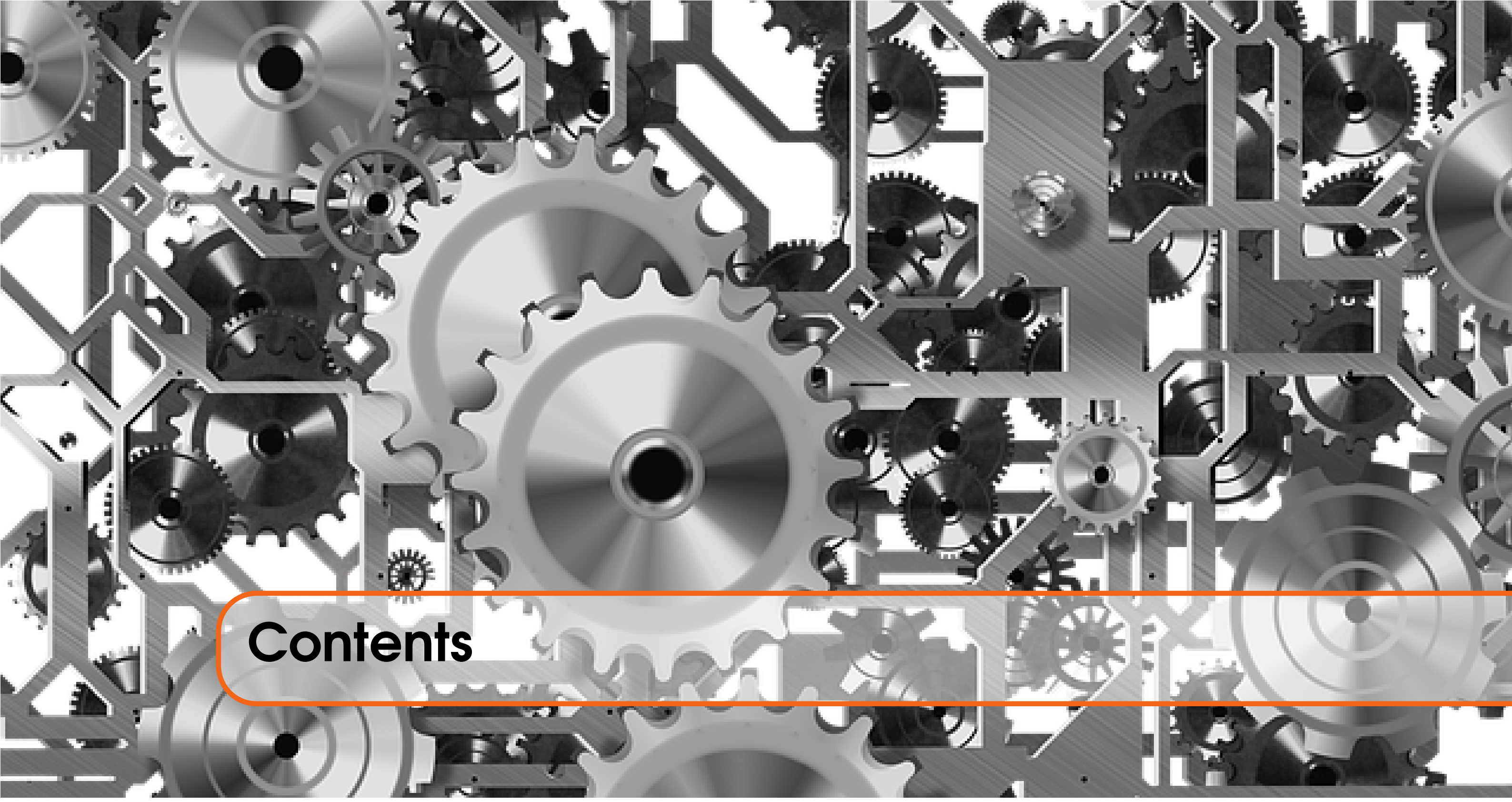
Instructor
Dr Alain Zemkoho
School of Mathematics
Building 54, Room 10027
`a.b.zemkoho@soton.ac.uk`

PhD Teaching Assistants[a]
Marton Benedek (`m.benedek@soton.ac.uk`)
Walton Coutinho (`w.p.coutinho@soton.ac.uk`)
Shenglong Zhou (`sz3g14@soton.ac.uk`)

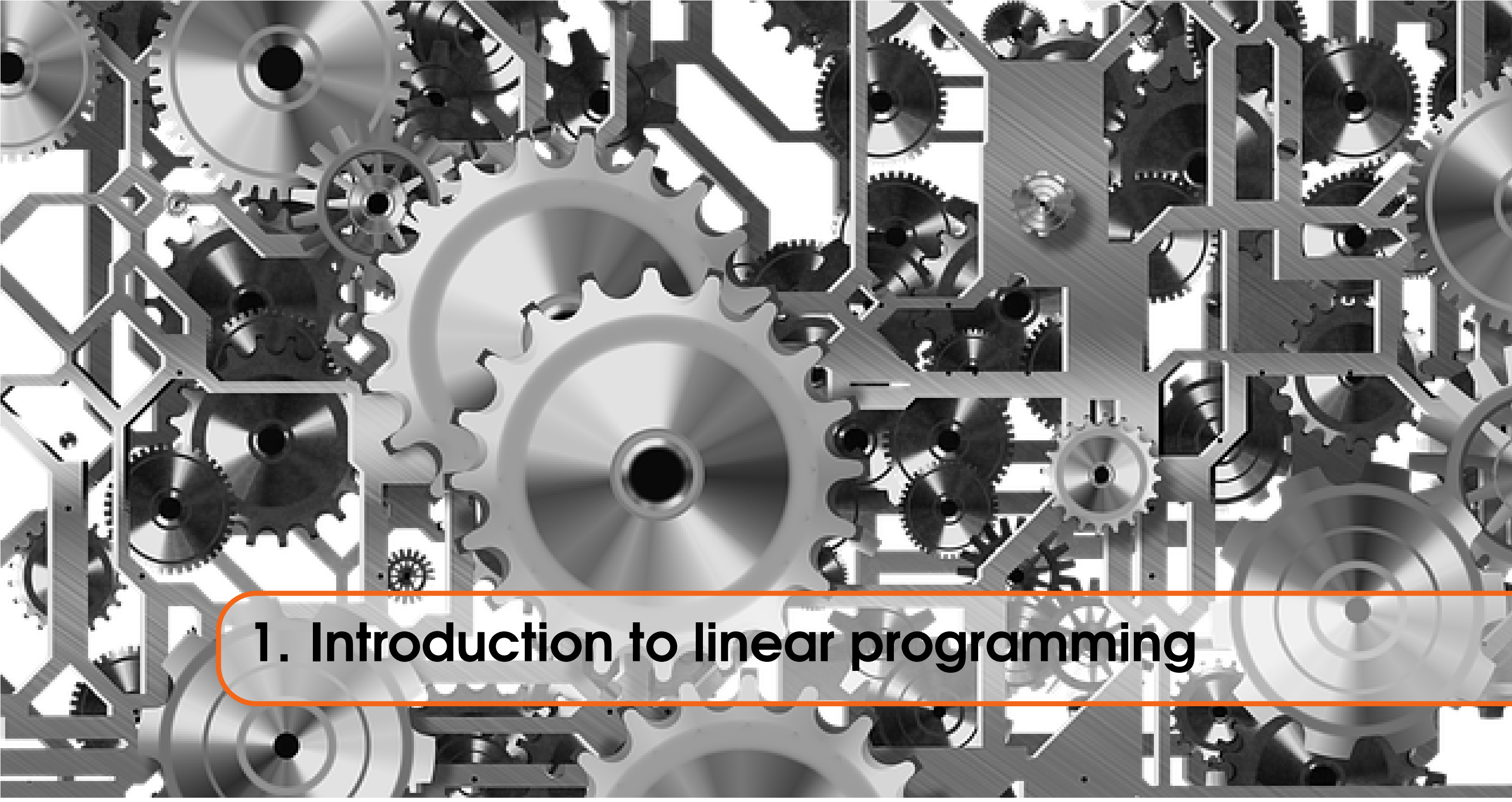
[a]They will assist in marking and providing feedback on the weekly problem sheets, and running the two computer labs in weeks 6 and 7.

MATH3017 Mathematical Programming 2016 - 17 full plan of activities [Last update: 20.01.2017]

Date	Start Time	End Time	Room	Session/Event	Details
Mondays	15:00	16:00	06/1081 (L/R B)	Lecture	See Lecture notes
Tuesdays	12:00	13:00	05/2017 (L/T J)	Lecture	See Lecture notes
Thursdays	14:00	15:00	58/1023 (L/R G)	Lecture	See Lecture notes
Fridays	09:00	10:00	07/3023 (L/R E)	Tutorial	See weekly problem sheet
Wednesdays	12:30	13:30	Student Centre (B56)	Office hour	Every week
Monday 27 February 2017	15:00		In class	Coursework handout	Copy will be posted on Blackboard
Friday 10 March 2017	12:00	14:00	25/1007	Computer Lab	On *Xpress - IVE*
Friday 17 March 2017	12:00	14:00	25/1007	Computer Lab	On *Xpress - IVE*
Friday 28 April 2017		15:00	Student Office	Coursework submission deadline	Submit a copy via Blackboard
By Friday 26 May 2017		16:00		and feedback release	Could be accessed via Blackboard



Contents



1. Introduction to linear programming

In this chapter, we start with a few interesting problems that can be modeled by linear programming. Many other examples can be found in standard text books on optimization; see, e.g., the ones recommended. We then give a formal definition of a linear programming problem (LP). We finish this chapter by introducing the graphical method for LPs with just two variables. It will motivate us to study the simplex method in next chapter.

1.1 Linear programming models

1.1.1 Modeling: examples

■ **Example 1.1** **(Cycle Trade)** A manufacturing company makes three products: Bicycles, mopeds, and child seats. For one period of production, the following data is available:

	Bicycles	Mopeds	Child seats	Resource
Unit profit (in £)	125	459	50	
Capital (in £)	320	1200	120	100 000
Storage (unit)	0.5	1	0.5	200

The problem is to decide how many units of each of the products should be produced in order to maximize the total profit. To obtain the model, we use the following steps:

Step 1: Set up the **variables**: Let x_1, x_2, and x_3 denote the number of bicycles, mopeds, and child seats to be produced, respectively.

Step 2: Set up the **objective function**: The profit, which is to be maximized, is

$$f(x_1,x_2,x_3) := 125x_1 + 459x_2 + 50x_3.$$

Step 3: Set up the **constraints**:

Constraint 1 (capital): The total cost should be under budget. That is

$$320x_1 + 1200x_2 + 120x_3 \leq 100000.$$

Constraint 2 (storage): The storage space should not be exceeded. That is

$$0.5x_1 + x_2 + 0.5x_3 \leq 200.$$

Constraint 3 (nonnegativity constraints):

$$x_1 \geq 0, \;\; x_2 \geq 0, \;\; x_3 \geq 0$$

Step 4: Set up the LP problem: Putting together the items above, we have the following model:

$$\begin{array}{ll} \text{maximize} & f = 125x_1 + 459x_2 + 50x_3 \\ \text{subject to} & 320x_1 + 1200x_2 + 120x_3 \leq 100000, \\ & 0.5x_1 + x_2 + 0.5x_3 \leq 200, \\ & x_1 \geq 0, \, x_2, \, x_3 \geq 0. \end{array}$$

■

■ **Example 1.2 (Diet Problem)** A nutritionist is planning a menu consisting of two main foods A and B. Each ounce of A contains 2 units of fat, 1 unit of carbohydrate, and 4 units of protein. Each ounce of B contains 3 units of fat, 3 units of carbohydrates, and 3 units of protein. The nutritionist wants the meal to provide at least 18 units of fat, at least 12 units of carbohydrate, and at least 24 units of protein. If an ounce of A costs 20 pence and an ounce of B costs 25 pence, how many ounces of each food should be served to minimize the cost of the meal yet satisfy the nutritionist's requirement? Proceeding as above, we successively have the following steps:

Step 1: Set up the **variables**: Let x_1 and x_2 denote the number of ounces of food A and B, which are to be served.

Step 2: Set up the **objective function**: The cost of the meal, which is to be minimized, is

$$f(x_1, x_2) := 20x_1 + 25x_2.$$

Step 3: Set up the **constraints**:

Constraint 1: The number of units of fat in the meal is no less than 18:

$$2x_1 + 3x_2 \geq 18.$$

Constraint 2 (carbohydrate constraint):

$$x_1 + 3x_2 \geq 12.$$

Constraint 3 (protein constraint):

$$4x_1 + 3x_2 \geq 24.$$

Constraint 4 (nonnegativity constraint):

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

Step 4: Set up the LP problem: Finally, we have

$$\begin{array}{ll} \text{minimize} & f = 20x_1 + 25x_2 \\ \text{subject to} & 2x_1 + 3x_2 \geq 18, \\ & x_1 + 3x_2 \geq 12, \\ & 4x_1 + 3x_2 \geq 24, \\ & x_1 \geq 0, \, x_2 \geq 0. \end{array}$$

■

1.1.2 Different forms of the problem

Here, we discuss different expressions of the linear programming problem.

The standard form

The following form of the linear programming problem is referred to as the *standard form*:

$$\begin{array}{lll} \text{maximize} & z = & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{subject to} & & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1, \\ & & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2, \\ & & \vdots \\ & & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m, \\ & & x_1 \geq 0,\ x_2 \geq 0,\ \ldots,\ x_n \geq 0. \end{array} \tag{1.1}$$

Common terminologies used in linear programming are as follows:

(a) The function being maximized, $c_1x_1 + \ldots + c_nx_n$, is called the *objective function*.
(b) The linear inequalities in the restrictions are referred to as *constraints*. For example, the first constraint is $a_{11}x_1 + \ldots + a_{1n}x_n \leq b_1$. There are in total m constraints in problem (1.1).
(c) The constraints $x_j \geq 0$, $j = 1, \ldots, n$ are often called *nonnegativity constraints*.
(d) The input constants, a_{ij}, b_i, c_j $(i = 1, \ldots, m,\ j = 1, \ldots, n)$, are often referred to as *parameters*. They are often called *coefficients* in the LP. For example, c_1 is the coefficient of variable x_1 in the objective function.
(e) $x_1, \ldots, x_n$ are called variables of the LP. n is the number of variables and m is the number of constraints not including the nonnegativity constraints.

Standard form in matrix-vector format

It is convenient to write linear programming problems in matrix notation. Let

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \text{ and } \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}.$$

We can write the standard linear programming problem as

$$\begin{array}{ll} \text{maximize} & z = \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{array} \tag{1.2}$$

Other forms

Linear programming problems may appear in different forms other than the standard form. But no matter what forms an LP may be formulated to, it can always be converted into the standard form. Common forms are the following:

(a) Minimizing rather than maximizing the objective function leads to the same optimal solution. Precisely, we have the following equivalence:

$$\min \mathbf{c}^T\mathbf{x} \iff \max -\mathbf{c}^T\mathbf{x}.$$

(b) Some constraints with a greater-than-or-equal-to inequality: Consider the inequality

$$a_{i1}x_1 + a_{12}x_2 + \cdots + a_{in}x_n \geq b_i$$

for some i. This can be converted to a less-than-or-equal-to inequality by multiplying both sides of the inequality by -1:

$$-a_{i1}x_1 - a_{12}x_2 - \cdots - a_{in}x_n \leq -b_i.$$

(c) Some constraints are in equation form: For a given i, the equality

$$a_{i1}x_1 + a_{12}x_2 + \cdots + a_{in}x_n = b_i$$

is equivalent to $\begin{cases} a_{i1}x_1 + a_{12}x_2 + \cdots + a_{in}x_n & \leq b_i, \\ -a_{i1}x_1 - a_{12}x_2 - \cdots - a_{in}x_n & \leq -b_i. \end{cases}$

(d) If the nonnegativity constraints for some decision variables are absent, we have

$$x_i \text{ unrestricted in sign} \iff x_i := x_{i1} - x_{i2}, \; x_{i1} \geq 0, \; x_{i2} \geq 0.$$

For example, we have

$$\begin{array}{ll} \min & z = 2x_1 - 3x_2 \\ \text{s.t.} & x_1 + 2x_2 \geq 10 \\ & x_1 \geq 0, \;\; x_2 \text{ is free} \end{array} \iff \begin{array}{ll} \max & z' = -2x_1 + 3(x_3 - x_4) \\ \text{s.t.} & -x_1 - 2(x_3 - x_4) \leq -10, \\ & x_1 \geq 0, \;\; x_3 \geq 0, \; x_4 \geq 0. \end{array}$$

Generalization: conic linear programming

Before we discuss the graphical method to solve the LP introduce above, we present an interesting extension of the problem. To proceed, consider let us examine the structure of the standard LP in (1.2). Then, we have the following mathematical elements:

(i) We have a finite-dimensional space $\Re^n$, the variable $\mathbf{x} \in \Re^n$ and the inner product $(\bullet)$ in $\Re^n$ defined for any $\mathbf{x}$ and $\mathbf{y}$ in $\Re^n$ by

$$\mathbf{x} \bullet \mathbf{y} = \mathbf{x}^T \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle.$$

The objective function is the inner product of $\mathbf{c}$ and $\mathbf{x}$.

(ii) Furthermore, we have the non-negative orthant:

$$\Re^n_+ := \{\mathbf{x} \in \Re^n \mid \mathbf{x} \geq 0\}.$$

The non-negative orthant is actually a closed convex cone satisfying the properties:

(P1) If $\mathbf{x} \in \Re^n_+$, then $\lambda \mathbf{x} \in \Re^n_+$ for any $\lambda \geq 0$.

(P2) If $\mathbf{x}$ and $\mathbf{y}$ both are in $\Re^n_+$, then

$$\mathbf{x} + \mathbf{y} \in \Re^n_+.$$

We now have a new formulation of the LP:

$$\begin{array}{ll} \text{maximize} & z = \mathbf{c} \bullet \mathbf{x} \\ \text{subject to} & \mathbf{a_i} \bullet \mathbf{x} \leq b_i, \;\; i = 1, \ldots, m, \\ & \mathbf{x} \in C = \Re^n_+, \end{array} \tag{1.3}$$

where $\mathbf{a_i}$ is the ith row vector of $\mathbf{A}$ and C is a closed convex set.

From what we have learnt over the past two years, we know that there are many spaces, many inner products and many types of convex cones. We can supply them to (1.3) and this would lead to different problems. But they are all linear!. Therefore (1.3) is often called the conic linear programming.

Replacing the cone C (which is a non-negative orthant in (1.3)) with a different type of cone, we can get a different class of optimization problem. For example, we have:

■ **Example 1.3** Let us consider the second-order cone or *ice-cream cone* in three dimensions:

$$\mathbf{Q}^3_+ := \left\{ (x_1, x_2, x_3) \in \Re^3 \;\middle|\; \sqrt{x_2^2 + x_3^2} \leq x_1 \right\}.$$

You can verify that $\mathbf{Q}^3_+$ is a closed convex cone. Then, we have the following example of second-order cone programming (SOCP) problem:

$$\begin{array}{ll} \max & 2x_1 + 3x_2 + x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 \leq 10, \\ & (x_1, x_2, x_3) \in \mathbf{Q}^3_+. \end{array}$$

■

1.2 Graphical method

Before properly going into the graphical method for two variables LPs, we first define the concepts of solution for problem (1.2).

1.2.1 Terminology for solutions in LP

Standard terminologies for solutions in LP are as follows:

(a) A *feasible solution* is a point satisfying *all* the constraints of the problem.
(b) The *feasible region* is the set of all feasible solutions.
(c) An *optimal solution* is a feasible solution that maximizes the objective function in (1.2).

Below are representations of the feasible regions of two examples of LPs:

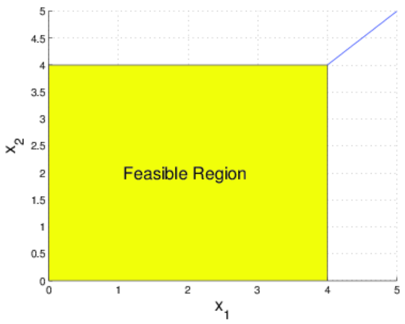


Figure 1.1: Feasible region (shaded) for the constraints: $x_1 \geq 0$, $x_2 \geq 0$, $x_1 \leq 4$ and $x_2 \leq 4$

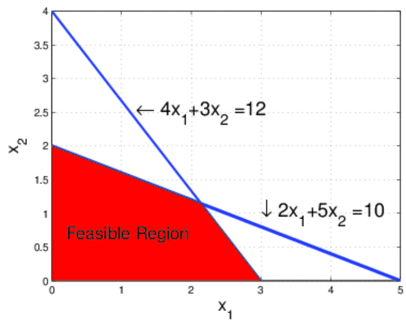


Figure 1.2: Feasible region (shaded) defined by $x_1 \geq 0$, $x_2 \geq 0$, $4x_1 + 3x_2 \leq 12$ and $2x_1 + 5x_2 \leq 10$

1.2.2 Graphical method

We use the following example to introduce the graphical method:

$$\begin{array}{ll} \max & z = 12x_1 + 15x_2 \\ \text{s.t.} & 4x_1 + 3x_2 \leq 12, \\ & 2x_1 + 5x_2 \leq 10, \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{array} \tag{1.4}$$

Steps of the graphical method for LPs with two variables:

S.1 Draw each constraint on a graph to decide the feasible region.
S.2 Draw the objective function on the graph.
S.3 Decide which corner point yields the largest (the smallest for minimization problem) objective function value. The corner point is the optimal solution.

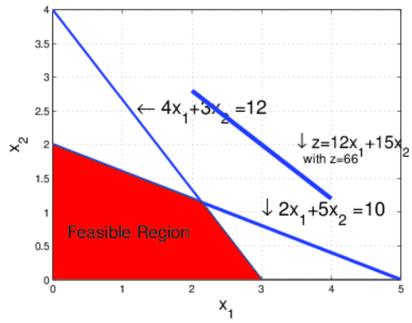


Figure 1.3: Graphical method: The optimal solution of problem (1.4) is $x_1 = 15/7$, $x_2 = 8/7$ with the optimal objective function value $z = 300/7$

An important theoretical result comes out of the graphical method is the following.

Theorem 1.2.1 For a linear programming problem, at least one corner point is the optimal solution provided that the LP has an optimal solution.

Our final result is about a way to construct new optimal solutions from existing ones.

Theorem 1.2.2 If $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ are two different optimal solutions of the LP (1.2), then any point of the form

$$\mathbf{x}^*(\rho) = \rho\mathbf{x}_1^* + (1-\rho)\mathbf{x}_2^* \quad \text{for} \quad 0 \le \rho \le 1,$$

is also an optimal solution of the problem.

Proof. Since $\mathbf{x}_i^*$ for $i = 1,2$ are optimal solutions, they must give the same objective function value

$$z^* = \mathbf{c}^T\mathbf{x}_i^* \quad \text{for} \quad i = 1,2.$$

They are also feasible:

$$\mathbf{x}_i^* \ge 0 \quad \text{and} \quad A\mathbf{x}_i^* \le \mathbf{b} \quad \text{for } i = 1,2.$$

We prove that for all $0 \le \rho \le 1$, the point $\mathbf{x}^*(\rho)$ is also feasible, i.e.,

$$\mathbf{x}^*(\rho) \ge 0 \quad \text{and} \quad A\mathbf{x}^*(\rho) = \rho A\mathbf{x}_1^* + (1-\rho)A\mathbf{x}_2^* \le \mathbf{b}$$

and give the same objective function value for all $0 \le \rho \le 1$, i.e.,

$$\mathbf{c}^T\mathbf{x}^*(\rho) = \rho\mathbf{c}^T\mathbf{x}_1^* + (1-\rho)\mathbf{c}^T\mathbf{x}_2^* = \rho z^* + (1-\rho)z^* = z^*.$$

■

1.3 Exercises

1. A general model for the diet problem (Example 1.2). Consider the the following elements:

n	The number of possible foods indexed by $j = 1,2,\ldots,n$
m	The number of nutritional categories indexed by $i = 1,2,\ldots,m$
x_j	The amount of food j to be included in the diet (measured in number of servings)
p_j	The cost of one serving of food j
b	The minimum daily requirement of nutrient i
A_{ij}	the amount of nutrient i contained in one serving of food j.

The diet problem is to determine a diet that achieved all the nutritional requirements of the individual while minimizing the total cost. Find a linear program for the above general model of the diet problem.

2. A firm produces three types of refined chemicals: A, B. and C. At least 4 tonnes of A, 2 tonnes of B and 1 tonne of C have to be produced each day. The inputs used are compounds X and Y. Each tonne of X yields $1/4$ tonnes of A, $1/4$ tonnes of B and $1/12$ tonnes of C. Each tonne of Y yields $1/2$ tonnes of A, $1/10$ tonnes of B and $1/12$ tonnes of C. Compound X costs £250 per tonne, compound Y costs £400 per tonne. The cost of processing each tonne of X and Y is £250 and £200 respectively.
   (a) Formulate the problem of minimizing the total daily cost as a linear programming problem.
   (b) Find the optimal solution graphically.
3. Convert the following linear programming to the standard form.

$$\begin{array}{ll} \min & z = 3x_1 + 4x_2 + 5x_3 \\ \text{s.t.} & x_1 + x_2 + x_3 = 10, \\ & 2x_1 + x_2 + x_3 \geq 1, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \text{ is unrestricted.} \end{array}$$

4. Decide whether the following linear programming problems are *feasible*. If so, find their optimal solutions.

$$(a)\left\{\begin{array}{ll} \max & 2x_1 + 4x_2 \\ \text{s.t.} & x_1 + x_2 \leq 1, \\ & x_1 + x_2 \geq 2, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}\right. \qquad (b)\left\{\begin{array}{ll} \min & y_1 - 2y_2 \\ \text{s.t.} & y_1 - y_2 \geq 2, \\ & y_1 - y_2 \geq 4, \\ & y_1 \geq 0,\ y_2 \geq 0. \end{array}\right.$$

$$(c)\left\{\begin{array}{ll} \max & x_1 - x_2 \\ \text{s.t.} & x_1 + x_2 \geq 1, \\ & 2x_1 - 3x_2 \geq 5, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}\right. \qquad (d)\left\{\begin{array}{ll} \min & -y_1 - 5y_2 \\ \text{s.t.} & -y_1 - 2y_2 \geq 1, \\ & -y_1 + 3y_2 \geq -1, \\ & y_1 \geq 0,\ y_2 \geq 0. \end{array}\right.$$

What conclusion can you draw from the solutions of the problem pair (a) and (b); and of the pair (c) and (d)?

5*. Let us consider the space of 2×2 symmetric matrices, denoted by $\mathscr{S}^2$:

$$\mathscr{S}^2 = \left\{ X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \mid X_{12} = X_{21} \right\}.$$

For example,

$$X = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$$

belongs to $\mathscr{S}^2$. We now consider all 2×2 symmetric matrices whose eigenvalues are non-negative. We collect all of such matrices in the set $\mathscr{S}^2_+$:

$$\mathscr{S}^2_+ = \left\{ X \in \mathscr{S}^2 \mid \text{all eigenvalues of } X \text{ are non-negtaive} \right\}.$$

We define an inner product in $\mathscr{S}^2$ by

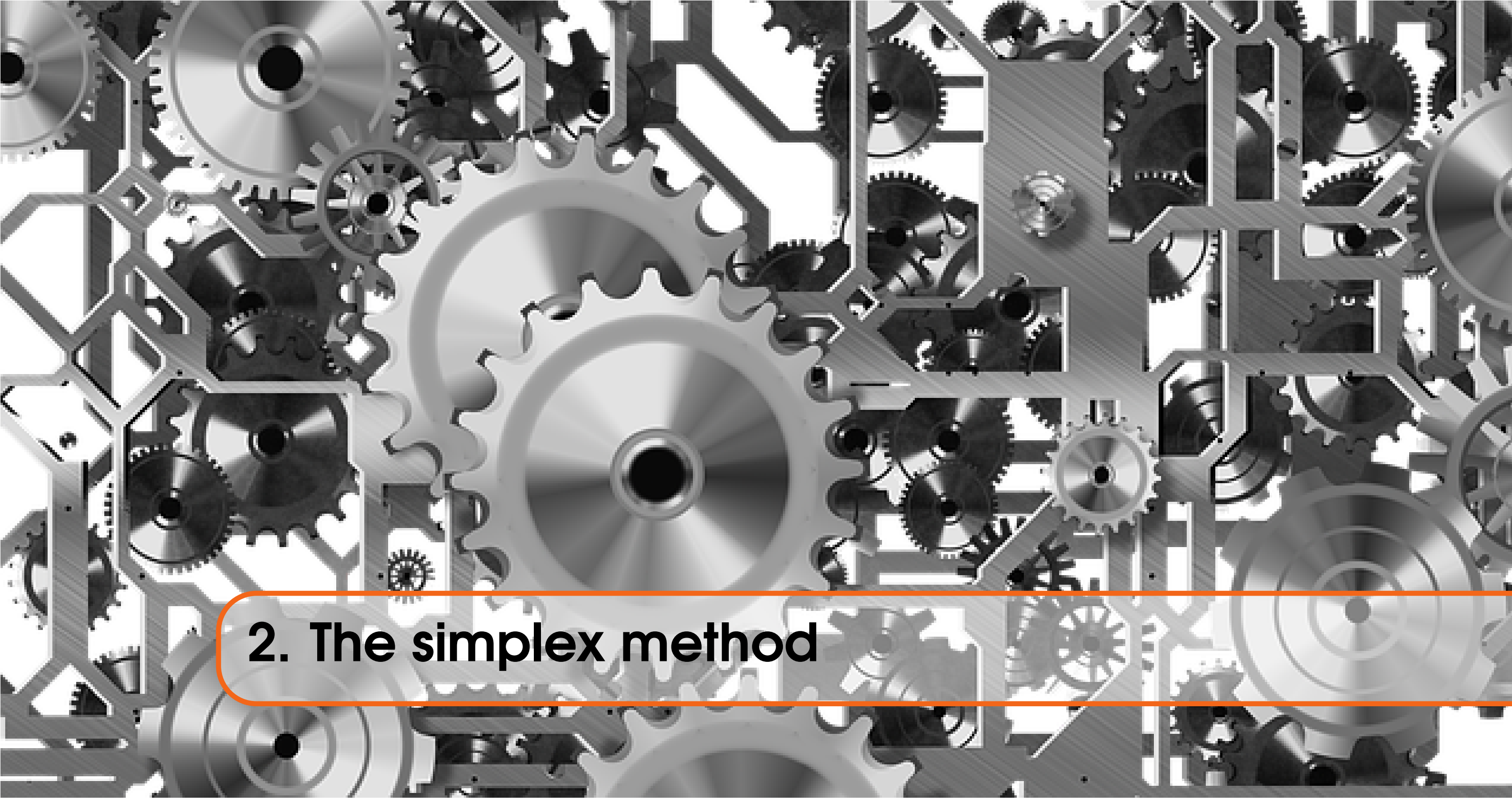
$$X \bullet Y = \text{Trace}(XY).$$

   (i) Prove that $\mathscr{S}^2_+$ is a closed convex cone.

(ii) Now consider the following conic linear programming:

$$\begin{aligned} \max \quad & X_{11} \\ \text{s.t.} \quad & I \bullet X \le 1, \\ & X \in \mathscr{S}_{+}^{2}, \end{aligned}$$

where I is the identity matrix in $\mathscr{S}^2$. Find the optimal solution of the above conic linear programming problem.



2. The simplex method

In this chapter, we introduce the simplex method in linear programming. The importance of the method has been well documented in most of textbooks and research papers in Operational Research. It was ranked one of the top ten methods in the last century by SIAM (the Society of Industrial and Applied Mathematics). It was one of the first numerical methods tested by the first generation digital computers in the 1950s. Two variants of the simplex method will be introduced. The first is the *standard simplex method* for the standard linear programming problem. The second called the *two-phase* simplex method handles the case where it is not possible to start the standard simplex method.

2.1 Under-determined system of linear equations

2.1.1 An example

We would like to solve the following system of linear equations:

$$\begin{cases} 2x_1 - 3x_2 + x_3 = 6, \\ 4x_1 + 5x_2 + x_4 = 20. \end{cases} \tag{2.1}$$

We solve this system for x_3 and x_4:

$$\begin{cases} x_3 = -2x_1 + 3x_2 + 6, \\ x_4 = -4x_1 - 5x_2 + 20. \end{cases} \tag{2.2}$$

In the setting of (2.2), the variables x_3 and x_4 are called **basic variables** and the variables are x_1 and x_2 **nonbasic variables**. A particular solution is

$$x_1 = x_2 = 0,\ x_3 = 6 \quad \text{and} \quad x_4 = 20.$$

That is, let the nonbasic variables be *zero* and the basic variables take the values of the corresponding constant terms.

Now we want to change the role of x_1 with x_3. That is, we want x_1 become basic variable and x_3 nonbasic. From the first equation of (2.2), we have

$$x_1 = \frac{3}{2}x_2 - \frac{1}{2}x_3 + 3. \tag{2.3}$$

Substituting it into the second equation of (2.2), we then have

$$x_4 = -11x_2 + 2x_3 + 8. \tag{2.4}$$

The equations in (2.2) can be put in the following tableau:

		x_1	x_2	rhs
x_3	=	-2	3	6
x_4	=	-4	-5	20

Equations (2.3) and (2.4) can be put into the following tableau:

		x_3	x_2	rhs
x_1	=	$-1/2$	$3/2$	3
x_4	=	2	-11	8

Later in this chapter, we introduce the simplex method based on similar representations of the LPs.

2.1.2 Geometric interpretation

Now let us return to the linear system (equations) considered in the last subsection while including the the nonnegativity constraints

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0. \tag{2.5}$$

The (2.1) is equivalent to the following system of inequalities

$$\begin{cases} 2x_1 - 3x_2 \leq 6, \\ 4x_1 + 5x_2 \leq 20. \end{cases} \tag{2.6}$$

We plot its feasible region in Figure 2.1 below.

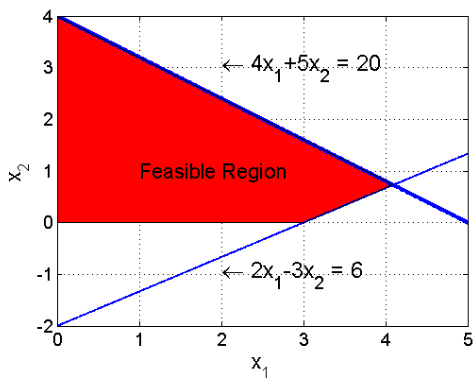


Figure 2.1: Feasible region of the system (2.6)

The first solution x with $x_1 = 0$, $x_2 = 0$, $x_3 = 6$, and $x_4 = 20$ corresponds to the corner point $(0,0)$ of the feasible region. The meaning of $x_3 = 6$ measures how far the constraint $2x_1 - 3x_2 = 6$ is from $(0,0)$. We say that x_3 is the slack value of the first constraint. Similarly, x_4 is the slack value of the second constraint.
The second solution x with $x_1 = 3$, $x_2 = 0$, $x_3 = 0$, and $x_4 = 8$ corresponds to the corner point $(3,0)$ of the feasible region. The slack value x_3 is 0, meaning the corner point is on the line of the first constraint.
Next, we introduce the Jordan exchange that will allow us to move from one corner point to the adjacent one.

2.2 Jordan exchange

2.2.1 General case*

Let us generalize the above procedure to the case where there are n independent variables $(x_1,\ldots,x_n)$ and m dependent variables $(y_1,\ldots,y_m)$, i.e.,

$$y_i = A_{i1}x_1 + \ldots + A_{in}x_n \quad \text{for} \quad i = 1,\ldots,m.$$

Then, we have the following tableau representation of the system:

$$\begin{array}{cc|ccccc|}
 & & x_1 & \cdots & x_s & \cdots & x_n \\ \hline
y_1 & = & A_{11} & \cdots & A_{1s} & \cdots & A_{1n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_r & = & A_{r1} & \cdots & A_{rs} & \cdots & A_{rn} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & = & A_{m1} & \cdots & A_{ms} & \cdots & A_{mn} \\ \hline
\end{array}$$

Suppose we want to exchange x_s with y_r. That is, the dependent variable y_r becomes independent, while x_s changes from being independent to being dependent. The process is carried out by the following three steps:

Step 1: Solve the rth equation

$$y_r = A_{r1}x_1 + \ldots + A_{rs}x_s + \ldots + A_{rn}x_n$$

for x_s in terms of y_r as well as x_1, ..., x_{s-1}, x_{s+1}, ..., x_n. Assuming $A_{rs} \neq 0$, this gives

$$x_s = \frac{1}{A_{rs}}y_r + \sum_{\substack{j=1\\ j\neq s}}^{n} \frac{-A_{rj}}{A_{rs}}x_j. \tag{2.7}$$

Step 2: Substitute (2.7) into all the remaining equations:

$$\begin{aligned}
y_i &= \sum_{\substack{j=1\\ j\neq s}}^{n} A_{ij}x_j + A_{is}\left(\frac{1}{A_{rs}}y_r + \sum_{\substack{j=1\\ j\neq s}}^{n} \frac{-A_{rj}}{A_{rs}}x_j\right) \\
&= \sum_{\substack{j=1\\ j\neq s}}^{n} \left(A_{ij} - \frac{A_{is}}{A_{rs}}\right)x_j + \frac{A_{is}}{A_{rs}}y_r.
\end{aligned} \tag{2.8}$$

Step 3: Write the new system in a new tableau form as follows

$$
\begin{array}{ccc}
 & & \begin{array}{ccccccc} x_1 & \cdots & x_{s-1} & y_r & x_{s+1} & \cdots & x_n \end{array} \\
\begin{array}{c} y_1 \\ \vdots \\ y_{r-1} \\ x_s \\ y_{r+1} \\ \vdots \\ y_m \end{array} &
\begin{array}{c} = \\ \vdots \\ = \\ = \\ = \\ \vdots \\ = \end{array} &
\left|\begin{array}{ccccccc}
B_{11} & \cdots & \cdots & B_{1s} & \cdots & \cdots & B_{1n} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
B_{r1} & \cdots & \cdots & B_{rs} & \cdots & \cdots & B_{rn} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
B_{m1} & \cdots & \cdots & B_{ms} & \cdots & \cdots & B_{mn}
\end{array}\right|
\end{array}
$$

with the following relations:

$$B_{rs} = \frac{1}{A_{rs}} \qquad \text{(pivot)}, \tag{2.9}$$

$$B_{rj} = \frac{-A_{rj}}{A_{rs}}, \ \forall\, j \neq s \qquad \text{(pivot row)}, \tag{2.10}$$

$$B_{is} = \frac{A_{is}}{A_{rs}}, \ \forall\, i \neq r \qquad \text{(pivot column)}, \tag{2.11}$$

and

$$B_{ij} = \left(A_{ij} - \frac{A_{is}}{A_{rs}} A_{rj}\right) = \left(A_{ij} - B_{is} A_{rj}\right), \ \forall\, i \neq r, \ j \neq s \qquad \text{(other elements)}. \tag{2.12}$$

2.2.2 A numerical example

Now we apply the Jordan exchange to the first tableau in of Subsection 2.1.1:

$$
\begin{array}{cc|cc|c|}
 & & x_1 & x_2 & \text{rhs} \\
\hline
x_3 & = & \boxed{-2} & 3 & 6 \\
x_4 & = & -4 & -5 & 20 \\
\hline
\end{array}
$$

We would like to exchange x_1 and x_3. That is, x_1 to become the basic variable and x_3 to become the non-basic variable. Therefore, the number at the intersection of the x_1 column and x_3 row is the pivotal number. We use a box to surround the number -2.

First, we swap the position between x_1 and x_3. In the new tableau, the pivotal number is replaced by $1/(-2)$ and the rest numbers in the pivotal row are replaced by

$$-\frac{\text{old number}}{\text{pivotal number}} = -\frac{\text{old number}}{-2}.$$

The rest of the numbers in the pivotal column are replaced by

$$\frac{\text{old number}}{\text{pivotal number}} = \frac{\text{old number}}{-2}.$$

We get a tableau like this:

$$\begin{array}{rcc|c|c|} & & x_3 & x_2 & \text{rhs} \\ \cline{3-5} x_1 & = & -1/2 & 3/2 & 3 \\ x_4 & = & 2 & x & y \\ \cline{3-5} \end{array}$$

There are two numbers to be updated, which are x and y. The formula to calculate x is

$$x = \alpha - \beta \times \gamma,$$

where

$$\begin{aligned} \alpha &= \text{The number corresponding to } x \text{ in the old tableau (i.e., } \alpha = -5), \\ \beta &= \text{The number in the new tableau at the intersection of the pivotal column} \\ &\quad \text{and the row containing } x \text{ (i.e., } \beta = 2), \\ \gamma &= \text{The number in the old tableau at the intersection of the pivotal row} \\ &\quad \text{and the column corresponding to } x \text{ (i.e., } \gamma = 3). \end{aligned}$$

Hence,

$$x = -5 - 2 \times 3 = -11.$$

Similarly, y can be calculated by

$$y = 20 - 2 \times 6 = 8.$$

This leads to the complete new tableau:

$$\begin{array}{rcc|c|c|} & & x_3 & x_2 & \text{rhs} \\ \cline{3-5} x_1 & = & -1/2 & 3/2 & 3 \\ x_4 & = & 2 & -11 & 8 \\ \cline{3-5} \end{array}$$

2.3 The simplex method

2.3.1 The phase II procedure

We start the description of the phase II simplex method with a few examples. As first example, we add an objective function to the linear systems made of (2.5) and (2.6):

■ Example 2.1

$$\begin{array}{ll} \max & z = 3x_1 + x_2 \\ \text{s.t.} & 2x_1 - 3x_2 \leq 6, \\ & 4x_1 + 5x_2 \leq 20, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

As before, we introduce two slack variables x_3 and x_4, each for one inequality constraint:

$$\begin{aligned} x_3 &= -2x_1 + 3x_2 + 6, \\ x_4 &= -4x_1 - 5x_2 + 20. \end{aligned} \tag{2.13}$$

Note that the slack values are always nonnegative (the bigger side minus the smaller side). With the two new variables introduced, the objective function becomes

$$z = 3x_1 + x_2 + 0x_3 + 0x_4. \tag{2.14}$$

We put (2.13) and (2.14) into a tableau:

Tableau 1

		x_1	x_2	rhs
x_3	=	$\boxed{-2}$	3	6
x_4	=	-4	-5	20
z	=	3	1	0

Let us recall the following about the variables:

- x_1 and x_2 are called *nonbasic variables* of this tableau (they are independent);
- x_3 and x_4 are called *basic variables* of this tableau (they are dependent).

We use the following rules to update to the next (new) tableau:

Rule-1: Choose the entering variable. Select the **largest positive number** in the z-row (excluding the right-hand side constant); the corresponding column variable (nonbasic variable) is the entering variable to the basics. For our example,

x_1 is the entering variable.

Rule-2: Choose the leaving variable. Consider the constraint rows only (not including z-row). Calculate all the ratios between each right-hand constant and its corresponding *negative number* in the column under the chosen entering variable x_1:

$$\frac{6}{-2} = -3 \leftarrow$$
$$\frac{20}{-4} = -5$$

The basic variable corresponding to the **largest ratio** is the leaving variable. In the case of our example,

x_3 is the leaving variable.

Rule-3: Choose the pivot element. The number at the intersection of the column under the entering variable and the row corresponding to the leaving variable is the pivot element.
The number is often frame-boxed. In this example, it is -2.

Rule-4: Perform one Jordan exchange at the chosen pivot element to get a new tableau:

Tableau 2

		x_3	x_2	rhs
x_1	=	$-1/2$	$3/2$	3
x_4	=	2	$\boxed{-11}$	8
z	=	$-3/2$	$11/2$	9

Feasible solution to this tableau:

$$x_1 = 3,\ x_2 = 0,\ x_3 = 0,\ x_4 = 8 \ \text{ with } \ z = 9.$$

Repeating the 4 rules above to this new tableau, we have

Entering variable: x_2;
Leaving variable: x_4.

The pivot element is frame-boxed. We do one Jordan exchange to get a new tableau:

Tableau 3

		x_3	x_4	rhs
x_1	=	$-5/22$	$-3/22$	$45/11$
x_2	=	$2/11$	$-1/11$	$8/11$
z	=	$-1/2$	$-1/2$	13

Basic feasible solution from this tableau:

$$x_1 = \frac{45}{11},\ x_2 = \frac{8}{11},\ x_3 = 0,\ x_4 = 0 \ \text{ with } \ z = 13.$$

There are no positive numbers under the basic variables in the z row in the above tableau. We cannot repeat the 4 rules above. So stop. An optimal solution is found, which is

$$x_1 = \frac{45}{11},\ x_2 = \frac{8}{11} \ \text{ with } \ z = 13.$$

■

Next, we apply the simplex method to solve the problem in Subsection 1.2.2, where it is solved using the graphical method.

■ **Example 2.2** We first start by recalling the problem:

$$\begin{array}{ll} \max & z = 12x_1 + 15x_2 \\ \text{s.t.} & 4x_1 + 3x_2 \leq 12, \\ & 2x_1 + 5x_2 \leq 10, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

We then introduce two slack variables x_3 and x_4, each for one inequality constraint:

$$\begin{aligned} x_3 &= -4x_1 - 3x_2 + 12, \\ x_4 &= -2x_1 - 5x_2 + 10. \end{aligned} \tag{2.15}$$

With the two new variables introduced, the objective function becomes

$$z = 12x_1 + 15x_2 + 0x_3 + 0x_4. \tag{2.16}$$

We put (2.15) and (2.16) into a tableau:

Tableau 1

		x_1	x_2	rhs
x_3	=	-4	-3	12
x_4	=	-2	$\boxed{-5}$	10
z	=	12	15	0

R Similarly to the previous example, we have

- x_1 and x_2 as *nonbasic variables* of this tableau (they are independent);
- x_3 and x_4 as *basic variables* of this tableau (they are dependent).

Repeating the 4 rules above to this new tableau, it holds that

Entering variable: x_2,
Leaving variable: x_4.

The pivot element is frame-boxed. Do one Jordan exchange to get a new tableau:
Tableau 2

		x_1	x_4	rhs
x_3	=	$\boxed{-14/5}$	$3/5$	6
x_2	=	$-2/5$	$-1/5$	2
z	=	6	-3	30

Feasible solution to this tableau:

$$x_1 = 0,\ x_2 = 2,\ x_3 = 6,\ x_4 = 0,\ \text{with } z = 30.$$

Repeating the 4 rules above to this new tableau:

Entering variable: x_1
Leaving variable: x_3

The pivot element is frame-boxed. Do one Jordan exchange to get a new tableau:

Tableau 3

		x_3	x_4	rhs
x_1	=	$-5/14$	$3/14$	$15/7$
x_2	=	$1/7$	$-2/7$	$8/7$
z	=	$-15/7$	$-12/7$	$300/7$

Feasible solution from this tableau:

$$x_1 = \frac{15}{7},\ x_2 = \frac{8}{7},\ x_3 = 0,\ x_4 = 0,\ \text{with } z = \frac{300}{7}.$$

There are no positive numbers in the z row in the above tableau. We cannot repeat the 4 rules above. So stop. An optimal solution is found, which is

$$x_1 = \frac{15}{7},\ x_2 = \frac{8}{7} \ \text{with}\ z = \frac{300}{7}.$$

■

Before we move on, we give one more example that has three variables.

■ **Example 2.3** Consider the LP with 3 variables:

$$\begin{array}{ll} \max & z = 60x_1 + 30x_2 + 20x_3 \\ \text{s. t.} & 8x_1 + 6x_2 + x_3 \le 48, \\ & 4x_1 + 2x_2 + 1.5x_3 \le 20, \\ & 2x_1 + 1.5x_2 + 0.5x_3 \le 8, \\ & x_2 \le 5, \\ & x_1, x_2, x_3 \ge 0. \end{array}$$

In the following, we leave out some details (e.g., we omit the process of introducing *slack variables*, applying the 4 rules to get a new tableau). Hence, we just list the tableaus we need to find the optimal solution. To proceed, we start with the first tableau:

Tableau 1

		x_1	x_2	x_3	rhs
x_4	=	-8	-6	-1	48
x_5	=	-4	-2	-1.5	20
x_6	=	$\boxed{-2}$	-1.5	-0.5	8
x_7	=	0	-1	0	5
z	=	60	30	20	0

Tableau 2

		x_6	x_2	x_3	rhs
x_4	=	4	0	1	16
x_5	=	2	1	$\boxed{-0.5}$	4
x_1	=	-0.5	-0.75	-0.25	4
x_7	=	0	-1	0	5
z	=	-30	-15	5	240

Tableau 3

		x_6	x_2	x_5	rhs
x_4	=	8	2	-2	24
x_3	=	4	2	-2	8
x_1	=	$-3/2$	$-5/4$	0.5	2
x_7	=	0	-1	0	5
z	=	-10	-5	-10	280

The optimal solution is the point

$$(x_1, x_2, x_3) = (2, 0, 8) \quad \text{with} \quad z = 280.$$

■

Unbounded Case

The above three examples show that the simplex method terminates at an optimal solution. However, there is a possibility that there exist no *negative ratios* found in Rule 2. Consequently, the simplex method will not be able to proceed. When this happens, we can safely claim that the problem is unbounded from above. We actually can get more valuable information than just claiming the problem is unbounded. Let us take a look at the following example.

■ **Example 2.4** Show that the following LP is unbounded from above, and find vectors u and v such that $u + \lambda v$ is feasible for all $\lambda \geq 0$. Find a feasible point of this form with objective value 98.

$$\begin{array}{ll} \max & z = 2x_1 + 3x_2 - x_3 \\ \text{s.t.} & -x_1 - x_2 - x_3 \leq 3, \\ & x_1 - x_2 + x_3 \leq 4, \\ & -x_1 + x_2 + 2x_3 \leq 1, \\ & x_1,\ x_2,\ x_3 \geq 0. \end{array}$$

Solution: Let us introduce three slack variables x_4, x_5, and x_6 by

$$\begin{aligned} x_4 &= x_1 + x_2 + x_3 + 3 \\ x_5 &= -x_1 + x_2 - x_3 + 4 \\ x_6 &= x_1 - x_2 - 2x_3 + 1. \end{aligned}$$

Tableau 1

		x_1	x_2	x_3	rhs
x_4	=	1	1	1	3
x_5	=	-1	1	-1	4
x_6	=	1	$\boxed{-1}$	-2	1
z	=	2	3	-1	0

Tableau 2

		x_1	x_6	x_3	rhs
x_4	=	2	-1	-1	4
x_5	=	0	-1	-3	5
x_2	=	1	-1	-2	1
z	=	5	-3	-7	3

According to Rule-1, x_1 should be the entering variable. But we cannot find negative numbers in the x_1 column. So we cannot apply Rule-2. Tableau 2 gives a set of feasible points:

$$x_3 = x_6 = 0,\ x_1 = \lambda, \quad \text{and} \quad \begin{cases} x_4(\lambda) = 2\lambda + 4, \\ x_5(\lambda) = 5, \\ x_2(\lambda) = \lambda + 1, \end{cases} \quad \text{with} \quad \lambda \geq 0.$$

For the original three variables, we have

$$x(\lambda) = \begin{bmatrix} \lambda \\ \lambda+1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

So by setting

$$u = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad v = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix},$$

we obtain the direction of unboundedness in the specified form. From Tableau 2, we also have

$$z = 5\lambda + 3.$$

Therefore, the value $z = 98$ is obtained by setting $\lambda = 19$. The corresponding value of x is

$$x = u + 19v = \begin{bmatrix} 19 \\ 20 \\ 0 \end{bmatrix}.$$

■

Identify all optimal solutions

The Simplex method described above is able to find an optimal solution or to identify that the problem is unbounded. Our next question is to identify the whole solution set if the problem has multiple solutions. As before, we use an example to illustrate the procedure.

■ **Example 2.5** Find all the solutions of the following linear program:

$$\begin{array}{ll} \max & z = 4x_1 + 5x_2 \\ \text{s.t.} & 2x_1 - 3x_2 \le 6, \\ & 4x_1 + 5x_2 \le 20, \\ & x_1,\ x_2 \ge 0. \end{array}$$

Solution: Let us introduce two slack variables x_3 and x_4 by

$$x_3 = -2x_1 + 3x_2 + 2,$$
$$x_4 = -4x_1 - 5x_2 + 20.$$

Tableau 1

		x_1	x_2	rhs
x_3	=	-2	3	6
x_4	=	-4	$\boxed{-5}$	20
z	=	4	5	0

Tableau 2

		x_1	x_4	rhs
x_3	=	$-22/5$	$-3/5$	18
x_2	=	$-4/5$	$\boxed{\text{-1/5}}$	4
z	=	0	-1	20

This is an optimal tableau (why?). It tells a system of linear equations:

$$\begin{array}{rcrcrcr} x_3 & = & -22/5x_1 & - & 3/5x_4 & + & 18, \\ x_2 & = & -4/5x_1 & - & 1/5x_4 & + & 4, \\ z & = & 0x_1 & - & x_4 & + & 20. \end{array}$$

The optimal solution is

$$x_1 = 0,\ x_2 = 4,\ x_3 = 18,\ x_4 = 0 \quad \text{with} \quad z = 20.$$

There is more information from the optimal tableau. For example, if we let $x_4 = 0$ and let x_1 to increase, the objective function value does not change at all because the contribution of x_1 to the objective is 0 (look at the objective function in the optimal tableau).
But we cannot increase x_1 arbitrarily large. We need to ensure $x_2 \geq 0$ and $x_3 \geq 0$. We then have (note that $x_4 = 0$):

$$0 \leq x_2 = -4/5x_1 + 4 \implies x_1 \leq 5 \quad \text{and}$$

$$0 \leq x_3 = -22/5x_1 + 18 \implies x_1 \leq \frac{45}{11}.$$

Hence, we conclude that if x_1 is in the range

$$0 \leq x_1 \leq \frac{45}{11},$$

x_2 and x_3 both are nonnegative. The objective value is $z = 20$, which is optimal.
The optimal solutions form a set:

$$\left\{ (x_1, x_2) \,\middle|\, 0 \leq x_1 \leq \frac{45}{11},\ x_2 = -4/5x_1 + 4 \right\}.$$

■

2.3.2 The phase I procedure

Any tableau in the Phase II procedure is called a *feasible tableau* in the sense that we can read a special type of feasible point of the linear programming problem. However, it is not always the case that we can easily start from a feasible tableau. Take a look at the following example.

■ **Example 2.6** Consider the problem

$$\begin{array}{ll} \max & z = 3x_1 + x_2 \\ \text{s.t.} & 2x_1 - 3x_2 \geq 6, \\ & 4x_1 + 5x_2 \leq 20, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

Like before, we introduce slack variables

$$\begin{array}{rl} x_3 = & 2x_1 - 3x_2 - 6, \\ x_4 = & -4x_1 - 5x_2 + 20. \end{array}$$

If we let $x_1 = x_2 = 0$, then $x_3 = -6$ and $x_4 = 20$ (x_1 violates the nonnegativity requirement for the slack variables). One way to remedy this problem is to add a big (positive) number (denoted by x_0) to -6 to make it positive, leading to

$$x_3 = 2x_1 - 3x_2 + x_0 - 6.$$

Now treat $x_0 \geq 0$ as a new variable and solve the following auxiliary problem:

$$\begin{array}{ll} \max & z_0 = -x_0 \\ \text{s.t.} & x_3 = 2x_1 - 3x_2 + x_0 - 6, \\ & x_4 = -4x_1 - 5x_2 + 20, \\ & x_i \geq 0, \ i = 0,1,2,3,4. \end{array} \tag{2.17}$$

We would like to solve the auxiliary problem (2.17) by the simplex method (Phase-II):

Tableau 0

		x_1	x_2	x_0	rhs
x_3	=	2	-3	$\boxed{1}$	-6
x_4	=	-4	-5	0	20
z	=	3	1	0	0
z_0	=	0	0	-1	0

Special pivot in the phase I procedure:
- **(Rule-1′)** Choose x_0 as the entering variable.
- **(Rule-2′)** Choose the **most negative** number in the `rhs` column. The corresponding row variable is the leaving variable.
- **(Rule-3′)** The number at the intersection is the pivot element.
- **(Rule-4′)** Perform Jordan exchange at the pivot element.

This leads to the following tableau:

Tableau 1

		x_1	x_2	x_3	rhs
x_0	=	$\boxed{-2}$	3	1	6
x_4	=	-4	-5	0	20
z	=	3	1	0	0
z_0	=	2	-3	-1	-6

Apply phase II simplex method to Tableau 1 with respect to the objective function z_0 and we get

Tableau 2

		x_0	x_2	x_3	rhs
x_1	=	$-1/2$	$3/2$	$1/2$	3
x_4	=	2	-11	-2	8
z	=	$-3/2$	$11/2$	$3/2$	9
z_0	=	-1	0	0	0

The following information is revealed by Tableau 2:

- All coefficients in z_0 are non-positive. Phase II procedure should stop and the auxiliary problem is solved.
- x_0 is a non-basic variable, which means x_0 takes value 0.
- The objective value of z_0 is also 0.

Strike out the x_0 column as well as the z_0 row to get a new tableau:

Tableau 3

		x_2	x_3	rhs
x_1	=	$3/2$	$1/2$	3
x_4	=	$\boxed{-11}$	-2	8
z	=	$11/2$	$3/2$	9

The procedure up to here is called *phase I procedure*. Tableau 3 provides an initial BFS for the original problem. Continue to Phase-II procedure with the objective z. This tableau is not optimal, we need to carry out the phase II simplex method:

Tableau 4: x_2 entering and x_4 leaving:

		x_4	x_3	rhs
x_1	=	$\boxed{-3/22}$	$5/22$	$45/11$
x_2	=	$-1/11$	$\boxed{-2/11}$	$8/11$
z	=	$-1/2$	$1/2$	13

Tableau 5: x_2 leaving and x_3 entering:

		x_4	x_2	rhs
x_1	=	$-1/4$	$-5/4$	5
x_3	=	$-1/2$	$-11/2$	4
z	=	$-3/4$	$-11/4$	15

All variable coefficients in z-row are non-negative. Stop. Phase-II is complete. We have found an optimal solution to the original problem:

$$x_1 = 5,\ x_2 = 0 \ \text{ with } \ z = 14.$$

■

Phase I and phase II combined together is called *two-phase simplex method* and is capable of solving any type of linear programming problems.

The following remarks are very useful in understanding the two phase simplex method:

(R1) If the right-hand-side of the original problem is not all positive (i.e., there is a negative constant), then we have to use phase I simplex method to find an initial feasible tableau for the original problem. Otherwise, phase II simplex method is sufficient for the original problem.

(R2) When the Phase I simplex method is needed. A new problem is constructed and this problem is called an *auxiliary* problem.

(R3) A special pivot has to be executed at the beginning of the phase I simplex method.

(R4) If at the final tableau of phase I method, the optimal value $z_0 < 0$, then the original problem has no feasible solution. If $z_0 = 0$, then strike out the x_0 column and z_0 row to get a new tableau, which is a feasible tableau for the original problem. From this new tableau, phase II simplex method can be used.

(R5) The procedure that uses phase I and phase II simplex methods is called *two-phase simplex method*.

2.3.3 Finite termination and complexity

Basic feasible solutions

The concept is best introduced with the *canonical* form of LP:

$$\begin{aligned} \max \quad & z = c^T x \\ \text{s.t.} \quad & \mathscr{A} x = b, \; x \geq 0, \end{aligned} \tag{2.18}$$

where $c, x \in \Re^{\ell}$, $b \in \Re^m$, and $\mathscr{A} \in \Re^{m \times \ell}$ (we often let $n := \ell - m$).

Definition 2.3.1 Given $\mathscr{A} \in \Re^{m \times \ell}$, consider the column submatrix $\mathscr{A}_B$ for some subset $B \subseteq \{1, 2, \ldots, \ell\}$. If $\mathscr{A}_B$ is invertible, it is called a *basis matrix*.

■ **Example 2.7** Consider LP in the standard form:

$$\begin{aligned} \max \quad & z = \bar{c}^T x \\ \text{s.t.} \quad & Ax \leq b, \; x \geq 0, \end{aligned} \tag{2.19}$$

where $A \in \Re^{m \times n}$, $x, \bar{c} \in \Re^n$, and $b \in \Re^m$.
By introducing slack variables $x_{n+1}, \ldots, x_{n+m}$, we have

$$A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + I \begin{bmatrix} x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} = b.$$

Let the parameters c and $\mathscr{A}$ be respectively defined by

$$c = \begin{bmatrix} \bar{c} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathscr{A} = [A, \; I].$$

It follows that the standard form can be rewritten in the canonical form. An obvious basis matrix is when $B = \{n+1, \ldots, n+m\}$ and $\mathscr{A}_B = I$, which is obviously nonsingular. ■

Suppose we have a basis matrix $\mathscr{A}_B$. Then the equation in problem (2.18) can be written as

$$\mathscr{A}_N x_N + \mathscr{A}_B x_B = b,$$

where $N := \{1, 2, \ldots, \ell\} \setminus B$ and x_N is the subvector of x consisting those elements belonging to B. This equation gives

$$x_B = -\mathscr{A}_B^{-1} \mathscr{A}_N x_N + \mathscr{A}_B^{-1} b. \tag{2.20}$$

If we let $x_N = 0$ in (2.20), we arrive at a particular solution:

$$\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} (\mathscr{A}_B)^{-1} b \\ 0 \end{pmatrix}. \tag{2.21}$$

Definition 2.3.2 Any point of the form (2.21) is called a **basic solution** of problem (2.18). If If $(\mathscr{A}_B)^{-1}b \geq 0$, it is called a **basic feasible solution** (BFS).

Note that the columns of $\mathscr{A}$ corresponding to the basic variables x_B (i.e., the columns in $\mathscr{A}_B$) are *linearly independent*.

■ **Example 2.8** Consider a linear programming of (2.18) with

$$\mathscr{A} = \begin{bmatrix} -1 & 2 & 2 \\ 0 & 1 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}.$$

Find all the basic solutions as well as all the basic feasible solutions.
Answer: Possible basis matrices are

$$\begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 2 & 2 \\ 1 & 0 \end{bmatrix}.$$

The corresponding basic solutions are

$$\begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ 0 \\ \frac{3}{2} \end{bmatrix},$$

of which the latter is the only basic feasible solution. ■

The simplex method moves from one BFS to another BFS. This gives us the following results.

Theorem 2.3.1 If the simplex method fails to terminate, then it must cycle.

Proof. There are at most $n!/(m!(n-m)!)$ of choosing m basic variables from n. Thus, if the Simplex method fails to terminate, then some basis appears in two different iterations. The above analysis shows that these tableaus are identical, and cycling occurs. ■

Theorem 2.3.2 For any linear programming problem,
(a) if it has no optimal solution, it is either infeasible or unbounded,
(b) if it is feasible, then it has a basic feasible solution,
(c) if it has an optimal solution, then it has a basic optimal solution.

Proof. Phase I of the simplex method either finds that the problem is *infeasible*, or it produce a basic feasible solution. Phase-II either finds that the problem is unbounded, or it produces a basic optimal solution. ■

Beale's example and Bland's rule

We once again consider the standard problem (2.19). Suppose we have a BFS x, which must have the form

$$x = \begin{bmatrix} x_B \\ 0 \end{bmatrix}.$$

Recall for a BFS, $x_B \geq 0$. If, furthermore, each of the components of x_B is positive (i.e, $x_B > 0$) x is said **nondegenerate**. Otherwise it is said **degenerate**. An LP is **nondegenerate** if all of its basic feasible solutions are nondegenerate.

■ **Example 2.9** (Degenerate example)

$$\begin{array}{ll} \max & z = 3x_1 + 5x_2 \\ \text{s.t.} & 2x_1 + 3x_2 \le 6, \\ & x_1 + 2x_2 \le 4, \\ & x_1, x_2 \ge 0. \end{array}$$

BASIC	BASIC FS	COMMENT	z
x_1, x_2	$(0,2,0,0)$	*Degenerate*	10
x_1, x_4	$(3,0,0,1)$	*Nondegenerate*	9
x_2, x_3	$(0,2,0,0)$	*Degenerate*	10
x_2, x_4	$(0,2,0,0)$	*Degenerate*	10
x_3, x_4	$(0,0,6,4)$	*Nondegenerate*	0

■

For nondegenerate LPs, no cycling can occur. But for degenerate problems, it can happen.

■ **Example 2.10** (Beale's example)

$$\begin{array}{ll} \max & z = 3/4x_1 - 150x_2 + 1/50x_3 - 6x_4 \\ \text{s.t.} & 1/4x_1 - 60x_2 - 1/25x_3 + 9x_4 \le 0, \\ & 1/2x_1 - 90x_2 - 1/50x_3 + 3x_4 \le 0, \\ & x_3 \le 1, \\ & x_1, x_2, x_3, x_4 \ge 0. \end{array}$$

Pivot column: Based on the largest coefficient in the z-row

Pivot row: When there are equal ratios, select the one nearest the top of the tableau.

The following bases are generated:

$$\begin{array}{c|c|c|c|c|c|c} x_5 & x_1 & x_1 & x_3 & x_3 & x_5 & x_5 \\ x_6 & x_6 & x_2 & x_2 & x_4 & x_4 & x_6 \\ x_7 & x_7 & x_7 & x_7 & x_7 & x_7 & x_7 \end{array}$$

Smallest subscript pivoting rule (Bland):

(R_b-1) Pivot column: From all the variables whose corresponding coefficients are positive in the z-row, choose the one with the smallest subscript.

(R_b-2) Pivot row: When there are equal ratios, select the one containing a basic variable with the smallest subscript.

(R_b-3) The number at the intersection is the pivot element.

(R_b-4) Perform Jordan exchange at the pivot element.

Bland shows that this method avoids cycling. ■

Klee-Minty's example and Smale's average complexity

The well-known *Klee-Minty example* is a problem with a large number of iterations.

■ **Example 2.11** (Klee-Minty example) Consider the problem

$$\begin{array}{ll} \max & z = \sum_{j=1}^{n} 10^{n-j} x_j \\ \text{s.t.} & 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \le 100^{i-1},\ i = 1, \dots, n, \\ & x_j \ge 0,\ \ j = 1, \dots, n. \end{array}$$

The optimal solution is:

$$x_1 = \ldots = x_{n-1} = 0,\; x_n = 100^{n-1}.$$

Using the "text book" pivoting rule, 2^n tableaus are generated (2^n corresponds to an exponential complexity). For $n = 3$, the problem above becomes

$$\begin{array}{ll} \max & z = 100x_1 + 10x_2 + x_3 \\ \text{s.t.} & x_1 \le 1, \\ & 20x_1 + x_2 \le 100, \\ & 200x_1 + 20x_2 + x_3 \le 10000, \\ & x_1,\, x_2,\, x_3 \ge 0. \end{array}$$

Base in successive tableaus are (they are total $2^3 = 8$ iterations):

$$\begin{array}{c|c|c|c|c|c|c|c} x_4 & x_1 & x_1 & x_4 & x_4 & x_1 & x_1 & x_4 \\ x_5 & x_5 & x_2 & x_2 & x_2 & x_2 & x_4 & x_5 \\ x_6 & x_6 & x_6 & x_3 & x_3 & x_3 & x_3 & x_3 \end{array}$$

This Klee-Minty problem shows that in the worst case the Simplex method takes as many as 2^n iterations (exponential time) to find a solution. However, extensive computation shows that it works well on average. So one has to answer the question: Why does the Simplex method work efficiently in practice? It took a Fields Medalist's effort to answer this question. Steve Smale in his 1983 paper (S. Smale, *On the average number of steps of the Simplex method for linear programming*. Mathematical Programming 27 (1983), pp. 241–262) shows that for a *random* problem with m constraints the number of the iterations grows in proportion to n, the number of variables. ■

2.4 Exercises

1. Consider the problem

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.t.} & Ax \le b,\; x \ge 0, \end{array}$$

with the parameters respectively defined by

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & -2 \\ -1 & 1 \end{bmatrix},\; b = \begin{bmatrix} 5 \\ 9 \\ 0 \\ 3 \end{bmatrix} \text{ and } c = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

   (i) Solve the problem graphically.
   (ii) Solve the problem by the Simplex method. In addition, trace the path that the Simplex method takes on your figure.

2. Solve the following problem by the Simplex method.

$$\begin{array}{ll} \max & z = 6x_1 + 5x_2 - x_3 + 4x_4 \\ \text{s.t.} & 3x_1 + 2x_2 - 3x_3 + x_4 \le 120, \\ & 3x_1 + 3x_2 + x_3 + 3x_4 \le 180, \\ & x_j \ge 0,\; j = 1,2,3,4. \end{array}$$

3. Show that the following LP is unbounded from above, and find vectors u and v such that $u+\lambda v$ is feasible for all $\lambda \geq 0$. Find a feasible point of this form with objective value 120:

$$\begin{array}{ll} \max & z = 3x_1 + 2x_2 - x_3 \\ \text{s.t.} & -x_1 - x_2 - x_3 \leq 3, \\ & -x_1 + x_2 + x_3 \leq 4, \\ & x_1 - x_2 + 2x_3 \leq 1, \\ & x_1,\ x_2,\ x_3 \geq 0. \end{array}$$

4. Consider the following problem:

$$\begin{array}{ll} \max & z = x_1 + x_2 + x_3 + x_4, \\ \text{s.t.} & x_1 + x_2 \leq 10, \\ & x_3 + x_4 \leq 15, \\ & x_j \geq 0,\ \ j = 1,2,3,4. \end{array}$$

Use the simplex method to find all the optimal solutions.

5. Solve the following linear programming problem by the two phase simplex method:

$$\begin{array}{ll} \max & z = 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 3, \\ & 4x_1 + 3x_2 \geq 3, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

6. Solve the following linear programming problem by the two phase simplex method:

$$\begin{array}{ll} \max & z = 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 3, \\ & 3x_1 + x_2 = 6, \\ & 4x_1 + 3x_2 \geq 3, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

3. Duality theory and sensitivity analysis

In this chapter, we study the dual problem in linear programming. It is a common question that we often ask ourselves when we study a new problem: Does it have a dual form? If yes, we would like to know whether it has a number of appealing properties. For instance, the dual problem of a linear programming problem is still a linear programming problem; the dual form of the dual problem is the original problem; the solution of the dual problem has some relationship with that of the original problem. In order to describe those relationships, we often call the original problem the primal problem so that we are sure which problem we are referring to. The first task is to investigate what form the dual problem should take given that the primal problem is known.

3.1 The dual problem

3.1.1 Mathematical derivation

Let us start with an example:

$$\begin{array}{ll} \max & z = x_1 + 2x_2 + x_3 \\ \text{s.t.} & 2x_1 + x_2 - x_3 \leq 2, \\ & 4x_1 + x_2 + x_3 \leq 6, \\ & x_1, x_2, x_3 \geq 0. \end{array} \tag{3.1}$$

We denote the optimal objective function value by z^*. Then it is easy to see that any feasible solution provides a *lower* bound for z^*. For example, the point x with

$$x_1 = 0,\ x_2 = 2,\ x_3 = 0$$

gives a lower bound

$$x_1 + 2x_2 + x_3 = 4 \leq z^*.$$

We can derive an upper bound. Using the second constraint,

$$\begin{aligned} z^* &= \max\{x_1 + 2x_2 + x_3\}, \\ &\leq \max\{2(4x_1 + x_2 + x_3)\} \leq 2 \times 6 = 12. \end{aligned}$$

So, 12 is an upper bound. We may obtain a tighter bound using both constraints

$$\begin{aligned} z^* &= \max\{x_1+2x_2+x_3\}, \\ &\le \max\left\{\frac{1}{2}(2x_1+x_2-x_3)+\frac{3}{2}(4x_1+x_2+x_3)\right\}, \\ &\le \frac{1}{2}\times 2+\frac{3}{2}\times 6 = 10. \end{aligned}$$

Thus, an improved bound is 10. To obtain the *best* upper bound using this technique, using multipliers y_1 and y_2 of the two constraints, where

$$y_1 \ge 0, \quad y_2 \ge 0. \tag{3.2}$$

The chain of (in)equalities

$$\begin{aligned} z^* &= \max\{x_1+2x_2+x_3\} \\ &\le \max\{y_1(2x_1+x_2-x_3)+y_2(4x_1+x_2+x_3)\} \\ &\le 2y_1+6y_2 \end{aligned}$$

is valid provided that

$$\begin{aligned} 2y_1+4y_2 &\ge 1, && (3.3) \\ y_1+y_2 &\ge 2, && (3.4) \\ -y_1+y_2 &\ge 1. && (3.5) \end{aligned}$$

The best upper bound is obtained if we solve

$$\begin{array}{ll} \min & 2y_1+6y_2 \\ \text{s.t.} & (3.2),\ (3.3),\ (3.4),\ (3.5). \end{array} \tag{3.6}$$

Table 3.1: Relations between primal and dual problems

Problem (3.1)	Relationships	Problem (3.6)
Maximization	$\Longrightarrow$	Minimization
Objective coefficients	$\Longrightarrow$	Right hand side constants
Right hand side constants	$\Longrightarrow$	Objective coefficients
Matrix of constraint coefficients	$\Longrightarrow$	Transpose of Matrix of constraint coefficients
$\le$ constraint type	$\Longrightarrow$	Non-negativity of variables
Non-negativity of variables	$\Longrightarrow$	$\ge$ constraint type

Problem (3.1) and problem (3.6) have striking physical symmetry, which can be clearly seen if we put them into the matrix-vector format. To proceed, let

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 4 & 1 & 1 \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \ \mathbf{c} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

Then problem (3.1) and problem (3.6) respectively have the forms

$$
(\text{Primal problem}) \left\{ \begin{array}{ll} \max & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \right. \quad \text{and} \quad \left. \begin{array}{ll} \min & \omega = \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq 0 \end{array} \right\} \quad (\text{Dual problem}).
$$

Let us now consider the general problems, where

$$
A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.
$$

Definition 3.1.1 We call the linear programming problem

$$
\begin{array}{ll} \max & z = \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \tag{3.7}
$$

the **primal** problem and call the following problem obtained by following the rules in Table 3.1

$$
\begin{array}{ll} \min & \omega = \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq 0 \end{array} \tag{3.8}
$$

the **dual** problem.

We note that the number of the dual variables (i.e., m) equals the number of the inequality constraints in the primal problem. We have the following interesting result.

Theorem 3.1.1 The dual of the dual is the primal.

Proof. Considering (3.7), its dual (3.8) can be put in the max form

$$
\begin{array}{ll} \max & -\mathbf{b}^T\mathbf{y} \\ \text{s.t.} & -A^T\mathbf{y} \leq -\mathbf{c}, \\ & \mathbf{y} \geq 0. \end{array}
$$

The dual of the latter problem according to Table 3.1 is

$$
\begin{array}{ll} \min & -\mathbf{c}^T\mathbf{x} \\ \text{s.t.} & -A\mathbf{x} \geq -\mathbf{b}, \\ & \mathbf{x} \geq 0, \end{array}
$$

which is equivalent to

$$
\begin{array}{ll} \max & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{array}
$$

The is exactly the primal problem. ■

In view of Theorem 3.1.1, the words *primal* and *dual* are used interchangeably. This justifies the following definition.

Definition 3.1.2 Consider the following two problems

$$\left\{\begin{array}{ll} \max & z=\mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x}\leq\mathbf{b} \\ & \mathbf{x}\geq 0 \end{array}\right. \quad \Bigg| \quad \left.\begin{array}{ll} \min & \omega=\mathbf{b}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y}\geq\mathbf{c} \\ & \mathbf{y}\geq 0 \end{array}\right\}.$$

We call one of the two problems the **primal** problem and the other its **dual problem**. By convention, the original problem is often called the primal problem.

In this course, we always call the maximization problem the primal problem.

3.1.2 Dealing with equality constraints

We may have noticed that the rules in Table 3.1 only apply to the standard linear programming problem. It cannot be applied directly to problems with equality constraints. But we already know that through simple operations, any form of the linear programming problem can be put into the standard form. We work with an example to see what form the dual problem appears when the equality constraints are present.

■ **Example 3.1** Suppose we have the following LP with equality constraints:

$$\begin{array}{ll} \max & z=\mathbf{c}^T\mathbf{x} \\ \text{s.t.} & A_1\mathbf{x}\leq\mathbf{b}, \\ & A_2\mathbf{x}=\bar{\mathbf{b}}, \\ & \mathbf{x}\geq 0, \end{array}$$

with $A_1\in\Re^{m_1\times n}$, $A_2\in\Re^{m_2\times n}$, $\mathbf{b}\in\Re^{m_1}$ and $\bar{\mathbf{b}}\in\Re^{m_2}$. We want to find its dual problem.
To proceed, we can write the constraints in the equivalent form:

$$\begin{bmatrix} A_1 \\ A_2 \\ -A_2 \end{bmatrix} x\leq\begin{bmatrix} \mathbf{b} \\ \bar{\mathbf{b}} \\ -\bar{\mathbf{b}} \end{bmatrix}.$$

Let $\mathbf{u},\mathbf{v},\bar{\mathbf{v}}$ denote vectors of dual variables. According to the rules in Table 3.1 the dual problem is

$$\begin{array}{ll} \min & \omega=\mathbf{b}^T\mathbf{u}+\bar{\mathbf{b}}\mathbf{v}-\bar{\mathbf{b}}\bar{\mathbf{v}}, \\ \text{s.t.} & A_1^T\mathbf{u}+A_2^T\mathbf{v}-A_2^T\bar{\mathbf{v}}\geq\mathbf{c}, \\ & \mathbf{u},\mathbf{v},\bar{\mathbf{v}}\geq 0. \end{array}$$

Let $\mathbf{w}=\mathbf{v}-\bar{\mathbf{v}}$. Then the dual problem can be written as

$$\begin{array}{ll} \min & \omega=\mathbf{b}^T\mathbf{u}+\bar{\mathbf{b}}\mathbf{w} \\ \text{s.t.} & A_1^T\mathbf{u}+A_2^T\mathbf{w}\geq\mathbf{c}, \\ & \mathbf{u}\geq 0. \end{array}$$

■

We can observe that the dual variable for the equality constraint is *free*. Therefore, we have a new rule concerning the equality constraints:

Table 3.2: Relations between primal and dual problems

Primal problem	Relationships	Dual problem
$=$ constraint type	$\Longrightarrow$	Free variables

■ **Example 3.2** The dual of the problem

$$\begin{array}{ll} \max & z = 2x_1 + x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 3, \\ & 3x_1 + x_2 = 6, \\ & 4x_1 + 3x_2 \geq 3, \quad \longrightarrow \quad -4x_1 - 3x_2 \leq -3, \\ & x_1 \geq 0,\ x_2 \geq 0, \end{array}$$

is

$$\begin{array}{ll} \min & \omega = 3y_1 + 6y_2 - 3y_3 \\ \text{s.t.} & y_1 + 3y_2 - 4y_3 \geq 2, \\ & 2y_1 + y_2 - 3y_3 \geq 1, \\ & y_1 \geq 0,\ y_3 \geq 0. \end{array}$$

Note that y_2 is a free variable (no restrictions). ■

3.2 Links with the primal problem and applications

In this section, we discuss the relationships between the primal and dual problems, and highlight some applications on the simplex method and other properties.

3.2.1 From primal to dual optimal solution

Let us consider the following example.

■ **Example 3.3** Solve the following linear program using the simplex method:

$$\begin{array}{ll} \max & z = x_1 + 2x_2 + x_3 \\ \text{s.t.} & 2x_1 + x_2 - x_3 \leq 2, \\ & 4x_1 + x_2 + x_3 \leq 6, \\ & x_1,\ x_2,\ x_3 \geq 0. \end{array}$$

We would like to formulate the dual of this problem and read off an optimal solution of the dual problem from the final tableau.
The dual of this problem is

$$\begin{array}{ll} \min & \omega = 2y_1 + 6y_2 \\ \text{s.t.} & 2y_1 + 4y_2 \geq 1, \\ & y_1 + y_2 \geq 2, \\ & -y_1 + y_2 \geq 1, \\ & y_1,\ y_2 \geq 0. \end{array}$$

This is a standard LP. After introducing slack variables, we have our initial feasible tableau, which also represents the dual problem after introducing the dual slack variables.

Tableau 1

			$-y_3 =$ x_1	$-y_4 =$ x_2	$-y_5 =$ x_3	$w =$ rhs
y_1	x_4	=	-2	$\boxed{-1}$	1	2
y_2	x_5	=	-4	-1	-1	6
rhs	z	=	1	2	1	0

Tableau 2

			$-y_3 =$ x_1	$-y_1 =$ x_4	$-y_5 =$ x_3	$w =$ rhs
y_4	x_2	=	-2	-1	1	2
y_2	x_5	=	-2	1	$\boxed{-2}$	4
rhs	z	=	-3	-2	3	4

Tableau 3 (Optimal tableau)

			$-y_3 =$ x_1	$-y_1 =$ x_4	$-y_2 =$ x_5	$w =$ rhs
y_4	x_2	=	-3	$-1/2$	$-1/2$	4
y_5	x_3	=	-1	$1/2$	$-1/2$	2
rhs	z	=	-6	$-1/2$	$-3/2$	10

Therefore, an optimal solution for the primal problem is:

$$x_1 = 0,\ x_2 = 4,\ x_3 = 2 \quad \text{with } z^* = 10.$$

The dual optimal solution can be read from the final tableau:

$$y_1 = 1/2 \quad \text{and} \quad y_2 = 3/2.$$

The resulting dual objective value is

$$\omega = 2 \times \frac{1}{2} + 6 \times \frac{3}{2} = 10 = z^*.$$

The weak duality theorem stated in the next section ensures that we read the dual optimal solution correctly. ■

We now demonstrate why the objective function in the final tableau contains valuable solution information for the dual problem. We would like to discuss it on general terms.
Let us consider the general primal problem

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.t.} & Ax \leq b, \\ & x \geq 0, \end{array}$$

with the matrix $A \in \Re^{m \times n}$. We assume (without loss of any generality) that we have $b \geq 0$. The initial feasible tableau of the problem is:

Tableau-P

$$
\begin{array}{rc|cccc|c|}
 & & x_1 & x_2 & \cdots & x_n & \text{rhs} \\
\cline{3-7}
x_{n+1} & = & -A_{11} & -A_{12} & \cdots & -A_{1n} & b_1 \\
\vdots & = & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_{n+m} & = & -A_{m1} & -A_{m2} & \cdots & -A_{mn} & b_m \\
\cline{3-7}
z & = & c_1 & c_2 & \cdots & c_n & 0 \\
\cline{3-7}
\end{array}
$$

Now consider the dual problem

$$
\begin{array}{ll}
\min & \omega = b^T y \\
\text{s.t.} & -A^T y \leq -c, \\
 & y \geq 0.
\end{array}
$$

This problem can be put into a tableau:

Tableau-D

$$
\begin{array}{r|cccc|c|}
 & y_{m+1} = & y_{m+2} = & \cdots & y_{m+n} = & \omega = \\
\cline{2-6}
y_1 & A_{11} & A_{21} & \cdots & A_{m1} & b_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & A_{1n} & A_{2n} & \cdots & A_{mn} & b_m \\
\cline{2-6}
\text{rhs} & -c_1 & -c_2 & \cdots & -c_n & 0 \\
\cline{2-6}
\end{array}
$$

Tableau-P and Tableau-D can be put into one tableau.

Tableau-U

$$
\begin{array}{rrc|cccc|c|}
 & & & -y_{m+1} = & -y_{m+2} = & \cdots & -y_{m+n} = & \omega = \\
 & & & x_1 & x_2 & \cdots & x_n & \text{rhs} \\
\cline{4-8}
y_1 & x_{n+1} & = & -A_{11} & -A_{12} & \cdots & -A_{1n} & b_1 \\
\vdots & \vdots & = & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & x_{n+m} & = & -A_{m1} & -A_{m2} & \cdots & -A_{mn} & b_m \\
\cline{4-8}
\text{rhs} & z & = & c_1 & c_2 & \cdots & c_n & 0 \\
\cline{4-8}
\end{array}
$$

We note the correspondence of the primal and dual variables in this Tableau-U.

$$
\text{(primal variables)} \quad \left\{ \begin{array}{ccc} x_1 & \leftrightarrow & y_{m+1} \\ x_2 & \leftrightarrow & y_{m+2} \\ \vdots & \vdots & \vdots \\ x_n & \leftrightarrow & y_{m+n} \end{array} \right\} \quad \text{(dual slack variables)} \tag{3.9}
$$

and

$$
\text{(primal slack variables)} \quad \left\{ \begin{array}{ccc} x_{n+1} & \leftrightarrow & y_1 \\ x_{n+2} & \leftrightarrow & y_2 \\ \vdots & \vdots & \vdots \\ x_{n+m} & \leftrightarrow & y_m \end{array} \right\} \quad \text{(dual variables)} \tag{3.10}
$$

After a sequence of Jordan exchanges from Tableau-P (exchanges of basic and nonbasic variables) we arrive at the following tableau:

Tableau-O (Optimal tableau)

$$
\begin{array}{ccc|c|c|}
 & & & -y_{\hat{B}} = & \omega = \\
 & & & x_N & \text{rhs} \\
\hline
y_{\hat{N}} & x_B & = & H & h \\
\hline
\text{rhs} & z & = & p & \alpha \\
\hline
\end{array}
$$

It follows from this optimal tableau (for the primal problem) that x with

$$x_B = h \geq 0,\ x_N = 0,$$

is an optimal solution and $p \leq 0$, $z = \alpha$.

It is also obvious that x with

$$y_{\hat{B}} = -p \geq 0,\ y_{\hat{N}} = 0 \tag{3.11}$$

is a **feasible** solution to the dual problem. Moreover, the objective function value is

$$\omega = \alpha.$$

By the weak duality theorem stated below, we know that (3.11) is an optimal solution of the dual problem. The final hurdles is to find the indexes of $\hat{B}$ by

$$\hat{B} \ \leftrightarrow\ N \text{ (through the relationship (3.9) and (3.10)).}$$

The technique developed above also leads to the so-called *Dual Simplex Method*, which we describe below.

3.2.2 Dual simplex method

Let us consider the primal-dual pair

$$
\text{(Primal)} \quad \left\{ \begin{array}{ll} \max & z = c^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{array} \right. \quad \Bigg| \quad \left. \begin{array}{lr} \min & \omega = b^T y \\ \text{s.t.} & -A^T y \leq -c \\ & y \geq 0 \end{array} \right\} \quad \text{(Dual)}
$$

Adding primal slack variables $x_{n+1}, \ldots, x_{n+m}$ and dual slack variables $y_{m+1}, \ldots, y_{m+n}$, we obtain:

Tableau-U

$$
\begin{array}{ccc|cccc|c|}
 & & & -y_{m+1} = & -y_{m+2} = & \cdots & -y_{m+n} = & \omega = \\
 & & & x_1 & x_2 & \cdots & x_n & \text{rhs} \\
\hline
y_1 & x_{n+1} & = & -A_{11} & -A_{12} & \cdots & -A_{1n} & b_1 \\
\vdots & \vdots & = & \vdots & \vdots & \vdots & \vdots & \vdots \\
y_m & x_{n+m} & = & -A_{m1} & -A_{m2} & \cdots & -A_{mn} & b_m \\
\hline
\text{rhs} & z & = & c_1 & c_2 & \cdots & c_n & 0 \\
\hline
\end{array}
$$

This tableau is a special case of the general tableau:

Tableau-G

			$-y_{\hat{B}} =$ x_N	$\omega =$ rhs
$y_{\hat{N}}$	x_B	=	H	h
rhs	z	=	p^T	α

where the sets b and N form a partition of $\{1,2,\ldots,n+m\}$ (containing m and n indices respectively), while the sets $\hat{B}$ and $\hat{N}$ for a partition of $\{1,2,\ldots,n+m\}$ (containing n and m indices respectively). The initial tableau above has $B=\{n+1,\ldots,n+m\}$ and $\hat{B}=\{m+1,\ldots,m+n\}$.
To proceed with the description of the dual simplex method, we make the following assumption:

$$p \leq 0.$$

Then, for the Tableau-G, we have the following key **rules for the dual simplex method**:
(Step 1) (Pivot row selection): The pivot row is any r with $h_r < 0$. If none exist, the current tableau is dual optimal.
(Step 2) (Pivot column selection): The pivot column is any column s such that

$$\frac{p_s}{H_{rs}} = \max_j \left\{ p_j / H_{rj} \mid H_{rj} > 0 \right\}.$$

If $H_{rj} \leq 0$ for all j, the dual objective is unbounded below.
We now illustrate the method with the following example:

■ **Example 3.4** Use the dual Simplex method to solve the following problem:

$$\begin{array}{ll} \max & -x_1 - x_2 \\ \text{s.t.} & 3x_1 + x_2 \geq 2, \\ & 3x_1 + 4x_2 \geq 5, \\ & 4x_1 + 2x_2 \geq 8, \\ & x_1 \geq 0,\ x_2 \geq 0. \end{array}$$

The initial tableau is obtained as

Tableau 1

			$-y_4 =$ x_1	$-y_5 =$ x_2	$w =$ rhs
y_1	x_3	=	3	1	-2
y_2	x_4	=	3	4	-5
y_3	x_5	=	$\boxed{4}$	2	-8
rhs	z	=	-1	-1	0

x_5 (y_3) – is the leaving (entering) variable,
x_1 (y_4) – is the entering (leaving variable.

Hence, the second tableau follows as

Tableau 2

			$-y_3 =$ x_5	$-y_5 =$ x_2	$w =$ rhs
y_1	x_3	=	$3/4$	$-1/2$	4
y_2	x_4	=	$3/4$	$5/2$	1
y_4	x_1	=	$1/4$	$-1/2$	2
rhs	z	=	$-1/4$	$-1/2$	-2

The primal optimal solution is

$$x_1 = 2, x_2 = 0 \text{ with } z = -2$$

and the dual optimal solution is

$$y_1 = y_2 = 0,\ y_3 = 1/4 \text{ with } w = -2.$$

■

R Note that if the primal simplex method were employed, we would have had to apply a phase I procedure first, and the computational effort would have been greater.

The dual simplex method is often used in the situation illustrated in the following example.

■ **Example 3.5** (i) First solve the following problem using the simplex method:

$$\begin{array}{ll} \max & 2x_1 + 3x_2 + 2x_3 \\ \text{s.t.} & x_1 + 2x_2 + 4x_3 \leq 8, \\ & 2x_1 + x_2 + x_3 \leq 6, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{array}$$

(ii) Next, the following new constraint is added to the above problem:

$$x_1 + x_3 \geq 3.$$

Then we would like to find an optimal solution to the new problem.

To solve the question, we first start with (i), by introducing two slack variables x_4 and x_5. Then, we have as initial tableau

Tableau 1

		x_1	x_2	x_3	rhs
x_4	=	-1	$\boxed{-2}$	-4	8
x_5	=	-2	-1	-1	6
z	=	2	3	2	0

Tableau 2

$$\begin{array}{ccc|ccc|c|}
 & & & x_1 & x_4 & x_3 & \text{rhs} \\
\cline{4-7}
x_2 & = & & -1/2 & -1/2 & -2 & 4 \\
x_5 & = & & \boxed{-3/2} & 1/2 & 1 & 2 \\
\cline{4-7}
z & = & & 1/2 & -3/2 & -4 & 12 \\
\cline{4-7}
\end{array}$$

Tableau 3

$$\begin{array}{ccc|ccc|c|}
 & & & x_5 & x_4 & x_3 & \text{rhs} \\
\cline{4-7}
x_2 & = & & 1/3 & -2/3 & -7/3 & 10/3 \\
x_1 & = & & -2/3 & 1/3 & 2/3 & 4/3 \\
\cline{4-7}
z & = & & -1/3 & -4/3 & -11/3 & 38/3 \\
\cline{4-7}
\end{array}$$

The optimal solution is

$$x_1 = \frac{4}{3},\ x_2 = \frac{10}{3},\ x_3 = 0,\ z = \frac{38}{3}.$$

Now we proceed it (ii), by adding the new constraint. Introduce a new slack variable x_6 and express it in terms of the nonbasic variables:

$$\begin{aligned}
x_6 &= x_1 + x_3 - 3 \\
&= -\frac{2}{3}x_5 + \frac{1}{3}x_4 + \frac{2}{3}x_3 + \frac{4}{3} + x_3 - 3 \\
&= -\frac{2}{3}x_5 + \frac{1}{3}x_4 + \frac{5}{3}x_3 - \frac{5}{3}.
\end{aligned}$$

Append this equation to the final optimal tableau in (ii) we have

Tableau 4

$$\begin{array}{ccc|ccc|c|}
 & & & x_5 & x_4 & x_3 & \text{rhs} \\
\cline{4-7}
x_2 & = & & 1/3 & -2/3 & -7/3 & 10/3 \\
x_1 & = & & -2/3 & 1/3 & 2/3 & 4/3 \\
x_6 & = & & -2/3 & 1/3 & \boxed{5/3} & -5/3 \\
\cline{4-7}
z & = & & -1/3 & -4/3 & -11/3 & 38/3 \\
\cline{4-7}
\end{array}$$

Tableau 5

		x_5	x_4	x_6	rhs
x_2	=	$-3/5$	$-1/5$	$-7/5$	1
x_1	=	$-2/5$	$1/5$	$2/5$	2
x_3	=	$2/5$	$-1/5$	$3/5$	1
z	=	$-9/5$	-3	$-11/5$	9

The optimal solution is

$$x_1 = 2, \;\; x_2 = 1, \;\; x_3 = 1, \;\; z = 9.$$

■

3.2.3 The weak and strong theorems in duality theory

Let us consider the primal and the dual problems:

Primal problem	Dual problem
$\max \quad z = \mathbf{c}^T\mathbf{x}$	$\min \quad \omega = \mathbf{b}^T\mathbf{y}$
s.t. $\quad A\mathbf{x} \le \mathbf{b}$	s.t. $\quad A^T\mathbf{y} \ge \mathbf{c}$
$\mathbf{x} \ge 0$	$\mathbf{y} \ge 0$

Let $\mathbf{x}$ and $\mathbf{y}$ be feasible solutions of the primal and dual problems respectively. Then

$$z = \mathbf{c}^T\mathbf{x} \le \mathbf{y}^T A\mathbf{x} \le \mathbf{y}^T\mathbf{b} = \omega.$$

The gives the following so-called *weak duality theorem*:

Theorem 3.2.1 (**Weak duality theorem**) Suppose $\mathbf{x}$ and $\mathbf{y}$ are the feasible solutions of the primal and dual problems respectively. Then, we must have

$$z = \mathbf{c}^T\mathbf{x} \le \mathbf{b}^T\mathbf{y} = \omega.$$

That is, we always have

$$z^* \le \omega^*,$$

where z^* is the optimal objective function value of the primal problem and ω^* is the optimal objective function value of the dual problem.

We now discuss one of the most fundamental theorems of linear programming.

Theorem 3.2.2 (**Strong duality theorem**)
(a) Both primal and dual problems are feasible and consequently both have optimal solutions with equal extrema.
(b) Exactly one of the problems is infeasible and consequently the problem has an unbounded objective function.
(c) Both primal and dual problems are infeasible.

Proof. (a) This has been proved in the previous section.
(b) Consider the case in which exactly one of the problems is infeasible. Suppose that the other problem has a bounded objective. The the simplex method with the smallest-subscript rule (Bland's rule)

will terminate at a primal-dual optimal tableau (Tableau-O in the previous section), contradicting infeasibility. Hence, the feasible problem cannot have a bounded objective.
(c) We can illustrate this case by setting $A = 0$, $b = -1$ and $c = 1$. ■

3.2.4 Applications of duality

Farkas Lemma

The Farkas Lemma is often used in linear programming to prove the strong duality theorem. In this section, we show how one can prove the Farkas Lemma from the strong duality. The Farkas Lemma is well known in dealing with linear equations and is in fact an example of a *theorem of the alternative*.

Let us consider n points $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$, in $\Re^m$. We collect them in the matrix

$$A = [\mathbf{a}_1,\ \mathbf{a}_2,\ \dots, \mathbf{a}_n].$$

We consider the set $\mathscr{C}$ spanned by those points in the following way:

$$\begin{aligned}\mathscr{C} &= \{x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \cdots + x_n\mathbf{a}_n \mid x_1 \ge 0,\ x_2 \ge 0, \dots, x_n \ge 0\} \\ &= \{A\mathbf{x} \mid \mathbf{x} \ge 0\}.\end{aligned}$$

Suppose there exists another point $\mathbf{b} \in \Re^m$. Then exactly one of the following two claims is true:
(I') $\mathbf{b} \in \mathscr{C}$.
(II') There exists a vector $0 \ne \mathbf{y} \in \Re^m$, which defines the hyperplane

$$H := \{\mathbf{v} \in \Re^m \mid \langle \mathbf{y},\ \mathbf{v}\rangle = 0\},$$

such that $\mathbf{b}$ is on the negative side of H and $\mathscr{C}$ is on the positive side of H.
The claim (II') is an example of a *separation theorem*. The above two claims are actually the content of the Farkas Lemma:

Theorem 3.2.3 (**Farkas Lemma**) Let $A \in \Re^{m\times n}$ and $\mathbf{b} \in \Re^m$. Then exactly one of the following systems has a solution.
(I) $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \ge 0$.
(II) $A^T\mathbf{y} \ge 0$ and $\langle \mathbf{b}, \mathbf{y}\rangle < 0$.

Proof. Consider the linear program:

$$\begin{aligned}\max\quad & \langle \mathbf{0},\ \mathbf{x}\rangle \\ \text{s.t.}\quad & A\mathbf{x} = \mathbf{b},\ \mathbf{x} \ge 0\end{aligned} \tag{3.12}$$

and its dual problem

$$\begin{aligned}\min\quad & \langle \mathbf{b},\ \mathbf{y}\rangle \\ \text{s.t.}\quad & A^T\mathbf{y} \ge 0.\end{aligned} \tag{3.13}$$

If (I) has a solution, then any feasible point of (I) is an optimal solution of the primal problem (3.12) with the optimal objective being 0. According to the strong duality theorem, we must have that the optimal objective of the dual problem is also 0, which means

$$\langle \mathbf{b},\ \mathbf{y}\rangle \ge 0,\ A^T\mathbf{y} \ge 0.$$

Hence $\langle \mathbf{b}, \mathbf{y}\rangle < 0$ is not possible for any $\mathbf{y}$ satisfying $A^T\mathbf{y} \ge 0$. That is, (II) cannot not hold.

Now suppose (I) does not hold. Then the primal problem is infeasible. The strong duality theorem says that the dual problem is unbounded from below. This means that there exists $\mathbf{y} \in \Re^m$ such that

$$\langle \mathbf{b}, \mathbf{y} \rangle < 0 \quad \text{and} \quad A^T \mathbf{y} \geq 0.$$

That is, (II) has a solution. We completed the proof of the Farkas Lemma. ■

We note that (II) is just an equivalent statement in (II'). (II) means that $\mathbf{b} \notin \mathscr{C}$. Therefore, it follows from (II') that there exists $0 \neq \mathbf{y} \in \Re^m$ such that

$$\langle \mathbf{b}, \mathbf{y} \rangle < 0 \quad \text{and} \quad \langle \mathbf{y}, \mathbf{z} \rangle \geq 0, \ \forall\, \mathbf{z} \in \mathscr{C}.$$

We further have

$$0 \leq \langle \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{y}, A\mathbf{x} \rangle = \langle A^T \mathbf{y}, \mathbf{x} \rangle, \quad \forall\, \mathbf{x} \geq 0,$$

which is equivalent to say that $A^T \mathbf{y} \geq 0$. Hence, (II) and (II') are equivalent.

Theorem of complementary slackness

Consider the following primal and dual problems:

Primal problem	Dual problem
$\begin{array}{ll} \max & z = \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & A\mathbf{x} + \mathbf{s} = \mathbf{b} \\ & \mathbf{x} \geq 0,\ \mathbf{s} \geq 0 \end{array}$	$\begin{array}{ll} \min & \omega = \mathbf{b}^T \mathbf{y} \\ \text{s.t.} & A^T \mathbf{y} - \mathbf{t} = \mathbf{c} \\ & \mathbf{y} \geq 0,\ \mathbf{t} \geq 0 \end{array}$

Theorem 3.2.4 (**Theorem of complementary slackness**) Let $(\mathbf{x}^*, \mathbf{s}^*)$ and $(\mathbf{y}^*, \mathbf{t}^*)$ be respectively the optimal solutions of the primal and dual problems. Then,

$$\begin{array}{ll} y_i^* s_i^* = 0 & \text{for } i = 1, \ldots, m, \\ x_j^* t_j^* = 0 & \text{for } j = 1, \ldots, n. \end{array}$$

Proof. Using the feasibility of the optimal solution, we have

$$\begin{aligned} z^* = \mathbf{c}^T \mathbf{x}^* = ((\mathbf{y}^*)^T A - (\mathbf{t}^*)^T)\mathbf{x}^* &= (\mathbf{y}^*)^T (\mathbf{b} - \mathbf{s}^*) - (\mathbf{t}^*)^T \mathbf{x}^* \\ &= \omega^* - (\mathbf{y}^*)^T \mathbf{s}^* - (\mathbf{t}^*)^T \mathbf{x}^*. \end{aligned}$$

From the strong duality theorem theorem, $z^* = \omega^*$. Thus,

$$(\mathbf{y}^*)^T \mathbf{s}^* + (\mathbf{t}^*)^T \mathbf{x}^* = 0,$$

which is

$$y_1^* s_1^* + \cdots + y_m^* s_m^* + t_1^* x_1^* + \cdots + t_n^* x_n^* = 0.$$

By non-negativity, each individual term in this equation is zero. ■

■ **Example 3.6** For the problem

$$\begin{array}{ll} \max & z = 24x_1 + 20x_2 + 9x_3 \\ \text{s.t.} & 8x_1 + 2x_2 - 3x_3 \leq 7, \\ & 4x_1 + 5x_2 + 3x_3 \leq 17, \\ & x_1 + 3x_2 + 4x_3 \leq 16, \\ & x_1, x_2, x_3 \geq 0, \end{array}$$

determine whether the following solution is optimal:

$$x_1 = 2,\ x_2 = 0,\ x_3 = 3.$$

Solution: The dual of the problem is

$$\begin{array}{ll} \min & \omega = 7y_1 + 17y_2 + 16y_3 \\ \text{s.t.} & 8y_1 + 4y_2 + y_3 \geq 24, \\ & 2y_1 + 5y_2 + 3y_3 \geq 20, \\ & -3y_1 + 3y_2 + 4y_3 \geq 9, \\ & y_1, y_2, y_3 \geq 0. \end{array}$$

Let s_1, s_2, s_3 be slack variables for the primal problem and t_1, t_2, t_3 be slack variables for the dual one. For the proposed solution, we have

$$s_1 = 0,\ s_2 = 0,\ s_3 = 2,\ z = 75.$$

Using complementary slackness,

$$\begin{array}{ll} x_j t_j = 0 & \text{yields } t_1 = 0,\ t_3 = 0, \\ y_i s_i = 0 & \text{yields } y_3 = 0. \end{array}$$

Thus, the dual constraints become

$$\begin{array}{ll} 8y_1 + 4y_2 & = 24, \\ 2y_1 + 5y_2 - t_2 & = 20, \\ -3y_1 + 3y_2 & = 9. \end{array}$$

Solving the first and third equations gives

$$y_1 = 1,\ \ y_2 = 4.$$

Substituting in the second equation gives

$$t_2 = 2.$$

Thus, we have a feasible dual solution and

$$\omega = 7y_1 + 17y_2 = 7 + 68 = 75 = z.$$

The weak duality theorem shows that the proposed primal solution is optimal. ∎

Existence of strictly complementary solutions

This part is the continuation of the preceding subsection on the complementarity theorem. If we introduce the Hadamard product between vectors (i.e., the componentwise product):

$$\mathbf{u} \circ \mathbf{v} = \begin{bmatrix} u_1 v_1 \\ u_2 v_2 \\ \vdots \\ u_n v_n \end{bmatrix} \quad \text{for all} \quad \mathbf{u}, \mathbf{v} \in \Re^n.$$

Then Theorem 3.2.4 says that

$$\mathbf{y}^* \circ \mathbf{s}^* = 0 \quad \text{and} \quad \mathbf{x}^* \circ \mathbf{t}^* = 0$$

with

$$\mathbf{x}^* \geq 0,\ \mathbf{y}^* \geq 0,\ \mathbf{s}^* \geq 0,\ \mathbf{t}^* \geq 0.$$

This result can be further strengthened to

$$\mathbf{y}^* + \mathbf{s}^* > 0 \quad \text{and} \quad \mathbf{x}^* + \mathbf{t}^* > 0.$$

But it only holds for certain pairs.

Theorem 3.2.5 (**Strict complementarity theorem**) Suppose the primal problem has an optimal solution (this assumption means that the dual problem also has an optimal solution by the strong duality theorem). Then there exist a pair of primal solution $(\mathbf{x}^*, \mathbf{s}^*)$ and a pair of the optimal dual solution $(\mathbf{y}^*, \mathbf{t}^*)$ such that

$$\mathbf{x}^* \circ \mathbf{t}^* = 0, \quad \mathbf{y}^* \circ \mathbf{s}^* = 0$$

and

$$\mathbf{y}^* + \mathbf{s}^* > 0 \ \text{ and } \ \mathbf{x}^* + \mathbf{t}^* > 0.$$

Proof. **Step 1** (Construction of a new LP problem). Let us consider a new (primal) problem:

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{y},\mathbf{s},\mathbf{t},\varepsilon} \quad & \varepsilon \\
\text{s.t.} \quad & A\mathbf{x} + \mathbf{s} = \mathbf{b}, \ \ \mathbf{x} \geq 0, \ \mathbf{s} \geq 0, && (3.14) \\
& -A^T\mathbf{y} + \mathbf{t} = -\mathbf{c}, \ \ \mathbf{y} \geq 0, \ \mathbf{t} \geq 0, && (3.15) \\
& \mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{y} = 0, && (3.16) \\
& -\mathbf{x} - \mathbf{t} + \varepsilon\mathbf{e} \leq 0, && (3.17) \\
& -\mathbf{y} - \mathbf{s} + \varepsilon\mathbf{e} \leq 0, && (3.18) \\
& \varepsilon \leq 1, && (3.19)
\end{aligned}$$

where $\mathbf{e}$ is the vector of all ones. We note that Constraint (3.14) is just the primal constraints and (3.15) is just the dual constraints. And (3.16) just requires that the primal objective and the dual objective should be equal.
If follows from Theorem 3.2.4 that

$$\mathbf{x}^*, \ \mathbf{y}^*, \ \mathbf{s}^*, \ \mathbf{t}^*, \ \varepsilon = 0$$

satisfy the constraints in the new problem. Since $\varepsilon \leq 1$ by the constraint (3.19), the new problem is bounded above and hence it must have an optimal solution. Since $\varepsilon = 0$ is feasible, the optimal objective $\varepsilon^* \geq 0$. From now on, we assume that the optimal objective value $\varepsilon^* = 0$.

Step 2: (The dual problem). Let us define the new matrix

$$\overline{A} = \begin{bmatrix} A & 0 & I & 0 & 0 \\ 0 & -A^T & 0 & I & 0 \\ \mathbf{c}^T & -\mathbf{b}^T & 0 & 0 & 0 \\ -I & 0 & 0 & -I & \mathbf{e} \\ 0 & -I & -I & 0 & \mathbf{e} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{c} \\ 0 \\ \mathbf{0} \\ 1 \end{bmatrix}, \quad \overline{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \\ \mathbf{t} \\ \varepsilon \end{bmatrix}.$$

Then the primal problem takes the following form

$$\begin{aligned}
\max \quad & \varepsilon = [\mathbf{0}, 1]\overline{\mathbf{x}} \\
\text{s.t.} \quad & \overline{A}\overline{\mathbf{x}} \begin{pmatrix} = \\ = \\ = \\ \leq \\ \leq \\ \leq \end{pmatrix} \overline{b} \ \text{ and } \ \mathbf{x} \geq 0, \ \mathbf{y} \geq 0, \ \mathbf{s} \geq 0, \ \mathbf{t} \geq 0, \ \varepsilon \text{ is free.}
\end{aligned}$$

Introducing the slack variables corresponding to the constraints in (3.14) to (3.19): $\mathbf{u}$, $\mathbf{v}$, τ, $\mathbf{w}$, $\mathbf{z}$, and ρ. The dual problem becomes

$$
\begin{aligned}
\min \quad & \langle \mathbf{b}, \mathbf{u} \rangle - \langle \mathbf{c}, \mathbf{v} \rangle + \rho \\
\text{s.t.} \quad & A^T \mathbf{u} + \tau \mathbf{c} - \mathbf{w} \geq 0, && (3.20) \\
& -A\mathbf{v} - \tau \mathbf{b} - \mathbf{z} \geq 0, && (3.21) \\
& \mathbf{u} - \mathbf{z} \geq 0, && (3.22) \\
& \mathbf{v} - \mathbf{w} \geq 0, && (3.23) \\
& \langle \mathbf{e}, \mathbf{w} + \mathbf{z} \rangle + \rho = 1, && (3.24) \\
& \mathbf{u},\ \mathbf{v},\ \tau \text{ are free},\ \mathbf{w} \geq 0,\ \mathbf{z} \geq 0,\ \rho \geq 0.
\end{aligned}
$$

Step 3 (Deriving a contradiction). By the strong duality theorem, the dual problem must have an optimal solution $\mathbf{u}^*$, $\mathbf{v}^*$, τ^*, $\mathbf{w}^*$, $\mathbf{z}^*$ and ρ^*. To simplify the notation, we drop the $*$ from the solution notation.

Pre-multiplying $\mathbf{v}^T$ with (3.20) and pre-multiplying $\mathbf{u}^T$ with (3.21) and adding them together, we get the inequality

$$\tau(\langle \mathbf{c}, \mathbf{v} \rangle - \langle \mathbf{b}, \mathbf{u} \rangle) - \langle \mathbf{w}, \mathbf{v} \rangle - \langle \mathbf{z}, \mathbf{u} \rangle \geq 0. \tag{3.25}$$

We recall that the optimal primal objective $\varepsilon^* = 0$, the last constraint (3.19) is in active. Hence,

$$\rho = 0.$$

Moreover, the dual objective is also zero due to the strong duality theorem:

$$0 = \langle \mathbf{b}, \mathbf{u} \rangle - \langle \mathbf{c}, \mathbf{v} \rangle + \rho.$$

It follows that

$$\langle \mathbf{b}, \mathbf{u} \rangle = \langle \mathbf{c}, \mathbf{v} \rangle.$$

It then follows from (3.25) that

$$\langle \mathbf{w}, \mathbf{v} \rangle + \langle \mathbf{z}, \mathbf{u} \rangle \leq 0. \tag{3.26}$$

We also note from (3.22) that (since $\mathbf{z} \geq 0$)

$$\mathbf{u} \geq \mathbf{z} \implies \langle \mathbf{u}, \mathbf{z} \rangle \geq \|\mathbf{z}\|^2 \geq 0. \tag{3.27}$$

Similarly, it follows from (3.23) that (since $\mathbf{w} \geq 0$)

$$\mathbf{v} \geq \mathbf{w} \implies \langle \mathbf{w}, \mathbf{v} \rangle \geq \|\mathbf{w}\|^2 \geq 0. \tag{3.28}$$

The inequality in (3.26) implies

$$\langle \mathbf{u}, \mathbf{z} \rangle = 0 \quad \text{and} \quad \langle \mathbf{w}, \mathbf{v} \rangle = 0,$$

which in turn implies that (see 3.27 and (3.28))

$$\mathbf{z} = 0, \ \ \mathbf{w} = 0.$$

The last constraint (3.24) yields

$$1 = \langle \mathbf{e}, \mathbf{w} + \mathbf{z} \rangle + \rho = 0.$$

This is a contradiction, which implies that our assumption $\varepsilon^* = 0$ cannot hold. Therefore $\varepsilon^* > 0$.

The result $\varepsilon^* > 0$ exactly implies the strict complementarity theorem holds. ■

Numerical examples

We note that the solutions obtained in Example 3.6 satisfy the strict complementarity condition. We now give one more example.

■ **Example 3.7** Find a strictly complementary optimal solution of the LP below:

$$\begin{array}{ll} \max & z = 7x_1 + 2x_2 + x_3 \\ \text{s.t.} & 2x_1 + x_2 - x_3 \le 2, \\ & 4x_1 + x_2 + x_3 \le 6, \\ & x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0. \end{array}$$

Solution: We first consider the dual problem:

$$\begin{array}{ll} \min & \omega = 2y_1 + 6y_2 \\ \text{s.t.} & 2y_1 + 4y_2 \ge 7, \\ & y_1 + y_2 \ge 2, \\ & -y_1 + y_2 \ge 1, \\ & y_1 \ge 0,\ y_2 \ge 0. \end{array}$$

The dual problem has two variables and hence can be solved by the graphical method, which yields that the dual optimal solution is

$$y_1^* = \frac{1}{2}, \qquad y_2^* = \frac{3}{2}, \;\; \omega^* = 10.$$

By the complementary slackness theorem, we know that the two linear constraints in the primal problems are equations at its optimal solution (i.e., the corresponding slack values are zero):

$$\left\{ \begin{array}{rcrcrcr} 2x_1 & + & x_2 & - & x_3 & = & 2, \\ 4x_1 & + & x_2 & + & x_3 & = & 6. \end{array} \right.$$

(We note that the slack values of the dual problem at its optimal solution are all zeros, i.e., $\mathbf{t}^* = 0$. Hence, we cannot claim that any of the x values must be zero).
The optimal objective of the dual problem is 10. Therefore, by the strong duality theorem, we must have that the optimal value of the primal problem is also 10:

$$7x_1 + 2x_2 + x_3 = 10.$$

Therefore, we have a 3×3 linear equations:

$$\left\{ \begin{array}{rcrcrcr} 2x_1 & + & x_2 & - & x_3 & = & 2, \\ 4x_1 & + & x_2 & + & x_3 & = & 6, \\ 7x_1 & + & 2x_2 & + & x_3 & = & 10. \end{array} \right.$$

Unfortunately, the system is singular and has infinitely many solutions.
Case 1: Assume $x_1 = 0$. This gives

$$x_2 = 4, \;\; x_3 = 2.$$

This is one of the optimal solutions and is denoted by

$$\mathbf{x}_1^* = (0,\ 4,\ 2)^T.$$

Case 2: Assume $x_2 = 0$. This gives

$$x_1 = \frac{4}{3}, \;\; x_3 = \frac{2}{3}.$$

This is another optimal solution and is denoted by

$$\mathbf{x}_2^* = (4/3,\ 0,\ 2/3)^T.$$

Case 3: Assume $x_3 = 0$. This gives

$$x_1 = 2, \ \ x_2 = -2.$$

Since $x_2 < 0$, this solution is infeasible.
We can construct many solutions from the two solutions found.

$$\mathbf{x}^* = \rho \mathbf{x}_1^* + (1-\rho)\mathbf{x}_2^* \quad \text{for all} \quad 0 \leq \rho \leq 1.$$

One of the strictly complementary optimal solution is

$$\mathbf{x}^* = \frac{1}{2}\mathbf{x}_1^* + \frac{1}{2}\mathbf{x}^{r}{*_2} = \frac{1}{2}\begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 4/3 \\ 0 \\ 2/3 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 2 \\ 4/3 \end{bmatrix}.$$

We must have

$$\mathbf{x}^* + \mathbf{t}^* > 0.$$

■

3.3 Sensitivity analysis

3.3.1 Canonical form and feasible tableaux

We recall the standard form of the linear programming problem

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.t.} & Ax \leq b, \ x \geq 0, \end{array} \tag{3.29}$$

where $A \in \Re^{m \times n}$, and now define another form of linear programming problem, that we have seen in the context of the feasible tableaus: is in this form.

Definition 3.3.1 The following problem is known as the *canonical form* of the linear programming problem:

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.t.} & \mathscr{A} x = b, \ x \geq 0. \end{array} \tag{3.30}$$

Here, $\mathscr{A} \in \Re^{m \times \ell}$.

We can convert the standard form into the canonical form by defining

$$\mathscr{A} = [A, \ \ I]. \tag{3.31}$$

However, $\mathscr{A}$ is not always given in the form of (3.31).
We have mentioned that every feasible tableau defines a linear programming problem in canonical form. Conversely, we can easily construct tableaux from a canonical form of the LP by specifying the set B of basic variables. Now look at the problem (3.30). Suppose we have

$$B \subset \{1,2,\ldots,\ell\} \quad \text{and} \quad N = \{1,2,\ldots,\ell\} \setminus B.$$

The equation $\mathscr{A}x = b$ can be rearranged as

$$\mathscr{A}_B x_B + \mathscr{A}_N x_N = b \tag{3.32}$$

and the objective function can be rearranged as

$$z = c_B^T x_B + c_N^T x_N. \tag{3.33}$$

Assuming $\mathscr{A}_B^{-1}$ exists (i.e., $\mathscr{A}_B$ is invertible), it follows from (3.32) and (3.33) that

$$x_B = -\mathscr{A}_B^{-1}\mathscr{A}_N x_N + \mathscr{A}_B^{-1} b$$

and

$$z = \left(c_N^T - c_B^T \mathscr{A}_B^{-1}\mathscr{A}_N\right) x_N + c_B^T \mathscr{A}_B^{-1} b.$$

A tableau is produced:

		x_N	rhs
x_B	=	$-\mathscr{A}_B^{-1}\mathscr{A}_N$	$\mathscr{A}_B^{-1} b$
z	=	$c_N^T - c_B^T \mathscr{A}_B^{-1}\mathscr{A}_N$	$c_B^T \mathscr{A}_B^{-1} b$

Table 3.3: Compact tableau

This tableau is feasible if

$$\mathscr{A}_B^{-1} b \geq 0,$$

and it is optimal if

$$\mathscr{A}_B^{-1} b \geq 0 \quad \text{and} \quad c_N^T - c_B^T \mathscr{A}_B^{-1}\mathscr{A}_N \leq 0. \tag{3.34}$$

We illustrate those results by an example.

■ **Example 3.8** Consider the problem

$$\begin{array}{ll} \max & z = -x_1 - 1.5x_2 - 3x_3 \\ \text{subject to} & x_1 + x_2 + 2x_3 \geq 6, \\ & x_1 + 2x_2 + x_3 \geq 10, \\ & x_1,\ x_2,\ x_3 \geq 0. \end{array}$$

By introducing slack variables x_4 and x_5, we arrive at the canonical form (3.30) with

$$\mathscr{A} = \begin{bmatrix} 1 & 1 & 2 & -1 & 0 \\ 1 & 2 & 1 & 0 & -1 \end{bmatrix}, \ b = \begin{bmatrix} 6 \\ 10 \end{bmatrix}, \ c = \begin{bmatrix} -1 \\ -1.5 \\ -3 \\ 0 \\ 0 \end{bmatrix}.$$

Consider the base $B = \{1, 2\}$, for which we have

$$\mathscr{A}_B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \ \mathscr{A}_N = \begin{bmatrix} 2 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathscr{A}_B^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.$$

The terms from the compact tableau defining the conditions in (3.34) are respectively

$$\mathscr{A}_B^{-1}b = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 10 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

and

$$\begin{aligned} c_N^T - c_B^T \mathscr{A}_B^{-1} \mathscr{A}_N &= [-3\ \ 0\ \ 0] - [-1\ \ -1.5] \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \\ &= [-3\ \ 0\ \ 0] - [-1.5\ \ 0.5\ \ 0.5] = [-1.5\ \ -0.5\ \ -0.5]. \end{aligned}$$

Thus, the basis $\{1,2\}$ is optimal. ■

3.3.2 Perturbations to b and c

■ **Example 3.9** (Perturbation to `rhs` – Example 3.8 continued) Suppose now we would like to perturb the right-hand-side (rhs) b to $\tilde{b}$ in the following way

$$\tilde{b} = b + \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix}.$$

Find the range of ε such that the basis $B = \{1,2\}$ still remains optimal.
Solution: By inspecting Tableau 3.3.1, we see that the bottom row does not change when b is perturbed to $\tilde{b}$. That is

$$c_N^T - c_B^T \mathscr{A}_B^{-1} \mathscr{A}_N \leq 0$$

by previous calculation. We only need $\mathscr{A}_B^{-1} b \geq 0$ (see (3.34)) in order to make sure B is optimal.

$$\begin{aligned} \mathscr{A}_B^{-1} \tilde{b} &= \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} b + \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \varepsilon \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 2\varepsilon \\ -\varepsilon \end{bmatrix} \geq 0, \end{aligned}$$

which leads to

$$-1 \leq \varepsilon \leq 4.$$

This is the range that will make B remain optimal. ■

■ **Example 3.10** (Perturbation to c – Example 3.8 continued) We still let

$$B = \{1,2\} \quad \text{and} \quad N = \{3,4,5\}.$$

We only consider the case where perturbation occurs to c_j, $j \in N$. Suppose c_3 is perturbed from the current value to $\tilde{c}_3 = c_3 + \delta$. What is the range of ε such that B remains optimal.
Solution: By inspecting Tableau 3.3.1, we see that the last column does not change when c is perturbed to $\tilde{c}$. That is

$$\mathscr{A}_B^{-1} b \geq 0$$

by previous calculation. We only need (see (3.34))

$$\tilde{c}_N^T - \tilde{c}_B^T \mathscr{A}_B^{-1} \mathscr{A}_N \leq 0$$

in order to make sure B is optimal.

We note that only c_3 was perturbed, we see that

$$\tilde{c}_B = c_B.$$

We have

$$\begin{aligned}
\tilde{c}_N^T - \tilde{c}_B^T \mathscr{A}_B^{-1} \mathscr{A}_N &= [\delta \;\; 0 \;\; 0] + c_N^T - c_B^T \mathscr{A}_B^{-1} \mathscr{A}_N \\
&= [\delta \;\; 0 \;\; 0] + [-1.5 \;\; -0.5 \;\; -0.5] \\
&= [-1.5+\delta \;\; -0.5 \;\; -0.5] \leq 0.
\end{aligned}$$

This leads to

$$\delta \leq 1.5.$$

This is the range of ε that keeps the basis $B = \{1,2\}$ optimal. ■

3.3.3 Parametric optimization of the objective function

Let us look at an example first:

$$\begin{aligned}
\max \quad & z(t) = x_1 + x_2 + t(x_1 - x_2) \\
\text{s.t.} \quad & x_1 + x_2 \leq 6, \\
& x_1 \leq 4, \\
& x_2 \leq 3, \\
& x_1 \geq 0, \; x_2 \geq 0,
\end{aligned}$$

where $t \in (-\infty, \infty)$. Since the problem has just 2 variables, we can solve it using the graphical method.

From the graph of the problem, we see that the feasible region has 5 vertices (corner points) and they are:

$$P_1: \; (0,0); \quad P_2: \; (0,3); \quad P_3: \; (3,3); \quad P_4: \; (4,2); \quad P_5: \; (4,0).$$

It follows from the basic theory of LP, we know for any given t, one of the vertices will solve the corresponding linear programming. In fact, we can verify the following cases.

Case 1. For $t \in (-\infty, -1]$, $P_2: \; (0,3)$ is the optimal solution.
Case 2. For $t \in [-1, 0]$, $P_3: \; (3,3)$ is the optimal solution.
Case 3. For $t \in [0,1]$, $P_4: \; (4,2)$ is the optimal solution.
Case 4. For $t \in [1, \infty)$, $P_5: \; (4,0)$ is the optimal solution.

In particular, for $t = -1$, both P_2 and P_3 are optimal. Therefore, the edge connecting P_2 and P_3 are optimal for $t = -1$. For $t = 0$, both P_3 and P_4 are optimal. Therefore, the edge connecting P_3 and P_4 are optimal for $t = 0$. And for $t = 1$, both P_4 and P_5 are optimal. Therefore, the edge connecting P_4 and P_4 are optimal for $t = -1$. Consequently, those values $t = -1, \; 0, \; 1$ are very special and they cut the whole interval $(-\infty, \infty)$ into smaller ranges. On each of them, we know the corresponding optimal solutions and the optimal objective values.

The optimal objective function value is given by

$$z^*(t) = \begin{cases} 3(1-t) & \text{for } t \leq -1, \\ 6 & \text{for } -1 \leq t \leq 0, \\ 2t+6 & \text{for } 0 \leq t \leq 1, \\ 4(1+t) & \text{for } t \geq 1. \end{cases}$$

It is obvious that the optimal objective z^* when regarded as a function of t is a piecewise linear function. This claim is true for general parametric optimization of the objective function, which can be stated described as follows.

Consider the problem:

$$\begin{array}{ll} \max & z(t) = \mathbf{c}^T\mathbf{x} + t\mathbf{q}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \le \mathbf{b}, \ \ \mathbf{x} \ge 0, \end{array} \tag{3.35}$$

where t is a parameter in $(-\infty, \infty)$, and as before, $\mathbf{c}$, $\mathbf{x} \in \Re^n$, $A \in \Re^{m\times n}$, $\mathbf{b} \in \Re^m$ and $\mathbf{q} \in \Re^n$. For any given t, we let $z^*(t)$ be the optimal objective value. For the simplicity of the argument below, we assume that the problem (3.35) has an optimal solution for any given t.

Theorem 3.3.1 Consider the problem (3.35) and assume that it has an optimal solution for any given $t \in (-\infty, \infty)$. Then it holds that:

(i) For any $\lambda \in (0,1)$ and $t_1, t_2 \in \Re$, we have

$$z^*(\lambda t_1 + (1-\lambda)t_2) \le \lambda z^*(t_1) + (1-\lambda)z^*(t_2).$$

(ii) $z^*(t)$ is piecewise linear over $(-\infty, \infty)$.

Proof. (i) Let $\mathscr{F}$ denote the feasible region of the problem (3.35). Then we have the following chain of inequalities:

$$\begin{aligned} z^*(\lambda t_1 + (1-\lambda)t_2) &= \max_{\mathbf{x}\in\mathscr{F}} (\mathbf{c} + (\lambda t_1 + (1-\lambda)t_2)\mathbf{q})^T\mathbf{x} \\ &= \max_{\mathbf{x}\in\mathscr{F}} (\lambda\mathbf{c} + (1-\lambda)\mathbf{c} + (\lambda t_1 + (1-\lambda)t_2)\mathbf{q})^T\mathbf{x} \\ &= \max_{\mathbf{x}\in\mathscr{F}} (\lambda(\mathbf{c} + t_1\mathbf{q})^T\mathbf{x} + ((1-\lambda)(\mathbf{c} + t_2\mathbf{q})^T\mathbf{x} \\ &\le \max_{\mathbf{x}\in\mathscr{F}} (\lambda(\mathbf{c} + t_1\mathbf{q})^T\mathbf{x} + \max_{\mathbf{x}\in\mathscr{F}} ((1-\lambda)(\mathbf{c} + t_1\mathbf{q})^T\mathbf{x} \\ &= \lambda z^*(t_1) + (1-\lambda)z^*(t_2) \end{aligned}$$

(ii) For any given t, by the theory of LP, one of the vertices in $\mathscr{F}$ must be optimal. Since $\mathscr{F}$ has a finite number of vertices, the number of optimal vertices is finite. Furthermore, if a vertex is optimal at t_1 and t_2 with $t_1 < t_2$, then that vertex is also optimal at any point in the interval (t_1, t_2) (this follows easily from the linearity of the objective). Hence, the range $(-\infty, \infty)$ can be partitioined into a finite number of subintervals separated by "breakpoints" $t_1, t_2, \ldots, t_M$ at which the solution switches from one vertex to another.
Suppose that x_{i+1} is a vertex that solves the problem over one of these subintervals $[t_i, t_{i+1}]$. Then, we have

$$z^*(t) = \mathbf{c}^T\mathbf{x} + t\mathbf{q}^T\mathbf{x}_{i+1} \ \text{ for all } \ t \in [t_i, t_{i+1}].$$

So, $z^*(t)$ is linear on this interval. $z^*(t)$ is also linear in the next subinterval and is continuous at the breakpoint t_{i+1} because $\mathbf{x}_{i+1}$ solves the problem on both intervals. ■

3.4 Exercises

1. Find the dual of the following maximization LP:

$$\begin{array}{ll} \max & z = x_1 + 2x_2 \\ \text{s.t.} & 3x_1 + x_2 \le 6, \\ & 2x_1 + x_2 = 5, \\ & x_1, x_2 \ge 0. \end{array}$$

2. Write down the dual of the following minimization LP:

$$\begin{array}{ll} \min & z = 50y_1 + 20y_2 + 30y_3 + 80y_4 \\ \text{s.t.} & 3y_1 + y_2 \ge 6, \\ & 400y_1 + 200y_2 + 150y_3 + 500y_4 \ge 500, \\ & 2y_1 + 2y_2 + 4y_3 + 4y_4 \ge 10, \\ & 2y_1 + 4y_2 + y_3 + 5y_4 \ge 8, \\ & y_1,\ y_2,\ y_3,\ y_4 \ge 0. \end{array}$$

3. Consider the following LP:

$$\begin{array}{ll} \max & 5x_1 + 3x_2 + x_3 \\ \text{s. t.} & 2x_1 + x_2 + x_3 \le 6, \\ & x_1 + 2x_2 + x_3 \le 7, \\ & x_1,\ x_2,\ x_3 \ge 0. \end{array}$$

Graphically solve the dual of this LP. Then use the complementarity slackness theorem to solve the max problem.

4. Consider the following linear program:

$$\begin{array}{ll} \max & x_1 + 4x_2 + x_3 \\ \text{s.t.} & 2x_1 + 2x_2 + x_3 = 4, \\ & x_1 - x_3 = 1, \\ & x_1,\ x_2,\ x_3 \ge 0. \end{array}$$

(i) Verify that an optimal basis for this problem is $B = \{1, 2\}$, and calculate the corresponding quantities $\mathscr{A}_B$, $\mathscr{A}_B^{-1}$, $\mathscr{A}_N$, c_B, c_N, and x_B, together with the vector

$$c_N^T - c_B^T \mathscr{A}_B^{-1} \mathscr{A}_N.$$

(ii) Suppose that the right-hand side 1 of the second constraint is replaced by $1 + \varepsilon$. Calculate the range of ε for which the basis B remains optimal, and give the solution x for each value of ε in this range.

(iii) Suppose that the coefficient of x_3 in the objective is replaced by $-1 + \delta$. Find the range of δ for which the basis B remains optimal.

5. Us the complementarity slackness theorem to check whether $x^T = [7, 0, 2.5, 0, 3, 0, 0.5]$ is an optimal solution of the following linear programming problem:

$$\begin{array}{ll} \max & z = x_1 + 2x_2 + x_3 - 3x_4 + x_5 + x_6 - x_7 \\ \text{s.t.} & x_1 + x_2 - x_4 + 2x_6 - 2x_7 \le 6, \\ & x_2 - x_4 + x_5 - 2x_6 + 2x_7 \le 4, \\ & x_2 + x_3 + x_6 - x_7 \le 2, \\ & x_2 - x_4 - 6x_6 + x_7 \le 1, \\ & x_i \ge 0,\ i = 1, \ldots, 7. \end{array}$$

6. Consider the following problem:

$$\begin{aligned} \max \quad & -x_1 - 2x_2 \\ \text{s.t.} \quad & -x_1 + x_2 \geq 1, \\ & x_1 + x_2 \geq 3, \\ & x_1,\ x_2 \geq 0. \end{aligned}$$

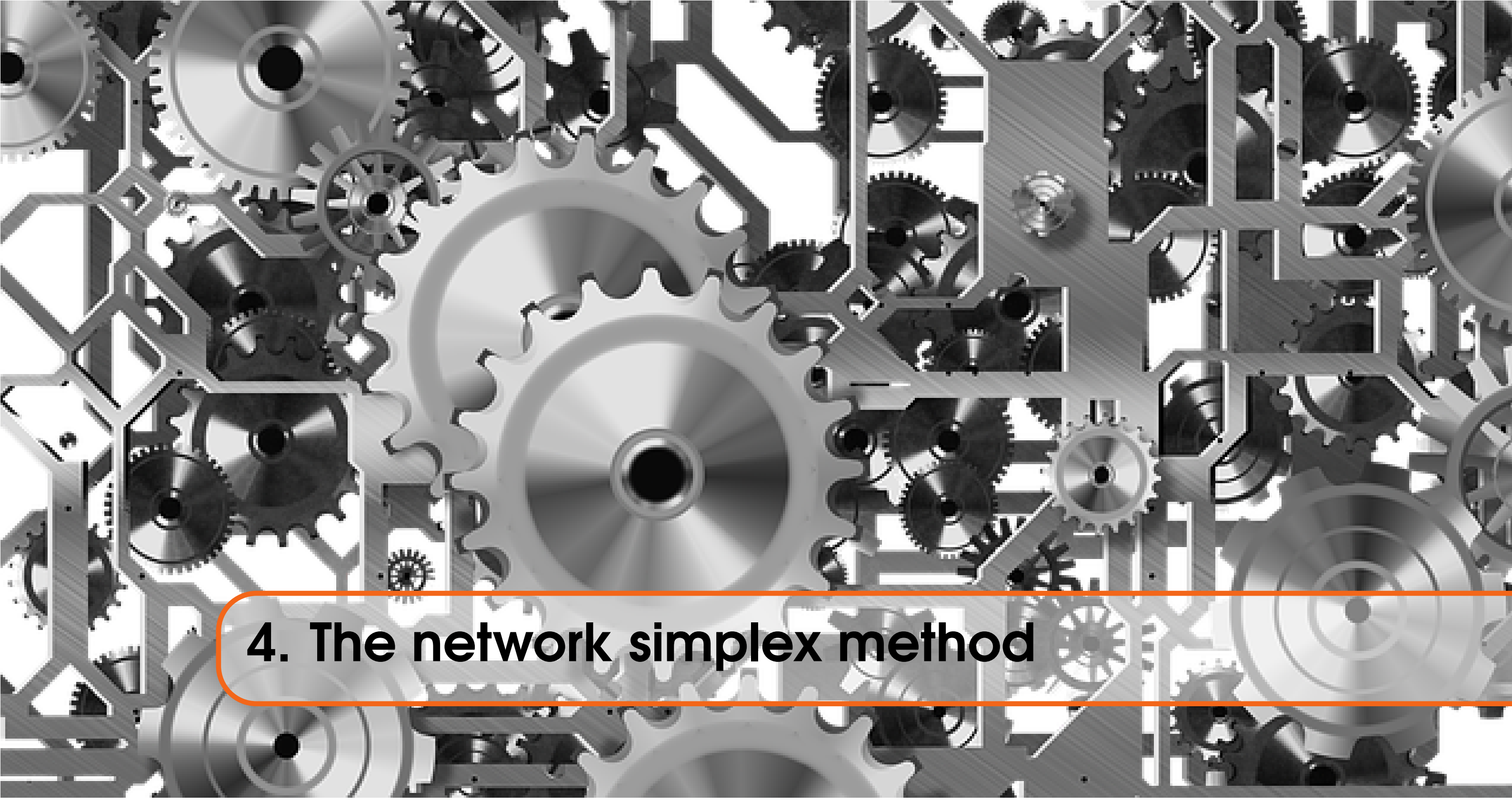
   (i) Solve this problem graphically.
   (ii) Use the dual simplex method to solve this problem.
   (iii) Trace graphically the path taken by the dual simplex method.

7. Use the dual simplex method to solve the following problem:

$$\begin{aligned} \max \quad & -5x_1 - 2x_2 - 4x_3 \\ \text{s.t.} \quad & 3x_1 + x_2 + 2x_3 \geq 4, \\ & 6x_1 + 3x_2 + 5x_3 \geq 10, \\ & x_1,\ x_2,\ x_3 \geq 0. \end{aligned}$$

8. Use the strong duality theorem and complementarity slackness theorem to find a strictly complementary solution for the following problem:

$$\begin{aligned} \max \quad & z = 24x_1 + 22x_2 + 9x_3 \\ \text{s.t.} \quad & 8x_1 + 2x_2 - 3x_3 \leq 7, \\ & 4x_1 + 5x_2 + 3x_3 \leq 17, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{aligned}$$



4. The network simplex method

4.1 The minimum cost network flow problem (MCNFP)

4.1.1 Problem description

Till now our focus has been on a general LP of the form (1.2) or (2.18) without any particular structure on the parameters c, $A/\mathscr{A}$ and b. In this chapter, we will focus our attention on minimum cost network flow problems which can be modeled as a LP with a particular structure on the matrix $A/\mathscr{A}$. To make this clear, we consider a practical example from transportation.

Consider a transportation network where each node represents a city. In total, we have 6 cities in this network and we label them from 1 to 6. We need to find the cheapest way to ship prescribed amounts of a commodity from cities 1 and 2 to cities 5 and 6, possibly via cities 3 and 4. The problem can be represented using the following network:

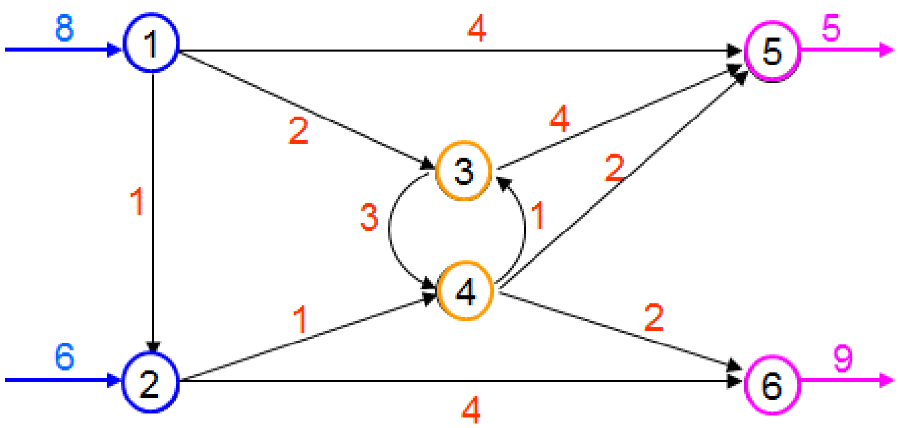


Figure 4.1: Network representation of an example of transportation problem

Nodes 1 and 2, called *supply nodes* represent the cities from where the commodities are shipped. Similarly, nodes 5 and 6 are the *demand nodes*. Nodes 3 and 4, are known as *intermediate nodes*. Now, denote by c_{ij} the unit cost of sending a commodity from city i to city j and x_{ij}, known as *flow*, represents the amount of the commodity to be shipped from origin i to destination j; see the following table for the definition of the remaining items appearing in the network above:

Nodes	i,j	$i,j=1,\dots,6$
Arcs	$e=(i,j)$	$(1,2),(1,3),(1,5),\dots$
Supply (at source)	s_i	$s_1=8,\ s_2=6$
Demand (at sink)	d_j	$d_5=5,\ d_6=9$
Supply at intermediate node	v_i	$v_3=0,\ v_4=0$
Unit costs (in red)	c_{ij}	$c_{12}=1,\ c_{13}=2,\dots$
Flow	x_{ij}	Unknown variable

Objective function

Using the above notation, the objective of our problem, as mentioned above, is to minimise the following function:

$$c_{12}x_{12}+c_{13}x_{13}+c_{15}x_{15}+c_{24}x_{24}+c_{26}x_{26}+c_{34}x_{34}+c_{35}x_{35}+c_{43}x_{43}+c_{45}x_{45}+c_{46}x_{46}.$$

Formulating the constraints

Two key assumptions are needed to formulate the constraints of the problem. The first one is that

$$\text{Total suppy = Total demand.} \tag{4.1}$$

It is clear from Figure 4.1 that this assumption is satisfied in our problem given that the total supply is $14=8+6$ (i.e., from nodes 1 and 2), while the total demand is $14=5+9$ (see nodes 5 and 6). Provided assumption (4.1) is satisfied, the following principle is used at each node to generate the constraints:

$$\text{Flow leaving - Flow entering = Supply.} \tag{4.2}$$

This leads to the following six constraints for the problem:

$$\begin{array}{cll}
1 & x_{12}+x_{13}+x_{15} & = \ \ 8,\\
2 & x_{24}+x_{26}-x_{12} & = \ \ 6,\\
3 & x_{34}+x_{35}-x_{13}-x_{43} & = \ \ 0,\\
4 & x_{43}+x_{45}-x_{46}-x_{34}-x_{24} & = \ \ 0,\\
5 & -x_{15}-x_{35}-x_{45} & = \ -5,\\
6 & -x_{26}-x_{46} & = \ -9.
\end{array}$$

Note that at the intermediate nodes 3 and 4, the supply is $s_3=s_4=0$, while it corresponds to the negative value of the demand at nodes 5 and 6; i.e., -5 and -9, respectively.
In summary, the problem is a LP, which can be put in the following canonical form (cf. (2.18)):

$$\begin{array}{ll}
\min & z=c^\top x\\
\text{s.t.} & Ax=b\\
& x\geq 0
\end{array} \tag{4.3}$$

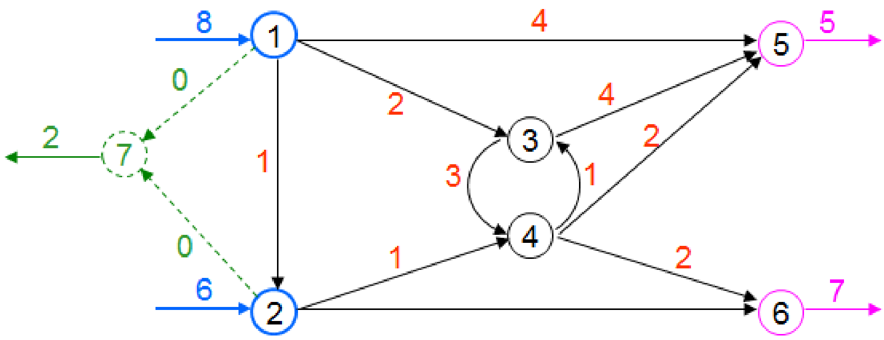
with $x = [x_{12}, x_{13}, x_{15}, x_{24}, x_{26}, x_{34}, x_{35}, x_{43}, x_{45}, x_{46}]^\top$, $c = [1, 2, 4, 1, 4, 3, 4, 1, 2, 2]^\top$,

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 8 \\ 6 \\ 0 \\ 0 \\ -5 \\ -9 \end{bmatrix}.$$

The matrix A is usually called *node-arc incidence matrix*, given that each row corresponds to a node while each column corresponds to an arc.

R The following two remarks on the problem above are very important:

- Observe that each variable occurs exactly twice: one with $+1$ as coefficient (corresponding to the leaving flow) and the other with -1 as coefficient (representing the entering flow). This is clearly reflected in the structure of the matrix A.
- If the total supply in the problem is not equal to the total supply as assumed above, then a dummy node is usually added to *balance* the problem. For the problem represented in Figure 4.1, if the demand at node 6 is reduced to 7, then the problem can be modified as follows:



The transportation problem discussed here reflects a general formulation of a minimum cost network flow problem (MCNFP) that will be solved in this chapter. Before introducing the corresponding version of the simplex method, we first discuss other more specific classes of applications.

4.1.2 Other applications of the MCNFP model

Production planning problem

During a four months planning period, the demand and production capacity relating to a product is as follows:

Months	1	2	3	4
Demand	80	200	150	200
Maximum umber produced in regular shift	100	150	140	120
Maximum umber produced in overtime shift	50	70	70	60

Regular production costs £10 per unit; overtime production costs £15 per unit. Each unit held in stock at the end of a month costs £3 for storage. The question is how to find a production plan which meets demand at minimum total cost?

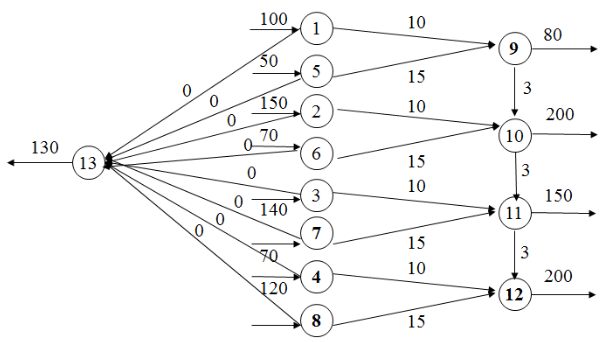
For each month i, let u_i be the production in regular shift, v_i be the production in overtime shift and s_i be the end-of-month inventory. Assuming that $s_4 = 0$, the problem can take the form:

$$\begin{array}{ll}
\min & z = 10(u_1+u_2+u_3+u_4)+15(v_1+v_2+v_3+v_4)+3(s_1+s_2+s_3) \\
\text{s.t.} & u_1+v+1-s_1 = 80, \\
& u_2+v_2+s_1-s_2 = 200, \\
& u_3+v_3+s_2-s_3 = 150, \\
& u_4+v_4+s_3 = 200, \\
& 0 \le u_1 \le 100, \\
& 0 \le u_2 \le 150, \\
& 0 \le u_3 \le 140, \\
& 0 \le u_4 \le 120, \\
& 0 \le v_1 \le 50, \\
& 0 \le v_2 \le 70, \\
& 0 \le v_3 \le 70, \\
& 0 \le v_4 \le 60, \\
& s_i \ge 0,\ i = 1,2,3.
\end{array}$$

Multiplying the first four constraints by -1 and adding slack variables t_i, $i = 1,\dots,8$, to the right-hand-side inequality constraints, we obtain the following problem:

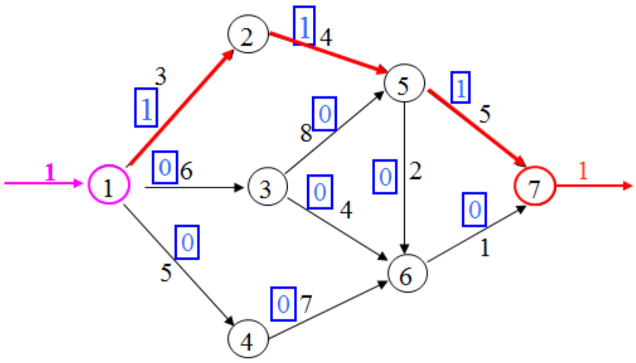
$$\begin{array}{ll}
\min & z = 10(u_1+u_2+u_3+u_4)+15(v_1+v_2+v_3+v_4)+3(s_1+s_2+s_3) \\
\text{s.t.} & -u_1-v+1+s_1 = -80, \\
& -u_2-v_2-s_1+s_2 = -200, \\
& -u_3-v_3-s_2+s_3 = -150, \\
& -u_4-v_4-s_3 = -200, \\
& u_1+t_1 = 100, \\
& u_2+t_2 = 150, \\
& u_3+t_3 = 140, \\
& u_4+t_4 = 120, \\
& v_1+t_5 = 50, \\
& v_2+t_6 = 70, \\
& v_3+t_7 = 70, \\
& v_4+t_8 = 60, \\
& -t_1-t_2-t_3-t_4-t_5-t_6-t_7-t_8 = -130, \\
& u \ge 0,\ v \ge 0, s \ge 0,
\end{array}$$

where the last equality results from adding all the previous equations together. Proceeding as in the case of the problem in Figure 4.1, we have the following network representation:



Shortest path problem

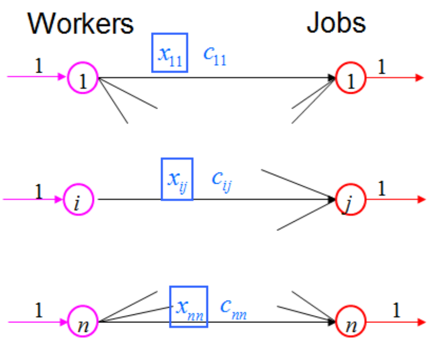
The problem of finding the shortest (in terms of time, cost, distance, etc.) path between two nodes of a given graph is a very interesting class of problem in operational research. The problem has many applications with the most notable one being the localization algorithms used in google maps or satellite navigation systems. As an illustration, consider the following network:



where we want to find the shortest path from node 1 to node 7, representing two different cities, for example. Regarding node 1 as a supply node with a supply of 1 and node 7 as a demand node with a demand of 1. Let distances be costs per unit of flow. Then, we have, for example, that path $1-2-5-7$ returns the cost $3+4+5=12$. Proceeding similarly for all the possible routes leading from node 1 to node 7 (e.g., $1-3-5-7$, $1-3-5-6-7$), we can easily identify the route/path with the smallest cost. The number of possible paths can grow exponentially depending on the size of the problem. Hence, a possible approach to deal with the problem is to use the minimum cost network flow modeling approach (4.3) described above to write down the problem as a LP problem.

Assignment problem

The assignment problem also has many applications and has been successfully applied to solve .... problems. To have a taste of the model, assume that there are n jobs to be assigned to n workers. The cost of assigning job j to worker i is c_{ij}. The aim is to find an assignment (one job per worker) that minimizes the total cost. The network representation of the problem is as follows:



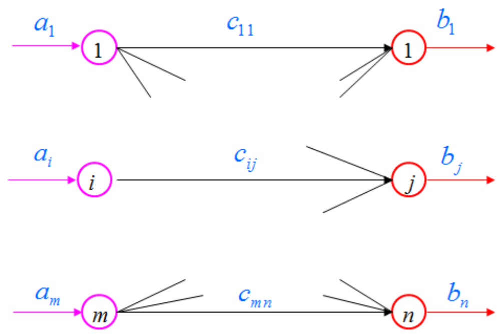
To formulate the mathematical model, let $x_{ij}=1$ if worker i does job j and $x_{ij}=0$ otherwise. Then we have the linear program

$$\begin{array}{ll} \min & z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} & \sum_{j=1}^n x_{ij} = 1,\ i = 1,\dots,n, \\ & \sum_{i=1}^n x_{ij} = 1,\ j = 1,\dots,n, \\ & x_{ij} \geq 0,\ i = 1,\dots,n,\ j = 1,\dots,n. \end{array}$$

Replacing the second set of constraints by $-\sum_{i=1}^n x_{ij} = -1,\ j = 1,\dots,n$, i.e., multiplying them by -1, we can easily check that the new problem has the form (4.3) with the corresponding matrix A having the property stated in Remark 4.1.1.

Transportation problem

A special case of the minimum cost network flow problem in which there are no intermediate nodes and all arcs are directed from supply to demand nodes.



Clearly, from assumption (4.1), we have have $\sum_{i=1}^n a_i = \sum_{j=1}^n b_j$. Proceeding as in the other cases, the problem can be written in the form (4.3). To practice the process, you can write this problem down while identifying the corresponding form of matrix A.

Converting some LPs into MCNFPs

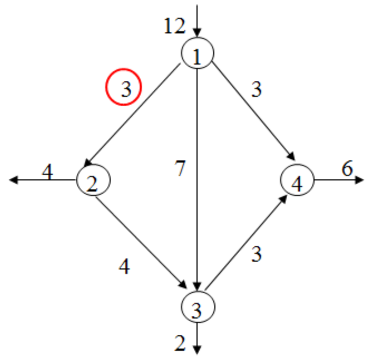
Some linear programs can be transformed into a minimum cost network flow problem (MCNFP). As illustration, consider the following example

$$\begin{array}{ll} \min & z = 6x_1 + 3x_2 + 7x_3 + 4x_4 + 3x_5 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 12, \\ & x_2 + x_3 + x_4 = 8, \\ & x_3 + x_4 - x_5 = 2, \\ & x_i \geq 0,\ i = 1,\dots,5. \end{array}$$

Let $x_1' = 2x_1$ and denote the first constraints from (1) to (3). Then, replace (2) by (2)–(1) and multiplying (3) by -1. Hence, we have the new problem

$$\begin{array}{ll} \min & z = 6x_1 + 3x_2 + 7x_3 + 4x_4 + 3x_5 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 12, \\ & -x_1' + x = -4, \\ & -x_3 - x_4 + x_5 = -2, \\ & -x_2 - x_5 = -6, \\ & x_1',\ x_i \geq 0,\ i = 2,\dots,5. \end{array}$$

This problem can easily be written in the form (4.3) and as in the previous example, it can be represented in the following network:



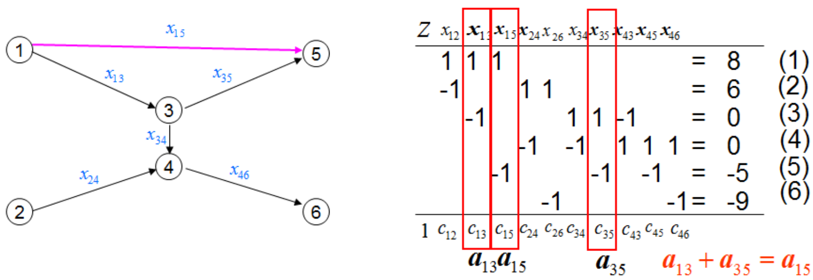
R The network simplex method is much simpler and efficient, as it will be clear in the next section. Hence, the reason to reformulate an LP into a MCNFP. It is important to note that not all LPs can be converted to MCNFPs.

4.2 The network simplex method

Before presenting the algorithm, we first present the crucial concept of a *tree*, which corresponds to a *basic feasible solution* in the context of a general LP.

Definition 4.2.1 A tree is a *connected* network with *no cycle*.

- A network is *connected* if there is at least one path connecting any pair of nodes.
- A *cycle* is a path/route in a network which begins and ends at a given node.
- For a network with n nodes, a tree will have $n-1$ arcs.
- In a tree, the addition of one extra arc produces a unique cycle.
- For a tree, the corresponding columns of the node-arc incidence matrix are linearly independent; so the corresponding arcs are acyclic. But the cycle formed by adding an extra arc forms a linearly dependent system; see the following case from the example in Figure 4.1:



The network simplex method is based on dual of the LP model of the MCNFP. Hence, to proceed, we consider the max form of problem (4.3) and its corresponding dual as

$$\begin{array}{ll} \max & Z_P = -c^\top x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad \text{and} \qquad \begin{array}{ll} \min & Z_D = b^\top y \\ \text{s.t.} & A^\top y \geq -c, \end{array}$$

respectively. In the context of the example in Figure 4.1, this leads to the following dual problem:

$$\begin{array}{ll} \min & Z_D = 8y_1 + 6y_2 - 5y_5 - 9y_6 \\ \text{s.t.} & y_i - y_j \geq -c_{ij}, \text{ for all arcs } (i,j). \end{array}$$

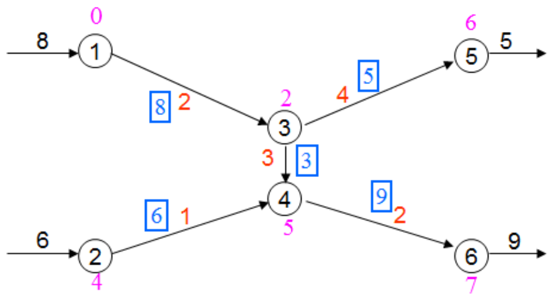
- y_i is the unit price of a commodity at node i;
- $y_i + c_{ij} - y_j$ is called the reduced unit cost for variable x_{ij};
- the objective is to maximize the total return;
- the constraints ensure that it is not possible to buy at node i and sell at node j and make a profit.

4.2.1 The phase II procedure

To illustrate the algorithm, we apply it on the problem represented in Figure 4.1.

Iteration 1

Step 1. Start with a basic feasible solution associated with a tree T. Calculate the current value of the dual variables using the information that variables corresponding to arcs in the tree solution have zero reduced unit cost.



$y_1 + 2 - y_3 = 0 \quad y_1 = 0^*, \quad y_3 = 2$

$y_2 + 1 - y_4 = 0 \quad y_2 = 4$

$y_3 + 3 - y_4 = 0 \quad y_4 = 5$

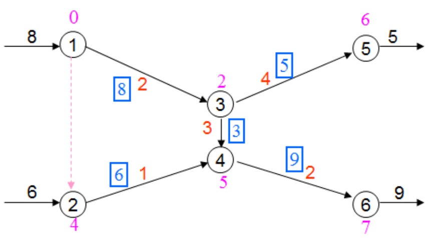
$y_3 + 4 - y_5 = 0 \quad y_5 = 6$

$y_4 + 2 - y_6 = 0 \quad y_6 = 7$

- The first dual variable y_1 is arbitrarily set to 0.
- The pink numbers on the nodes represent the calculated values of $y_1, \ldots, y_6$.
- The values on each arc in the tree represent the corresponding cost.
- In the blue boxes, we have the corresponding components of x based on the initial tree:

$$x_{15} = x_{12} = x_{26} = x_{43} = x_{45} = 0,\ x_{13} = 8,\ x_{24} = 6,\ x_{34} = 3,\ x_{35} = 5,\ x_{46} = 9.$$

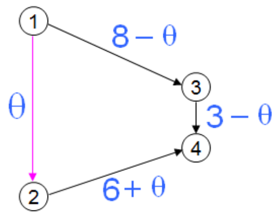
Step 2. Compute reduced unit costs for all non-basic or out of tree arcs. If all the reduced unit costs are nonnegative, the current solution is optimal. Otherwise, choose an arc with the smallest negative reduced unit cost to enter the basis.



(i,j)	$y_i + c_{ij} - y_j$
(1,2)	-3
(1,5)	-2
(2,6)	1
(4,3)	4
(4,5)	1

As -3 is the smallest negative reduced unit cost, arc $(1,2)$ enters the basis. This then creates a cycle that has to be identified to proceed with next step.

Step 3. Decide the leaving arc by sending a flow of θ around the cycle in the direction of entering arc. Choose θ as large as possible, subject to non-negativity constraints on the flows.

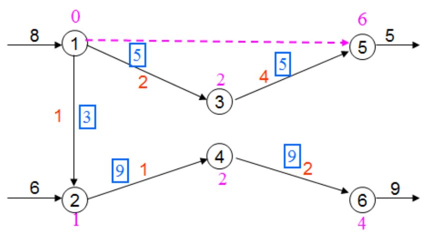


$(1,2)$ entering the basis leads to the above cycle. Because of the non-negativity constraints on the flows, the only possible choice for θ is $\theta = 3$. Hence, the leaving arc is $(3,4)$.

Step 4. Iterate until all the reduced unit cost are nonnegative (Based on the duality theory from Chapter 3, this implies that an optimal solution has been found.)

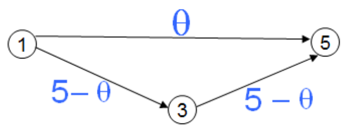
Iteration 2

Repeating the process from steps 1–3 above, we have the following tree (considered without arc $(1,5)$) and the corresponding table of reduced unit costs:



(i,j)	$y_i + c_{ij} - y_j$
(1,5)	-2
(2,6)	1
(3,4)	3
(4,3)	1
(4,5)	-2

Given that arcs $(1,5)$ and $(4,5)$ have the same reduced unit costs, we arbitrarily chose $(1,5)$ as entering arc. This leads to the following cycle:

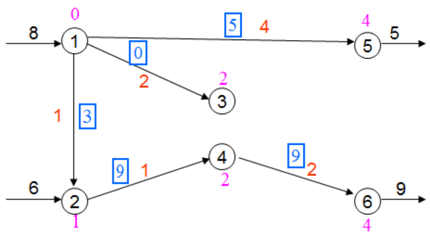


Hence, $\theta = 5$ and arc $(3,5)$ is arbitrarily chosen as leaving arc.

R We do not need to recalculate all the dual variables, e.g., following the same path as in Step 1 of the previous iteration, it would be clear that y_1, y_3 and y_5 remain unchanged.

Iteration 3

Repeating the process again, we have



(i,j)	$y_i + c_{ij} - y_j$
(2,6)	1
(3,4)	3
(3,5)	2
(4,3)	1
(4,5)	0

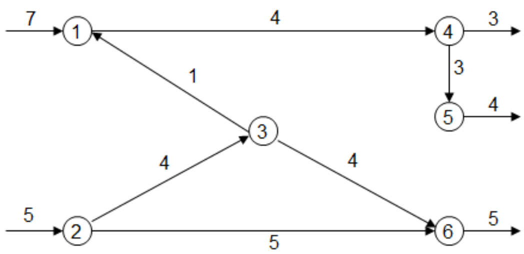
No negative reduced unit costs, so current solution tree is optimal.

The minimum cost $z = 3 \times 1 + 0 \times 2 + 5 \times 4 + 9 \times 1 + 9 \times 2 = 50$.

R The zero reduced cost, arc $(4,5)$, suggests an alternative optimal solution.

4.2.2 The phase I procedure

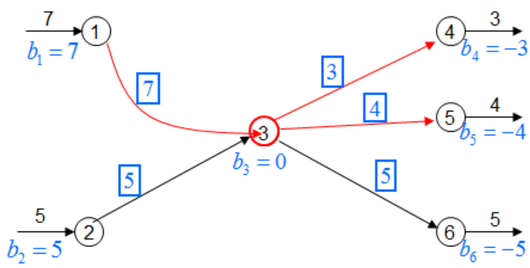
As stated in step 1 of phase 2 of the network simplex method, a basic feasible solution associated with a tree is necessary. The following algorithm allows us to compute such a point. To illustrate how the algorithm works, we consider the following example:



Step 1. Let b_i be the supply at node i, where $b_i < 0$ if i is a demand node. For our example, we obviously have $b_1 = 7$, $b_2 = 5$, $b_3 = 0; b_4 = -3$, $b_5 = -4$ and $b_6 = -5$.
Step 2. Choose any node u and form a tree solution by setting flows as follows:

- For each node i, where $b_i > 0$ and $i \neq u$, set a flow of bi in arc (i,u);
- For each node j, where $b_j \leq 0$ and $j \neq u$, set a flow of $-b_j$ in arc (u,j).

Taking $u = 3$, we can form the following tree solution based on this rule:

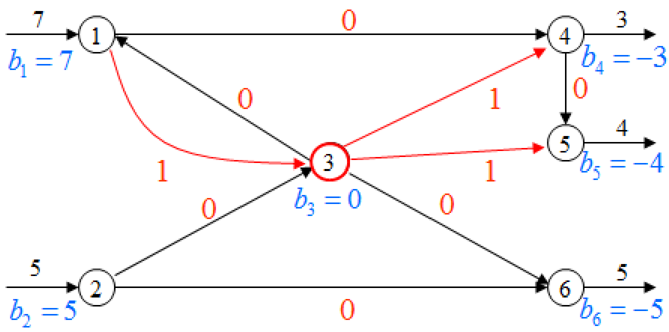


If any of these arcs (i,u) and (u,j) do not exist in the original network, then they are called artificial arcs.

Step 3. This leads to a new problem, where the objective is to minimize $z' = \sum_i \sum_j c'_{ij} x_{ij}$ with the new costs defined by

$$c'_{kl} = \begin{cases} 1 & \text{if } (k,l) \text{ is an artificial arc;} \\ 0 & \text{if } (k,l) \text{ is an original arc.} \end{cases}$$

For the example above, we get the following graph formulation of the problem:



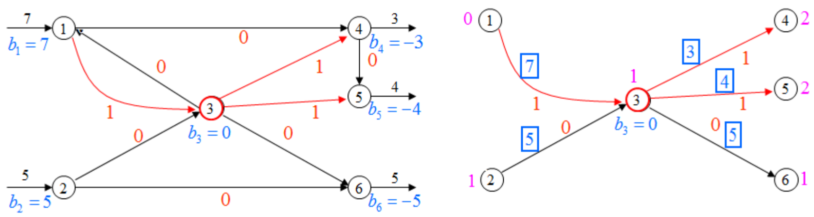
Step 4. Solve the problem using the network simplex method (described in Phase II) and suppose we obtain z as the value of the optimal solution. Then we have one of the following options:

- If $z' > 0$, then the original problem is infeasible;
- If $z' = 0$, and all artificial arcs are non-basic, then a feasible tree solution is found. We proceed with phase II to solve the original problem;
- If $z' = 0$, but an artificial arc is basic, then the original problem is decomposed into sub problems, which are solved separately.

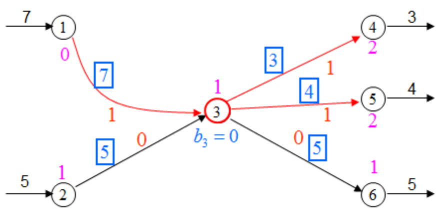
Next, we provide a detailed implementation of the Simplex Method - Phase I on the example above.

Iteration 1

Step 1. Form the tree solution and calculate the current value of the dual variables using the information that variables corresponding arcs in the tree solution have zero reduced unit cost.

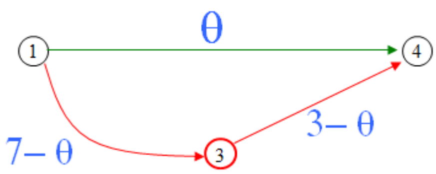


Step 2. Compute reduced unit costs for all non-basic or out of tree arcs. If all the reduced unit costs are nonnegative, the current solution is optimal. Otherwise, choose an arc with the smallest negative reduced unit cost to enter the basis.



(i,j)	$y_i + c'_{ij} - y_j$
(1,4)	-2
(3,1)	1
(4,5)	0
(2,6)	0

Step 3. Decide the leaving arc by sending a flow of θ round the cycle in the direction of entering arc. Choose θ as large as possible, subject to non-negativity constraints on the flows.

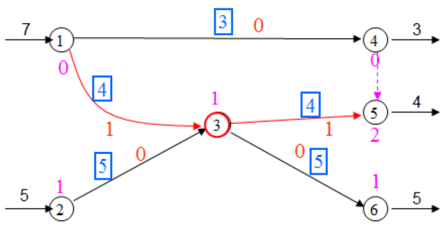


The leaving arc is $(3,4)$. Since this is an artificial arc, it can be removed.

Step 4. Iterate until all the reduced unit cost are nonnegative.

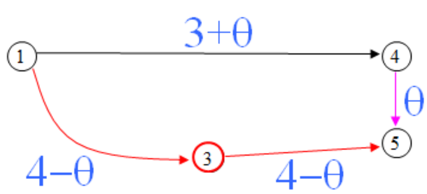
Iteration 2

Repeating the process above, we successively have the following:



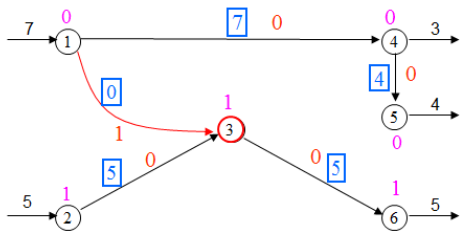
(i,j)	$y_i + c'_{ij} - y_j$
(3,1)	1
(4,5)	-2
(2,6)	0

Obviously, the entering arc is $(4,5)$, leading to this corresponding cycle on the right. We arbitrarily chose $(3,5)$ as leaving arc.



Iteration 3

Proceeding similarly to the previous iteration, we have the following tree solution and corresponding table of reduced unit costs:



(i,j)	$y_i + c'_{ij} - y_j$
(3,1)	1
(2,6)	0

Thus we obtain $z' = 0$, but with an artificial arc $(1,3)$ in the tree solution.
Use the y's to partition nodes into sets:

$$S = \{1,4,5\} \quad T = \{2,3,6\}$$

The problem decomposes because:
(a) For the nodes of T, total supply is equal to total demand;
(b) There are no arcs from nodes of S to nodes of T.

Theorem 4.2.1 (**Decomposition theorem**) If phase1 ends with a tree solution containing artificial arcs which each have a zero flow, then the problem decomposes.

Proof. Let (u, v) be an artificial arc in the tree solution. Let

$$S = \{k | \, y_k \le y_u\} \text{ and } T = \{k | \, y_k > y_u\}.$$

Since $u \in S$ and $v \in T$ $(y_v = y_u + c'_{uv} = y_u + 1)$, S and T are non-empty.

There are no original arcs (i, j), where $i \in S$ and $j \in T$ since for such arc we would have $y_i + c'_{ij} - y_j = y_i - y_j < 0$, which is not possible at the end of phase 1.

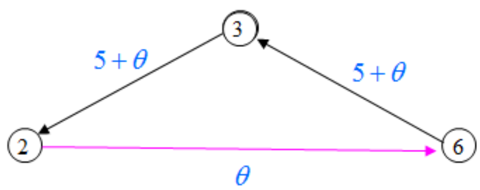
For any original arc (i, j), where $i \in S$ and $j \in T$, we have $y_i + c'_{ij} - y_j = y_i - y_j > 0$.

So It cannot be in the tree solution; thus, it has zero flow. Since all artificial arcs have zero flow, there is no flow from nodes of S to nodes of T, or from nodes of T to nodes of S. Therefore, for nodes of T (and S), the total supply is equal to the total demand. We see that the two conditions from decomposition are satisfied. ■

4.2.3 Implementation strategies

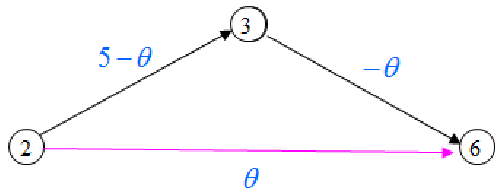
Detecting unbounded problems

Having found an entering arc, and the corresponding cycle, if there are no $-\theta$ terms in the adjusted flow, the problem is unbounded. (The cycle has a negative cost per unit of flow.)



Degeneracy

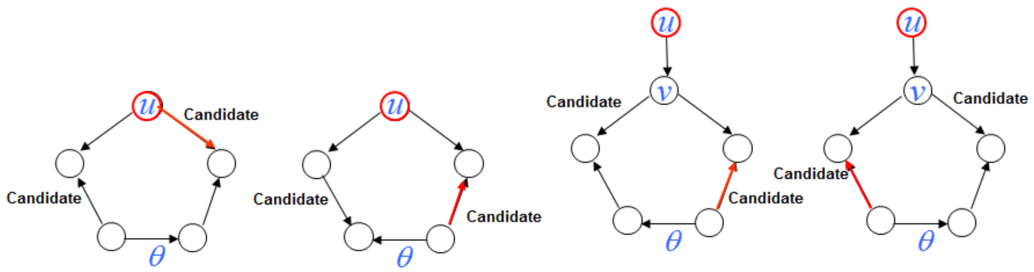
Even though one may tend to think of each iteration as reducing the value of the objective function, this is not always the case: sometimes $\theta = 0$ is forced in the new feasible solution.



Cycling

Cunningham shows that the following approach avoids cycling.

- Node u (of phase 1) is a fixed root node.
- Choose an entering arc in the usual way and find the corresponding cycle if u lies on the cycle, set $v = u$; otherwise choose node v to be the node where a path from node u joins the cycle.
- Moving round the cycle in the direction of the entering arc, starting from node v, choose the first candidate leaving arc.



Computing values of the dual variables

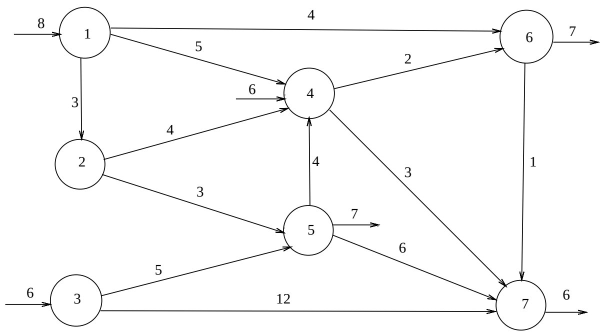
Update $y_1, \dots y_n$ from their values at the previous iteration, rather than compute them from scratch.

Integrality of the solution

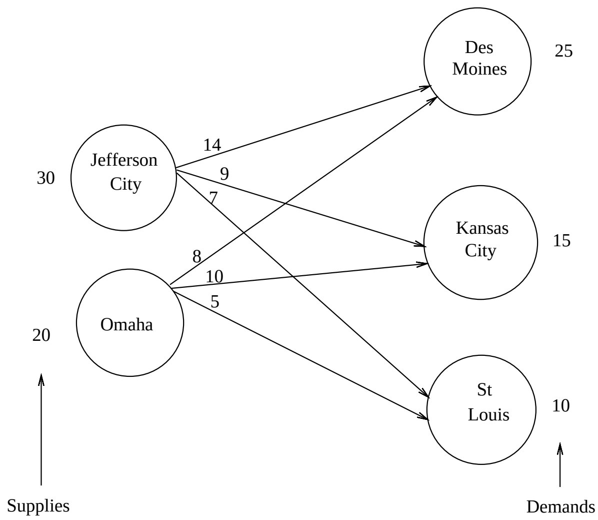
Suppose all supplies and demands are integers. Then the initial phase 1 solution is integer-valued. Since θ is an integer (it is the value of a flow in some arc), flows remain integer-valued throughout the algorithm.

4.3 Exercises

1. Formulate the following network problem as a linear program:

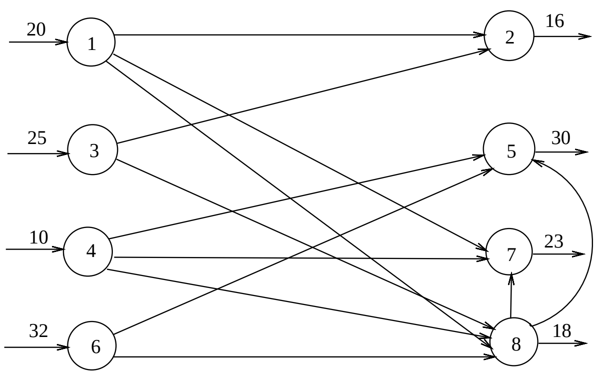


2. Consider the following network representation of a transportation problem:



Develop a linear programming model for the transportation cost minimization problem.

3. Consider Phase II of the network simplex method.
   (a) Discuss the analogies that you find between this phase of the method and the simplex method for a general linear program discussed in the previous chapters of the course.
   (b) Present the simplifications that appear in the network simplex method.
4. Consider the network problem



where

$$c_{17} = 8,\ c_{18} = 7,\ c_{12} = 2,\ c_{32} = 6,\ c_{38} = 2,$$
$$c_{47} = 6,\ c_{48} = 5,\ c_{45} = 8,\ c_{85} = 7,\ c_{65} = 9,\ c_{68} = 8,\ c_{87} = 0.$$

Starting with a solution in which $x_{18}, x_{38}, x_{48}, x_{85}, x_{68}$ and x_{87} take positive values, and the last constraint is satisfied as a strict inequality, use the network simplex method to find the minimum value of z.

5. Show that the following linear programming problem can be formulated as a minimum cost network flow problem:

$$\begin{array}{ll} \text{Minimize} & z = 5x_1 + 8x_2 + 11x_3 + 10x_4 + 4x_5 + 9x_6 + 6x_7 + 8x_8 + 7x_9 \\ \text{subject to} & x_1 + x_2 = 15 \\ & x_2 + x_3 + x_4 = 20 \\ & x_4 + x_5 = 12 \\ & x_6 + x_7 + x_8 = 27 \\ & x_8 + x_9 = 14 \\ & x_3 + x_7 \leq 18 \\ & x_1, \ldots, x_9 \geq 0. \end{array}$$

Starting with a solution in which x_1, x_2, x_4, x_7 and x_8 take positive values, and the constraint $x_3 + x_7 \leq 18$ is satisfied as a strict inequality, use the network simplex method to solve the problem.

6. Solve the first problem of Problem Sheet 8 using the two phase network simplex method.
7. Ace Manufacturing has orders for three similar products:

Product	**Orders (units)**
A	2000
B	500
C	1200

Three machines are available. Machine capacities for the next week, and the unit costs, are as follows:

Machine	**Capacity (units)**
1	1500
2	1500
3	1000

Machine	**Product**		
	A	B	C
1	£1.00	£1.20	£0.90
2	£1.30	£1.40	£1.20
3	£1.10	£1.00	£1.20

Formulate a linear programming model to minimize the cost and show that it is a minimum cost network flow problem.

8. Scott and Associates, Inc., is an accounting firm that has three new clients. Project leaders will be assigned to the three clients. Based on the leaders' different backgrounds and experience, the various leader-client assignments differ in term of projected completion times. The possible assignments and the estimated completion times in days are given in the following table.

Project Leader	**Client** 1	2	3
Jackson	10	16	32
Ellis	14	22	40
Smith	22	24	34

(a) Develop a network representation of this problem.
(b) Formulate the problem as a linear program.

9. The distribution system for the Herman Company consists of three plants, two warehouses, and four customers. The following tables how plant capacity and the shipping cost (in £) from each plant to each warehouse, and customer demand and shipping costs per unit (in £) from each warehouse to each customer.

	Warehouse		
Plant	1	2	**Capacity**
1	4	7	450
2	8	5	600
3	5	6	380

	Customer			
Warehouse	1	2	3	4
1	6	4	8	4
2	3	6	7	7
Demand	300	300	300	400

(a) Develop a network model for this problem.
(b) Formulate a linear programming model for the problem.

5. Integer programming

In the previous chapters, we have discussed only problems where the components of the variables can take any value from the real line. There are however many classes of optimization problems where this cannot be possible. Integer (linear) programming problems arise when variables correspond to numbers of workers, machines, etc., which cannot be split. Also, it is possible to model *do/don't* decisions by variables, which take the values zero or one. The main classes of of such problems include: (a) *Pure integer programming problems*, where all variables are constrained to take integer values; (b) *Mixed integer programming problems*, where some variables are constrained to take integer values; and (c) *Zero-one programming problems*, where all variables take the values zero or one. After presenting a few applications of integer programming (Section 5.1), we will discuss connections with linear programming problems studied in the previous chapters (Section 5.2). Next, we will introduce one of the main solution methods for integer programming problems and implement special versions of the method on Knapsack problems and zero-one-type problems (Section 5.3).

5.1 Some applications

5.1.1 The Knapsack problem

The Knapsack problem is a very interesting class of problem in its own right and has been applied in various areas. It refers to the common problem of problem of packing your most valuable items in a knapsack/rucksack without overloading your luggage. To illustrate its usefulness, we introduce an application in portfolio optimization.

A company has £b to invest on a selection of n projects. Each project i requires an investment of £a_i and gives a return of £c_i. How the money should be invested to maximize the total profit? Let

$$x_i = \begin{cases} 1 \text{ if the company invests in project } i, \\ 0 \text{ otherwise.} \end{cases}$$

Then it suffices to solve the following optimization problem, where the components of the variable

x can take only the value 0 or 1:

$$\begin{array}{ll} \max & z = \sum_{i=1}^{n} c_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} a_i x_i \le b, \\ & x_i \in \{0,1\}, i = 1,\dots,n. \end{array}$$

5.1.2 The plant location problem

Consider a supplier who serves n towns. Depots can be sited in any of the towns. For each town i, the fixed annual cost of running a depot is b_i, and the cost of supplying a customer at town j is c_{ij}. The question is: *where should the depots be located to minimize the costs*? To model the problem, define variables by

$$y_i = \begin{cases} 1 \text{ if a depot is located in town } i, \\ 0 \text{ otherwise,} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1 \text{ if a customer in town } j \text{ is supplied from a depot in town } i, \\ 0 \text{ otherwise.} \end{cases}$$

To find the optimal locations of the depots, the following problem should be solved:

$$\begin{array}{ll} \min & z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{n} b_i y_i \\ \text{s.t.} & \sum_{i=1}^{n} x_{ij} = 1, \; j = 1,\dots,n, \\ & x_{ij} \le y_i, \; i,j = 1,\dots,n, \\ & x_{ij} \in \{0,1\}, \; i,j = 1,\dots,n, \\ & y_i \in \{0,1\}, i = 1,\dots,n. \end{array}$$

5.1.3 Modeling of specific type of conditions

A number of specific conditions often appear when modeling the feasible set of many mathematical programming problems. Here we consider the “Either-Or Constraints” and “If-Then Constraints” and show how they can be reformulated in to standard forms of constraints using integer variables.

Either-or constraints

We are given two constraints of the form:

$$f(x) \le 0, \tag{5.1}$$

$$g(x) \le 0, \tag{5.2}$$

where f and g are real-valued functions. We want to ensure that at least one of (5.2) and (5.2) is satisfied, often called *either-or constraints*. Adding the following two constraints to the conditions above will ensure that at least one of (5.2) and (5.2) is satisfied:

$$f(x) \le My, \quad g(x) \le M(1-y).$$

Here, y is a $0-1$ variable, and M is a number chosen large enough to ensure that $f(x) \le M$ and $g(x) \le M$ hold for all values of x that satisfy the other constraints in the problem.

If-then constraints

Suppose we want to ensure that $f(x) \ge 0$ implies that $g(x) \ge 0$. Then, we can include the following constraints in the formulation:

$$f(x) \le M(1-y), \quad -g(x) \le My,$$

where similarly to the previous case, y is also a $0-1$ variable, and M is a number chosen large enough to ensure that $f(x) \le M$ and $-g(x) \le M$ for all values of x that satisfy the other constraints in the problem.

5.2 Relationships to linear programming

5.2.1 Linear programming relaxation

Most solution methods to solve integer linear programming problems rely on the corresponding linear programming relaxation. Hence, before we discuss solution algorithms, it is important to look at some key links betweens integer linear programs (IP) and their linear programming (LP) relaxations. To proceed, we consider the problem

$$\begin{array}{rl} \max/\min & z = c^\top x \\ \text{s.t.} & Ax = b, \\ & x \geq 0 \text{ integer}, \end{array}$$

where "$x \geq 0$ integer" means that each component of the vector x is non-negative and integer-valued. Then, the LP relaxation of this problem is

$$\begin{array}{rl} \max/\min & z = c^\top x \\ \text{s.t.} & Ax = b, \\ & x \geq 0. \end{array}$$

Theorem 5.2.1 We have the following relationships between (IP) and (LP):

1. If (IP) is a minimization, the optimal objective value for (LP) is less than or equal to the optimal objective for (IP), i.e., it is a lower bound.
2. If (IP) is a maximization, the optimal objective value for (LP) is greater than or equal to that of (IP), i.e., it is an upper bound.
3. If (LP) is infeasible, then so is (IP).
4. If (LP) is optimized by integer variables, then that solution is feasible and optimal for (IP).
5. If the objective function coefficients are integer, then for minimization (resp. maximization), the optimal objective for (IP) is greater than or equal to the "round up" (resp. less than or equal to the "round down") of the optimal objective for (LP).

Proof. Denote by $\mathscr{F}^{IP}$ and $\mathscr{F}^{LP}$ the feasible sets of (IP) and (LP), respectively. Similarly, let z_{IP} and z_{LP} represent the optimal values of problems (IP) and (LP), respectively. Given that (LP) is less constrained than (IP), we obviously have

$$\mathscr{F}^{IP} \subseteq \mathscr{F}^{LP}. \tag{5.3}$$

1. Denote by $\bar{x}$ and $\underline{x}$ an optimal solution for (LP) and (IP), respectively. We have $z_{LP} = c^\top \bar{x} \leq c^\top x$ for all $x \in \mathscr{F}^{LP}$. From (5.3), it follows that $\underline{x} \in \mathscr{F}^{LP}$. Hence, $z_{LP} = c^\top \bar{x} \leq c^\top \underline{x} = z_{IP}$.
2. Proceeding in a way similarly to 1, we can show that $z_{IP} \leq z_{LP}$ in the maximization case.
3. Obviously follows from (5.3).
4. Let us suppose that we are dealing here with a maximization problem and let $\bar{x}$ be an optimal solution of (LP). We have $z_{LP} = c^\top \bar{x} \geq c^\top x$ for all $x \in \mathscr{F}^{LP}$. It follows from (5.3) that the latter holds for all $x \in \mathscr{F}^{IP}$. Now assuming that $\bar{x} \in \mathscr{F}^{IP}$, it holds that $z_{LP} = c^\top \bar{x} = z_{IP}$.
5. We proceed only with the minimization case as the maximization one can be obtained in a similar way. It follows from 1 that $z_{LP} \leq z_{IP}$. We assume that z_{LP} is fractional. Hence, there exists a real number $0 < \varepsilon < 1$ such that $z_{LP} = a - \varepsilon$, where a is the value obtained by rounding up z_{LP}. We claim that $a \leq z_{IP}$. Otherwise, we have $a - 1 < a - \varepsilon \leq z_{IP} < a$. This is impossible given that z_{IP} is an integer, as the coefficients of the objective function are integers.

■

Another interesting aspect of the link between (LP) and (IP) is that when we think about both problems naively, it sounds like just solving the former and taking the closest integer optimal solution (by rounding up or down) would give an optimal solution to the latter. The following example shows that this is usually not true:

$$\begin{array}{ll} \max & z = x_1 + 5x_2 \\ \text{s.t.} & x_1 + 10x_2 \leq 20, \\ & x_1 \leq 2, \\ & x_1,\, x_2 \geq 0. \end{array} \tag{5.4}$$

The optimal solution of the (LP) form of (5.4) is $(2, 1.8)$ and the corresponding optimal value is

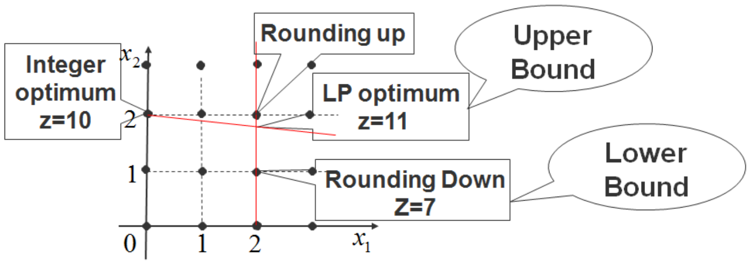


Figure 5.1: Comparing the optimal solution of (LP) and (IP) for example (5.4)

$z_{LP} = 11$. Rounding up or down the optimal solution, we respectively get the points $(2, 1)$ and $(2, 2)$ with optimal values 7 and 12, respectively. None of them corresponds to the optimal value of the (IP) version of problem (5.4), which is 10 with optimal solution $(0, 2)$.
It is however important to mention that for problem with large variables, rounding the optimal solutions of the linear relaxation remains a viable approach as slight changes might be seen as insignificant for the decision maker. The main challenge with such an approach is that it may not be straightforward to round the fractional variables so that all constraints of the problem are satisfied. Such techniques are out of the scope of this lecture. Instead, for the remaining part of this notes, we will focus our attention on the branch and bound method to obtain an optimal solution for integer programs. This is very efficient, although it may have a very large computation time requirements.

5.2.2 Unimodularity

For some IPs, the optimal solution to the LP relaxation will also be the optimal solution to the IP. Suppose the constraints of the IP are written as $Ax = b$. If the determinant of every square submatrix of A is $+1$, -1, or 0, we say that the matrix A is *unimodular*. If A is unimodular and each element of b is an integer, then the optimal solution to the LP relaxation will assign all variables integer values and will therefore be the optimal solution to the IP. It can be shown that the constraint matrix of any MCNFP is unimodular. Hence, as we saw in the previous chapter (cf. integrality), any MCNFP in which each node's net outflow and each arc's capacity are integers will have an integer-valued solution.
As a general rule, the more an IP looks like an MCNFP, the easier the problem is to solve by branch-and-bound methods. Thus, in formulating an IP, it is good to choose a formulation in which

as many variables as possible have small coefficients, e.g., of $+1$, -1, and 0. To illustrate this consider the following illustration of the feasible set of a problem:

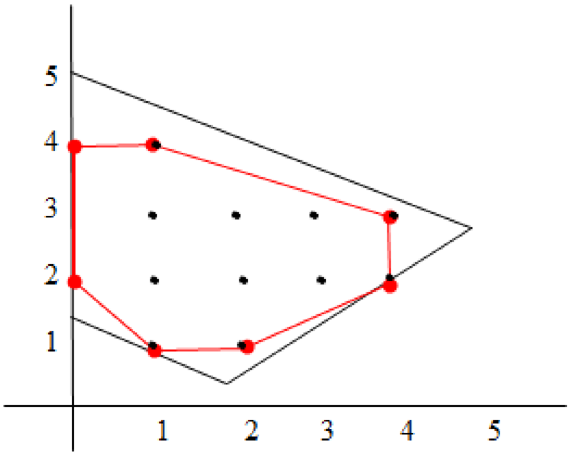


Figure 5.2: The bold lines are delimiting the feasible set of the LR problem. The red constraints show the tightest possible formulation for which an integer solution is guaranteed.

5.3 Branch and bound methods

The branch and bound method is one of the main techniques to solve integer programming problems. The method find an optimal solution by efficiently enumerating points in the subproblem's feasible region. The process exploits the link between the IP and its linear programming relaxation as described in Theorem 5.2.1.

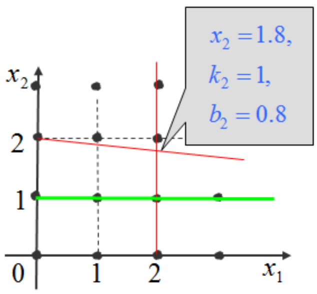
5.3.1 General branch and bound method for pure IP

Assuming that we are solving a maximization problem, at each node of the search tree, an *upper bound* is computed by solving the linear programming relaxation. If this solution is integer valued, it gives a *lower bound* on the optimal value of the objective. Otherwise, some x_j, which is required to be an integer, takes a fractional value $k_j + b_j$, where k_j is an integer and $0 < b_j < 1$.

Branching creates two new nodes:

- For the first, the constraint $x_j \leq k_j$ is added;
- For the second, the constraint $x_j \geq k_{j+1}$ is added.

For example (5.4), the box in the picture gives the values of k_2 and b_2 for the fractional component.



Besides the above branching rule, there are four strategies that allow the implementation of the branch and bound method (with the first one already mentioned in the introduction of this subsection):

(S1) *Bounding rule*: Assuming that we are solving a maximization problem, at each node of the search tree, an *upper bound* is computed by solving the linear programming relaxation. If this solution is integer valued, it gives a *lower bound* on the optimal value of the objective.

(S2) *Choosing a branching variable*: One way to choose a branching variable, is to select one which is furthest from taking an integer value.

(S3) *Selecting the node to explore next*: There are two strategies usually applied:
- The depth-First Search approach which consists to choose one furthest down the search tree (this economizes on storage space)
- The best-first search approach which consists to choose the node with the largest upper bound, in case of a maximization problem.

(S4) *Fathoming a node*: A node is fathomed when it can be discarded–this is usually in one of the following three cases:
- the linear programming relaxation is infeasible;
- the solution of the linear programming relaxation is integer valued and is not greater than the current lower bound;
- the upper bound is not greater than the current lower bound.

■ **Example 5.1** To illustrate the branch and bound method on a pure integer programming problem, we apply it on the following problem:

$$\begin{array}{ll} \max & z = 3x_1 + 3x_2 + 13x_3 \\ \text{s.t.} & -3x_1 + 6x_2 + 7x_3 \leq 9, \\ & 5x_1 - 2x_2 + 6x_3 \leq 7, \\ & x_1,\, x_2,\, x_3 \geq 0. \text{ are integers.} \end{array} \tag{5.5}$$

Node 0. At the first node, we solve the LP relaxation of the original problem:

$$\begin{array}{ll} \max & z = 3x_1 + 3x_2 + 13x_3 \\ \text{s.t.} & -3x_1 + 6x_2 + 7x_3 \leq 9, \\ & 5x_1 - 2x_2 + 6x_3 \leq 7, \\ & x_1,\, x_2,\, x_3 \geq 0. \end{array} \tag{5.6}$$

This can also be labeled as sub-problem 0. Solving problem (5.6) with the simplex method described in Chapter 2, we get the optimal solution $X = (0, 0.10, 1.20)$ and the corresponding optimal value $Z = 15.90$ represents an upper bound for (5.5). Based on (S2) above, branching is done on x_3. This leads to 2 new subproblems; cf. nodes 1 and 2.

Node 1. At node 1, Subproblem 1 is obtained by adding constraint $x_3 \leq 1$ to problem (5.6). Solving this problem by the simplex method, we get $x = (0.41, 0.54, 1)$ with $Z = 15.88$.

Node 2. In an attempt to use the best-first search, we also attempt to solve Subproblem 2, obtained by adding $x_3 \geq 2$ to problem (5.6). But this problem is infeasible. Hence, the node is fathomed.

Node 3. Subproblem 3 is obtained by adding constraint $x_3 \leq 1$ and $x_2 \leq 0$ to problem (5.6). Solving this problem by the simplex method, we get $x = (0.20, 0, 1)$ with $Z = 13.60$.

Node 4. Subproblem 4 is obtained by adding constraint $x_3 \leq 1$ and $x_2 \geq 1$ to problem (5.6). Solving this problem by the simplex method, we get $x = (0.85, 1, 0.79)$ with $Z = 15.85$. Based on the best-first strategy, we first branch from this node. The branching strategy implies the choice of x_3.

Node 5. Subproblem 5 is obtained by adding constraint $x_3 \le 1$ and $x_2 \ge 1$ and $x_3 \le 0$ to problem (5.6). Solving this problem by the simplex method, we get $x = (2.50, 2.75, 0)$ with $Z = 15.75$.

Node 6. The subproblem here is obtained by adding the constraints $x_3 \le 1$, $x_2 \ge 1$ and $x_3 \ge 1$ to problem (5.6). The problem is infeasible.

Node 7. By the depth-first search, we now move to this node with Subproblem 7 obtained by adding constraint $x_3 \le 1$ and $x_2 \ge 1$, $x_3 \le 0$ and $x_1 \le 2$ to problem (5.6). Solving this problem by the simplex method, we get $x = (2, 2.50, 0)$ with $Z = 13.50$.

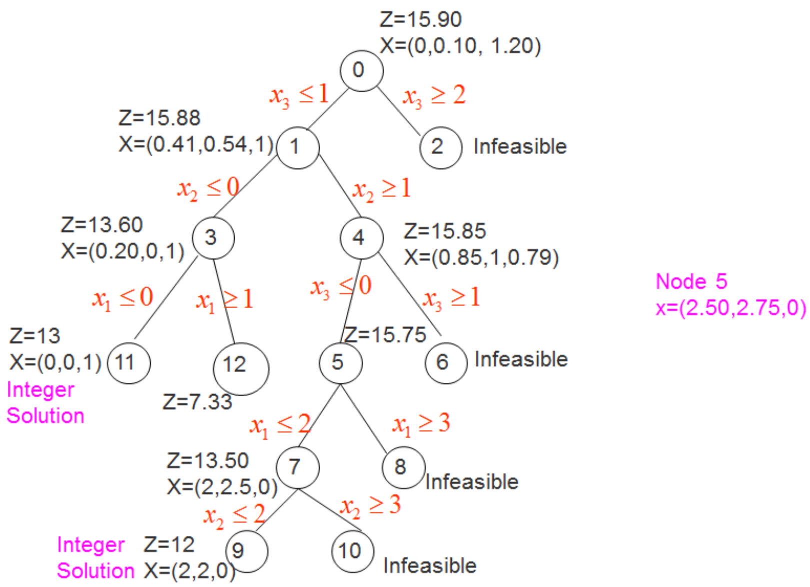
Node 8. The subproblem here is obtained by adding the constraints $x_3 \le 1$, $x_2 \ge 1$, $x_3 \le 0$ and $x_1 \ge 3$ to problem (5.6). The problem is infeasible.

Node 9. With the depth-first search, we consider Subproblem 9, which is obtained by adding constraint $x_3 \le 1$ and $x_2 \ge 1$, $x_3 \le 0$, $x_1 \le 2$ and $x_2 \le 2$ to problem (5.6). Solving this problem, we get $x = (2, 2, 0)$ with $Z = 12$. This node is fathomed based on item 2 of (S4).

Node 10. Subproblem here is obtained by adding the constraints $x_3 \le 1$ and $x_2 \ge 1$, $x_3 \le 0$, $x_1 \le 2$ and $x_2 \ge 3$ to problem (5.6). The problem is infeasible.

Node 11. Backtracking to node 3, we get Subproblem 11 by adding constraint $x_3 \le 1$, $x_2 \le 0$ and $x_1 \le 0$ to (5.6). Solving it, we get $x = (0, 0, 1)$ with $Z = 13$ and the node is fathomed.

Node 12. Subproblem 12 is obtained by adding constraint $x_3 \le 1$, $x_2 \le 0$ and $x_1 \ge 1$ to problem (5.6). Solving this problem by the simplex method, we get $Z = 7.33$ and the node is fathomed.



It is clear from the branch and bound tree that the optimal solution of the problem is $x = (0, 0, 1)$ with the optimal value being $Z = 13$. ∎

5.3.2 Branch and bound method for the knapsack problem

In this section, we focus our attention on a branch and bound method tailored to the Knapsack problem introduced in Section 1. Here, denote by $c_i = v_i$ and $a_i = w_i$ the value and weight of each item, respectively, while $b = W$ is the total capacity of the Knapsack:

$$\begin{array}{ll} \max & z = \sum_{i=1}^{n} v_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} w_i x_i \leq W, \\ & x_i \in \{0,1\},\ i = 1,\dots,n. \end{array} \tag{5.7}$$

Instead of implementing the general branch and bound rules discussed in the previous section, in the context of pure integer problems, we consider the following more efficient ones:

(K1) *Bounding rules*: Upper and lower bounds are obtained as follows:
   - *Upper bounding rule*: relax the problem by allowing variables to be fractional (see below how the relaxed problem is solved).
   - *Lower bounding rule*: optimal value of the objective function of LP relaxation or round down any fractional variables in its optimal solution.

(K2) *Branching rule*: select an item, j (see below for a selection method), and force the item either to be included or excluded from the knapsack.

(K3) *Selecting the node to explore next*: Same as in the pure integer case; i.e., branch from a node with the largest upper bound (dept first strategy), breaking ties by choosing a node furthest down the search tree (best first strategy).

(K4) *Fathoming a node*: Under the same conditions as in (S4) of previous subsection.

To obtain the upper bound, we start by indexing the items in the problem in such a way that the following condition is satisfied:

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \dots \geq \frac{v_n}{w_n}.$$

Then choose index j such that items $1,\dots,j-1$ all fit within the knapsack, but items $1,\dots,j$ will not all fit. This leads to the following upper bound.

$$UB = v_1 + \dots + v_{j-1} + v_j(W - w_1 - \dots - w_{j-1})/w_j.$$

To prove that UB is an upper bound of problem (5.7), it suffices to show that the number corresponds to an optimal solution of the linear relaxation problem. The conclusion is based on the connections between IP and LP discussed in Theorem 5.2.1 and the strong duality result.

Proof. The linear programming relaxation of problem (5.7) and its dual are obtained as

$$\begin{array}{ll} \max & z = \sum_{i=1}^{n} v_i x_i \\ \text{s.t.} & \sum_{i=1}^{n} w_i x_i \leq W, \\ & 0 \leq x_i \leq 1,\ i = 1,\dots,n, \end{array} \quad \text{and} \quad \begin{array}{ll} \min & z = Wu + \sum_{i=1}^{n} y_i \\ \text{s.t.} & w_i u + y_i \geq v_i,\ i = 1,\dots,n, \\ & u \geq 0,\ y_i \geq 0,\ i = 1,\dots,n. \end{array} \tag{5.8}$$

One can easily check that the following point is an optimal solution of the linear relaxation problem

$$x_1 = \dots = x_{j-1} = 1,\ \ x_j = (W - w_1 - \dots - w_{j-1})/w_j,\ \ x_{j+1} = \dots = x_n = 0. \tag{5.9}$$

This point leads to the following optimal value for the relaxation problem:

$$z = v_1 + \dots + v_{j-1} + v_j(W - w_1 - \dots - w_{j-1})/w_j.$$

As for the dual problem, an optimal solution is given by

$$\begin{aligned}
&y_i = v_i - w_i v_j / w_j \geq 0, \;\; i = 1, \ldots, j-1, \\
&y_i = 0, \;\; i = j \ldots, n, \\
&u = v_j / w_j.
\end{aligned}$$

We can easily verify that this point is feasible for the dual of the relaxation problem in (5.8):

$$\begin{aligned}
&w_i u + y_i = w_i v_j / w_j + v_i - w_i v_j / w_j = v_i \text{ for } i = 1, \ldots, j-1, \\
&w_j u + y_j = w_j v_j / w_j + 0 = v_j, \\
&w_i u + y_i = w_i v_j / w_j + 0 \geq v_i \text{ for } i = j+1, \ldots, n.
\end{aligned}$$

Thus, the solution is feasible. Moreover, the objective function value is given by

$$\begin{aligned}
z_D &= v_j W / w_j + v_1 + \ldots + v_{j-1} - v_j (w_1 + \ldots + w_{j-1}) / w_j \\
&= v_1 + \ldots + v_{j-1} + v_j (W - w_1 - \ldots - w_{j-1}) / w_j
\end{aligned}$$

which is the same as for the primal. Thus, both solutions are optimal. This shows that the upper bound is valid, based on item (a) of the strong duality result in Theorem 3.2.2. ■

R On the upper bound, we have the following observations:

- If all objective function coefficients are integers, then the optimal solution value is also integer valued. Thus, LB coincides with the upper bound.
- In the upper bound evaluation, suppose that a fraction of item j is included in the knapsack. Then we select item j for branching. In the case where there is no fractional item, the solution at that node is exact and there is no need for branching.

■ **Example 5.2** We consider the following example with 7 items and total weight $W = 21$.

i	1	2	3	4	5	6	7
v_i	17	18	11	10	13	4	1
w_i	8	9	6	6	8	3	2

We start by observing that the following holds:

$$\frac{17}{8} \geq \frac{18}{9} \geq \frac{11}{6} \geq \frac{10}{6} \geq \frac{13}{8} \geq \frac{4}{3} \geq \frac{1}{2}.$$

Node 1 Clearly, items 1 and 2 can fit in the Knapsack, but item 3 cannot. Hence, by rule (K2), we have $j = 3$ and the LP relaxation

$$\begin{aligned}
\max \quad & z = \textstyle\sum_{i=1}^{n} 17x_1 + 18x_2 + 11x_3 + 10x_4 + 13x_5 + 4x_6 x_7 \\
\text{s.t.} \quad & 8x_1 + 9x_2 + 6x_3 + 6x_4 + 8x_5 + 3x_6 + 2x_7 \leq 21, \\
& 0 \leq x_i \leq 1, \; i = 1, \ldots, 7,
\end{aligned} \tag{5.10}$$

has as solution $x_1 = 1$, $x_2 = 1$, $x_3 = 4/6$ and $x_4 = x_5 = x_6 = x_7 = 0$, cf. (5.9). By the bounding rule (K1), the upper and lower bounds are respectively obtained as $UB = 17 + 18 + [(4/6)11] = 42.33$ and $LB = 17 + 18 = 35$. Branching is obviously done on x_3, as $j = 3$. This leads to 2 new nodes: $x_3 = 1$ (or equivalently, $3 \in S$, meaning that item 3 is in the knapsack) and $x_3 = 0$ (or similarly $3 \notin S$).

Node 2 ($3 \notin S$). Here, the constraint $x_3 = 0$ is included to the LP relaxation (5.10) [this maintains the capacity of the knapsack to $W = 21$] and repeating the same process as above (in node 1), we get $j = 4$ and the upper and lower bounds $UB = 17 + 18 + [(4/6)10] = 41.66$ and $LB = 17 + 18 = 35$, respectively.

Node 3 ($3 \in S$). In an attempt to use the best-first search, cf. (K3), we first look at what happens at node 3. With the condition $x_3 = 1$ added to problem (5.10), the new knapsack capacity is $W = 15$. The resulting problem leads to $j = 2$ and $UB = 17 + [(7/9)18] + 11 = 42.00$ and $LB = 17 + 11 = 28$. Based on the best-first search, we first branch from node 3. This leads to nodes 4 and 5.

Node 4 ($3 \in S$, $2 \notin S$). Here, the process starts by adding the constraints $x_3 = 1$ and $x_2 = 0$ to problem (5.10). This keeps the capacity to $W = 15$ and we have $j = 5$. $UB = 17 + 10 + [(1/8)13] + 11 = 39.62$ and $LB = 17 + 10 + 11 = 38$.

Node 5 ($3 \in S, 2 \in S$). Similarly to node 4, we add the constraints $x_3 = 1$ and $x_2 = 1$ to problem (5.10). But here $W = 6$ and we have $j = 1$. $UB = [(6/8)17] + 18 + 11 = 41.75$ and $LB = 18 + 11 = 29$. By the best-first search, we continue branching from this node.

Node 6 ($3 \in S, 2 \in S, 1 \notin S$). By the best-first search, choosing between nodes 4 and 5, we first branch from the latter. Hence, adding $x_3 = 1$, $x_2 = 1$ and $x_1 = 0$ to problem (5.10), we have $W = 6$. This leads to an integer optimal solution with $UB = 10 + 18 + 11 = 39.00$ and $LB = 10 + 18 + 11 = 39$. Hence, the node is fathomed.

Node 7 ($3 \in S, 2 \in S, 1 \in S$). Adding the constraints $x_3 = 1$, $x_2 = 1$ and $x_1 = 1$ to problem (5.10) leads to an infeasible problem (knapsack overfull!). Hence, this node is fathomed. As the current best lower bound is 39, obtained at node 6, there is no need branching further from node 4, as this can only lead to smaller upper bounds, which can eventually be less than 39 [This justifies why the fractional parts have not been included in the search three below]. The search can now continue from node 2; cf. nodes 8 and 9.

Node 8 ($3 \notin S, 4 \notin S$). Adding $x_3 = 0$ and $x_4 = 0$ to problem (5.10), we have $W = 21$ and $j = 5$. Hence, $UB = 17 + 18 + [(4/8)13] = 41.5$ and $LB = 17 + 18 = 35$.

Node 9 ($3 \notin S, 4 \in S$). Adding $x_3 = 0$ and $x_4 = 1$ to problem (5.10), we have $W = 15$ and $j = 2$. Hence, $UB = 17 + [(7/9)18] + 10 = 41.00$ and $LB = 17 + 10 = 27$. By the best-first search, we branch next from node 8, cf. nodes 10 and 11.

Node 10 ($3 \notin S, 4 \notin S, 5 \notin S$). Adding the corresponding constraints to problem (5.10), we have $W = 21$ and $j = 7$. Hence, $UB = 17 + 18 + 4 + [(1/2)1] = 39.5$ and $LB = 17 + 18 + 4 = 39$. For the same reason as at node 4, this node is fathomed (cf. discussion above under node 7).

Node 11 ($3 \notin S, 4 \notin S, 5 \in S$). Adding the corresponding constraints to problem (5.10), we have $W = 13$ and $j = 2$. Hence, $UB = 17 + [(5/9)18] + 13 = 40.00$ and $LB = 17 + 13 = 30$. One way to proceed here would be by first branching from node 9, corresponding to the best-first search (this is the step taken here); otherwise, one can first proceed with nodes 14 and 15 (depth-first search) and the same result will be obtained.

Node 12 $(3 \notin S, 4 \in S, 2 \notin S)$. Adding the corresponding constraints to problem (5.10), we have $W = 15$ and $j = 5$. Hence, $UB = 17 + [(7/8)13] + 10 = 38.37$ and $LB = 17 + 10 = 27$. The upper bound obtained is smaller than the current best lower bound, 39. Hence this node is fathomed.

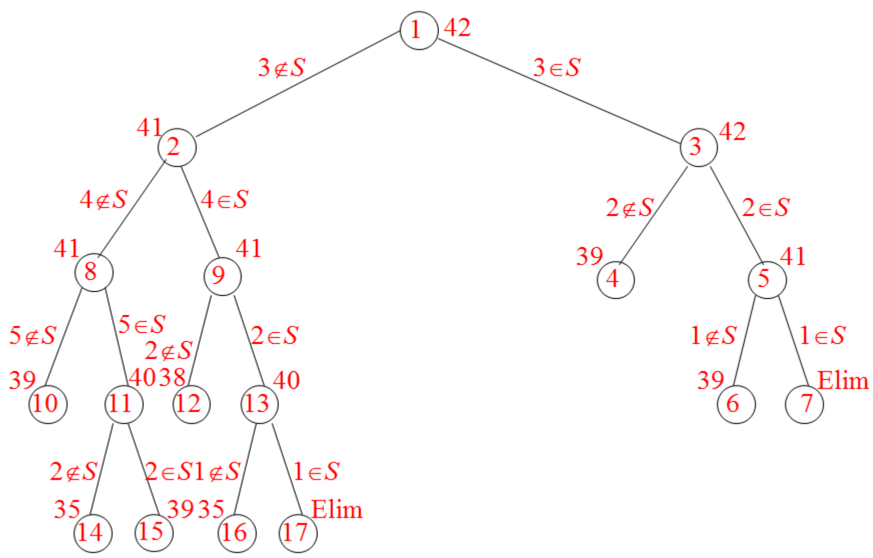
Node 13 $(3 \notin S, 4 \in S, 2 \in S)$. Adding the corresponding constraints to problem (5.10), we have $W = 6$ and $j = 1$. Hence, $UB = [(6/8)17] + 18 + 10 = 40.75$ and $LB = 18 + 10 = 28$. By the best-first search (considered between nodes 11 and 13, we should now branch from node 13). But we first proceed with node 11, as this will make no difference, considering their upper bounds, see discussion under node 7 above.

Node 14 $(3 \notin S, 4 \notin S, 5 \in S, 2 \notin S)$. Adding the corresponding constraints to problem (5.10), we have $W = 13$ and we have an integer optimal solution leading to $UB = 17 + 4 + 1 + 13 = 35.00$ and $LB = 17 + 4 + 1 + 13 = 35$. The upper bound obtained is smaller than the current best lower bound, 39. Hence this node is fathomed.

Node 15 $(3 \notin S, 4 \notin S, 5 \in S, 2 \in S)$. Adding the corresponding constraints to problem (5.10) leads to $W = 4$ and $j = 1$. Hence, $UB = [(4/8)17] + 18 + 13 = 39.5$ and $LB = 18 + 13 = 31$. Node fathomed for the same reason as node 10.

Node 16 $(3 \notin S, 4 \in S, 2 \in S, 1 \notin S)$. Adding the corresponding constraints to problem (5.10) leads to $W = 6$ and $j = 5$. Hence, $UB = [(6/8)13] + 18 + 10 = 37.75$ and $LB = 18 + 10 = 28$. The upper bound obtained is smaller than the current best lower bound, 39. Hence this node is fathomed.

Node 17 $(3 \notin S, 4 \in S, 2 \in S, 1 \in S)$. Adding the corresponding constraints to problem (5.10) leads to an infeasible problem (knapsack overfull!) and node fathomed. The search is now complete.



The best lower bound defines an optimal solution, which in this case is the lower bound found at node 6. Thus, items 2, 3 and 4 should be included in the knapsack, giving a total benefit of 39. ■

R Compared with branch and bound method for pure integer programming problems, discussed in the previous subsection, we have the following particularities for the knapsack problem:
- There is at most one fractional component in the solution of a subproblem.
- Solving the linear relation problem is straightforward, using (5.9).

5.3.3 A branch and bound method for general 0-1 problems

The method that we discuss here is known as *Balas' additive algorithm* as it was introduced by Egon Balas in 1965. The method is designed to solve general zero-one programming problems. To proceed, the objective function should be written as a minimization with *non-negative coefficients*. For illustration, let us consider the following example:

$$\begin{aligned} \min \quad & z = 3x_1 - 5x_2 + 6x_3 + 9x_4 + 10x_5 + 10x_6 \\ & -2x_1 - 6x_2 - 3x_3 + 4x_4 + x_5 - 2x_6 \geq -4, \\ & -5x_1 + 3x_2 + x_3 + 3x_4 - 2x_5 + x_6 \geq 1, \\ & 5x_1 + x_2 + 4x_3 - 2x_4 + 2x_5 - x_6 \geq 4, \\ & x_i \in \{0,1\}, \; i = 1,\dots,6. \end{aligned}$$

Given that the coefficient associated to variable x_2 is negative in the objective function, we make a variable change with $y_2 = 1 - x_2$ and $y_i = x_i$ for $i = 1,3,\dots,6$. This leads to the new problem

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 - 5 \\ & -2y_1 + 6y_2 - 3y_3 + 4y_4 + y_5 - 2y_6 \geq 2, \\ & -5y_1 - 3y_2 + y_3 + 3y_4 - 2y_5 + y_6 \geq -2, \\ & 5y_1 - y_2 + 4y_3 - 2y_4 + 2y_5 - y_6 \geq 3, \\ & y_i \in \{0,1\}, \; i = 1,\dots,6. \end{aligned} \tag{5.11}$$

R Three important observations can be made here:
- Why not just set $y_2 = -x_2$? Doing so will lead to the constraint $y_2 \in \{0,-1\}$. But as we want to solve a 0-1 problem $y_2 = 1 - x_2$ allow us to have $y_2 \in \{0,1\}$.
- In the sequel, the constant -5 that appears in the new objective as a consequence of the variable change will not influence the optimal solution of the problem. Hence, it will be ignored in the solution process and taken into account only when calculating the optimal value of the problem.
- If the problem given is a maximization one, min transformation should be considered as discussed in Chapter 1 and then proceed as above for resulting negative coefficients.

The branch and bound algorithm based on Balas' approach has the following key rules:

(B1) *Bounding rules*: As we are dealing with minimization problems, the *lower bounds* are key here and are obtained by *setting unfixed variables to zero*.

(B2) *Branching rule*: branching is made on a variable which set to 1, *causes the greatest reduction in total infeasibility* (see example below on how to calculate infeasibility).

(B3) *Selecting the node to explore next*: dept first and best first strategies can be used.

(B4) *Fathoming a node*: The criteria from the previous cases, see (B4) or (K4), remain valid here. But a key element in Balas' method is that infeasibility or an *integer-valued* solution can be obtained by fixing additional variables and *making logical deductions from the constraints*.

To illustrate the variable fixing process, consider the constraint

$$2v_1 - 8v_2 + 5v_3 + 8v_4 - 4v_5 + 3v_6 \geq 12, \tag{5.12}$$

where $v_1, \ldots, v_6$ are 0-1 variables. Clearly, the term with the most negative coefficient is $-8v_2$. If we set the variable corresponding to this to 1, i.e., $v_2 = 1$, then (5.12) becomes

$$2v_1 + 5v_3 + 8v_4 - 4v_5 + 3v_6 \geq 20. \tag{5.13}$$

The maximum value of the left-hand-side of (5.13) is 18. This clearly means that setting $v_2 = 1$ is not possible as it leads to an infeasible condition.
If now, we instead fix $v_2 = 0$, then the term with most positive coefficient being $8v_4$, we set $v_4 = 0$. This leads to

$$2v_1 + 5v_3 - 4v_5 + 3v_6 \geq 12.$$

The maximum value of the left-hand-side of this condition is 10 - meaning that this condition is infeasible. Therefore, we fix $v_4 = 1$ and we have

$$2v_1 + 5v_3 - 4v_5 + 3v_6 \geq 4.$$

No further variable fixing is possible. In the process of Balas' algorithm we discuss how to proceed with the branching step when it is not possible fix a variable and *make logical deductions*.

■ **Example 5.3** We apply the algorithm on the transformed problem in (5.11), while ignoring the constant term -5, as mentioned in the remark above:

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 \\ & -2y_1 + 6y_2 - 3y_3 + 4y_4 + y_5 - 2y_6 \geq 2, \\ & -5y_1 - 3y_2 + y_3 + 3y_4 - 2y_5 + y_6 \geq -2, \\ & 5y_1 - y_2 + 4y_3 - 2y_4 + 2y_5 - y_6 \geq 3. \end{aligned}$$

Node 1: At node 1, there is no obvious variable fixing that can be done to stop with the search of the optimal solution. Hence, the optimal value is set to $z = 0$ (with $y_1, \ldots, y_6 = 0$). To decide on which component of y to branch, we proceed with the *calculation of total infeasibility*: For y_1, we set this variable to 1 and all the other components to 0, i.e., $y_1 = 1$, $y_2 = \ldots = y_6 = 0$. This leads to $0 \geq 4$, $0 \geq 3$ and $5 \geq 3$ for the first, second and third constraint, respectively. Clearly, we have infeasibility only for the first and second constraints. Summing $0 \geq 4$ and $0 \geq 3$ gives the inequality $0 \geq 7$. The number 7 here corresponds to the value of the *total infeasibility* for the first variable, cf. table below:

Variable set to 1	y_1	y_2	y_3	y_4	y_5	y_6
Infeasibility	7	5	5	5	$\boxed{2}$	8

The total infeasibility for the other variables is calculated in a similar way. As the smallest infeasibility is due to y_5. Hence, branching is done on this variable.

Node 2: At this node, we have $y_5 = 1$, which leads to problem

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 \\ & -2y_1 + 6y_2 - 3y_3 + 4y_4 - 2y_6 \geq 1, \\ & -5y_1 - 3y_2 + y_3 + 3y_4 + y_6 \geq 0, \\ & 5y_1 - y_2 + 4y_3 - 2y_4 - y_6 \geq 1. \end{aligned}$$

No obvious variable fixing observed, we proceed as in node 1 with the calculation of the total infeasibility of each variable:

Variable set to 1	y_1	y_2	y_3	y_4	y_6
Infeasibility	8	5	4	$\boxed{3}$	5

Branching is therefore done on variable 4. As the lower bounds are generally weak, we use only the depth first search strategy and directly create nodes 4 and subsequently 5, before backtracking.

Node 3: With $y_5 = 1$ and $y_4 = 1$, the new problem we have is

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 \\ & -2y_1 + 6y_2 - 3y_3 - 2y_6 \geq -3, \\ & -5y_1 - 3y_2 + y_3 + y_6 \geq -3, \\ & 5y_1 - y_2 + 4y_3 - y_6 \geq 3, \end{aligned}$$

and with no variable fixing, total infeasibilities are obtained as

Variable set to 1	y_1	y_2	y_3	y_6
Infeasibility	2	4	$\boxed{0}$	4

Having the total infeasibility as 0 for variable y_3 indicates that the points $y = (0,0,1,1,1,0)$ is feasible for the problem. Hence, generating a node 4, which is feasible.

Node 4: Here, the point $y = (0,0,1,1,1,0)$ is feasible with $LB = UB = 25$. Hence, this node is fathomed.

Node 5: Here, $y_5 = 1$, $y_4 = 1$ and $y_3 = 0$. This leads to infeasibility at the third constraint. Hence, the node is fathomed.

Node 6: Backtracking to node 6, where $y_5 = 1$ and $y_4 = 0$, we can also check easily that the second constraint is infeasible. Hence node 6 is also fathomed.

Node 7: Further backtracking to node 7, where $y_5 = 0$, no variable fixing seems obvious. Hence, we consider the resulting problem

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 \\ & -2y_1 + 6y_2 - 3y_3 + 4y_4 - 2y_6 \geq 2, \\ & -5y_1 - 3y_2 + y_3 + 3y_4 + y_6 \geq -2, \\ & 5y_1 - y_2 + 4y_3 - 2y_4 - y_6 \geq 3. \end{aligned}$$

Calculating the total infeasibility for each of the variables gives

Variable set to 1	y_1	y_2	y_3	y_4	y_6
Infeasibility	7	$\boxed{5}$	5	5	8

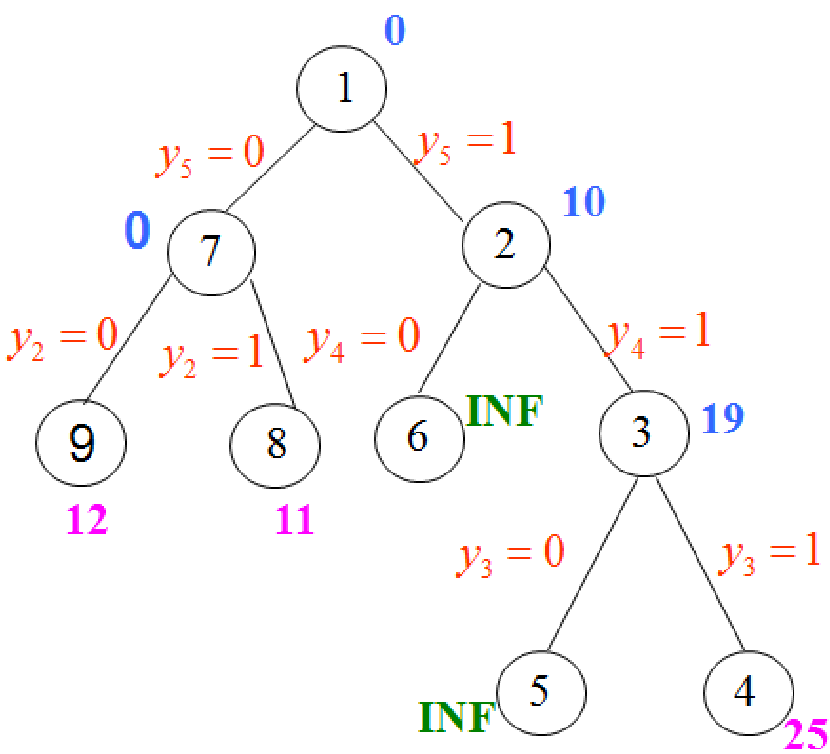
Branching is therefore performed on variable y_2 and leads to nodes 8 and 9.

Node 8: With $y_5 = 0$ and $y_2 = 1$, we have the new problem

$$\begin{aligned} \min \quad & z = 3y_1 + 5y_2 + 6y_3 + 9y_4 + 10y_5 + 10y_6 \\ & -2y_1 - 3y_3 + 4y_4 - 2y_6 \geq 2, \\ & -5y_1 + y_3 + 3y_4 + y_6 \geq -2, \\ & 5y_1 + 4y_3 - 2y_4 - y_6 \geq 3. \end{aligned}$$

Proceeding with logical deductions, cf. (B4), starting by observing that constraint 2 can only hold for $y_1 = 0$, it subsequently follows that the third constraint can only be satisfied if $y_3 = 1$, $y_4 = 0$ and $y_6 = 0$. This leads to a feasible point for the problem, with $LB = 11$.

Node 9: With $y_5 = 0$ and $y_2 = 0$, we can easily check that the second constraint of the resulting problem can only be satisfied if $y_1 = 0$. Subsequently, it follows from the third constraint that we must have $y_3 = 1$, $y_4 = 0$ and $y_6 = 0$. This leads to a feasible point with $LB = 12$. This completes the search and the following tree is obtained:



Clearly, the optimal solution is obtained at node 8: $y_2 = y_3 = 1$, $y_1 = y_4 = y_5 = y_6 = 0$. This implies that for the original problem, $x_1 = 1$, $x_1 = x_2 = x_4 = x_5 = x_6 = 0$ with $z' = 6$. ∎

R To conclude this section, we would like to note that the main advantage of Balas' algorithm is that the computations at each node of the search tree can be performed very quickly. However, the variable fixing tests are not very effective towards the top of the search tree. Also, the lower bounds are weak, and do not effectively restrict the search. Some improvements on the method have been proposed by other authors. But this is out of the scope of this lecture. Further references could be provided to students interested in this.

5.4 Exercises

1. Use the Branch and Bound algorithm to solve the following integer programming problems:

   (a)

$$\begin{array}{ll} \text{Maximize} & z = x_1 + 2x_2 \\ \text{subject to} & x_1 + x_2 \leq 5, \\ & x_1 - x_2 \leq 0, \\ & 2x_1 + 6x_2 \leq 21, \\ & x_1, x_2 \geq 0 \ \text{ integers.} \end{array}$$

   (b)

$$\begin{array}{ll} \text{Maximize} & z = 5x_1 + 2x_2 \\ \text{subject to} & 3x_1 + x_2 \leq 12, \\ & x_1 + x_2 \leq 5, \\ & x_1, x_2 \geq 0 \ \text{ integers.} \end{array}$$

2. Use the Branch and Bound algorithm to solve the following (0-1) Knapsack problems:

   (a)

$$\begin{array}{ll} \text{Maximize} & z = 5x_1 + 8x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 4x_1 + 7x_2 + 5x_3 + 3x_4 \leq \mathbf{12}, \\ & x_i \in \{0,1\}, \ i = 1, \dots, 4. \end{array}$$

   (b)

$$\begin{array}{ll} \text{Maximize} & 7x_1 + 5x_2 + 8x_3 + 3x_4 \\ \text{subject to} & 4x_1 + 3x_2 + 5x_3 + 2x_4 \leq 6, \\ & x_i \in \{0,1\}, \ i = 1, \dots, \ 4. \end{array}$$

3. Use Balas' additive algorithm to solve the following zero-one programming problems:

   (a)

$$\begin{array}{ll} \text{Minimize} & z = 7y_1 + 5y_2 + 2y_3 + 6y_4 + 4y_5 + 19y_6 \\ \text{subject to} & 4y_1 + 6y_2 + 3y_3 + 7y_5 + 3y_6 \geq 13, \\ & 7y_1 - 5y_2 + 6y_3 - 3y_4 + 4y_5 + 2y_6 \geq 7, \\ & 3y_1 + 2y_2 - 2y_3 + 4y_4 - 3y_5 + 7y_6 \geq 4, \\ & y_i \in \{0,1\}, \ i = 1, \dots, 6. \end{array}$$

   (b)

$$\begin{array}{ll} \text{Minimize} & z = 8y_1 + 2y_2 + 3y_3 + 7y_4 + 2y_5 + 6y_6 \\ \text{subject to} & 3y_1 + 6y_2 - y_3 + 5y_4 + 3y_5 + 7y_6 \geq 16, \\ & 5y_1 + 3y_2 - 4y_3 + 8y_4 + 6y_5 - 5y_6 \geq 4, \\ & 6y_1 - 3y_2 + 5y_3 + 5y_4 - 2y_5 + 3y_6 \geq 6, \\ & y_i \in \{0,1\}, \ i = 1, \dots, 6. \end{array}$$

4. Coach Night is trying to choose the starting lineup for the basketball team. The team consists of seven players who have been rated (on scale of 1=poor to 3=excellent) according to their ball-handling, shooting, rebounding, and defense abilities. The positions that each player is allowed to play and the player's abilities are listed in the table below.
   The five-player starting line-player starting lineup must satisfy the following restrictions:
   (a) At least 4 members must be able to play guard, at least 2 members must be able to play forward, and least 1 member must be able to play center.
   (b) The average ball-handling, shooting, and rebounding level of the starting lineup must be at least 2.
   (c) If player 3 starts, then player 6 cannot start.
   (d) If player 1 starts, then player 4 and 5 must both start.
   (e) Either player 2 or player 3 must start.
   Given these constraints, Coach Night wants to maximize the total defensive ability of the starting team. Formulate an integer programming problem that will help him choose his starting team.

Player	Position	Ball-handling	Shooting	Rebounding	Defence
1	G	3	3	1	3
2	C	2	1	3	2
3	G-F	2	3	2	2
4	F-C	1	3	3	1
5	G-F	3	3	3	3
6	F-C	3	1	2	3
7	G-F	3	2	2	1

Table 5.1: for Problem 2

5. A bank is open from 9:00 to 17:00. During each hour of the day, the number of clerks required is shown in the following table:

Time period	**Number of clerks**
$9:00-10:00$	4
$10:00-11:00$	3
$11:00-12:00$	4
$12:00-13:00$	6
$13:00-14:00$	5
$14:00-15:00$	6
$15:00-16:00$	8
$16:00-17:00$	8

   The bank can hire full-time and part-time clerks. Full-time clerks work from 9:00 to 17:00 except for a one-hour lunch break, which is from 12:00-13:00 or from 13:00-14:00 (the bank decides the time at which each clerk takes their lunch break). The clerks are paid £8 per hour (and receive payment for their lunch break). Part-time clerks work for three consecutive hours and the bank specifies the start time for each of them. Part-time clerks are paid £6 per hour. No more than five part time clerks can be hired.
   Use linear/integer programming to model the problem of finding a minimum cost hiring policy for clerks.