LINEAR OPTIMIZATION IN APPLICATIONS
## CONTENTS

Preface vii

**Chapter 1: Introduction** 1
1.1 Formulation of a linear programming problem 1
1.2 Solving a linear programming problem 2
   1.2.1 Graphical method 3
   1.2.2 Simplex method 4
   1.2.3 Revised simplex method 6

**Chapter 2: Primal and Dual Models** 9
2.1 Shadow price / opportunity cost 9
2.2 The dual model 9
2.3 Comparing primal and dual 11
2.4 Algebraic way to find shadow prices 13
2.5 A worked example 15

**Chapter 3: Formulating Linear Optimization Problems** 19
3.1 Transportation problem 19
3.2 Transportation problem with distributors 22
3.3 Trans-shipment problem 24
3.4 Earth moving optimization 27
3.5 Production schedule optimization 30
3.6 Aggregate blending problem 32
3.7 Liquid blending problem 34
3.8 Wastewater treatment optimization 37
3.9 Critical path of a precedence network 40
3.10 Time-cost optimization of a project network 42

**Chapter 4: Transportation Problem and Algorithm** 51
4.1 The general form of a transportation problem 51
4.2 The algorithm 53
4.3 A further example 59
4.4 More applications of transportation algorithm 65
   4.4.1 Trans-shipment problem 65
   4.4.2 Earth moving problem 68
   4.4.3 Product schedule problem 69
4.5 An interesting example using transportation algorithm 71
This book is not written to discuss the mathematics of linear programming. It is designed to illustrate, with practical examples, the applications of linear optimization techniques. The simplex method and the revised simplex method, therefore, are included as appendices only in the book.

The book is written in simple and easy to understand language. It has put together a very useful and comprehensive set of worked examples, based on real life problems, using linear programming and its extensions including integer programming and goal programming. The examples are used to explain how linear optimization can be applied in engineering and business/management problems. The steps shown in each worked example are clear and easy to read.

The book is likely to be used by teachers, taught course students and research students of both engineering and business/management disciplines. It is, however, not suitable for students of pure mathematics because the contents of the book emphasize applications rather than theories.
1.1 Formulation of a Linear Programming Problem

Linear programming is a powerful mathematical tool for the optimization of an objective under a number of constraints in any given situation. Its application can be in maximizing profits or minimizing costs while making the best use of the limited resources available. Because it is a mathematical tool, it is best explained using a practical example.

Example 1.1
A pipe manufacturing company produces two types of pipes, type I and type II. The storage space, raw material requirement and production rate are given as below:

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type I</th>
<th>Type II</th>
<th>Company Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage space</td>
<td>5 m²/pipe</td>
<td>3 m²/pipe</td>
<td>750 m²</td>
</tr>
<tr>
<td>Raw materials</td>
<td>6 kg/pipe</td>
<td>4 kg/pipe</td>
<td>800 kg/day</td>
</tr>
<tr>
<td>Production rate</td>
<td>30 pipes/hour</td>
<td>20 pipes/hour</td>
<td>8 hours/day</td>
</tr>
</tbody>
</table>

The profit for selling one type I pipe is $10 and that for type II is $8. The pipes produced each day are taken by trucks to sales outlets in the early morning of the next day before a new day’s manufacturing work starts. Our objective is to formulate for the company a linear programming model which can determine how many pipes of each type should be manufactured each day so that the total profit can be maximized.

Solution 1.1

Let \( Z = \) total profit

\( x_1 = \) number of type I pipes produced each day

\( x_2 = \) number of type II pipes produced each day

Since our objective is to maximize profit, we write an objective function, equation (0), which will calculate the total profit:

\[
\text{Maximize } Z = 10x_1 + 8x_2 \quad (0)
\]
x₁ and x₂ in equation (0) are called decision variables.

There are three constraints which govern the number of type I and type II pipes produced. These constraints are: (1) the availability of storage space, (2) the raw materials available, and (3) the working hours of labourers. Constraints (1), (2) and (3) are written as below:

Storage space: 5x₁ + 3x₂ ≤ 750  
(1)

Raw material: 6x₁ + 4x₂ ≤ 800  
(2)

Working hours: \( \frac{x₁}{30} + \frac{x₂}{20} ≤ 8 \)  
(3)

When constraint (3) is multiplied by 60, the unit of hours will be changed to the unit of minutes (i.e. 8 hours to 480 minutes). Constraint (3) can be written as:

\( 2x₁ + 3x₂ ≤ 480 \)  
(3)

Lastly, there are two more constraints which are not numbered. They are x₁ ≥ 0 and x₂ ≥ 0, simply because the quantities x₁ and x₂ cannot be negative.

We can now summarize the problem as a linear programming model as follows:

Maximize: \( Z = 10x₁ + 8x₂ \)  
(0)

subject to

5x₁ + 3x₂ ≤ 750  
(1)

6x₁ + 4x₂ ≤ 800  
(2)

2x₁ + 3x₂ ≤ 480  
(3)

x₁ ≥ 0

x₂ ≥ 0

1.2 Solving a Linear Programming Problem

There are two methods in solving linear programming models, namely, the graphical method and the simplex method. The graphical method can only solve linear programming problems with two decision variables, while the simplex method can solve problems with any number of decision variables. Since this book will only concentrate on the applications of linear programming,
the mathematical details for solving the models will not be thoroughly treated. In this section, the graphical method and the simplex method will only be briefly described.

1.2.1 Graphical Method

Let us look at the linear programming model for Example 1.1:

\[
\text{Max } Z = 10x_1 + 8x_2 \quad (0) \\
\text{subject to} \\
5x_1 + 3x_2 \leq 750 \quad (1) \\
6x_1 + 4x_2 \leq 800 \quad (2) \\
2x_1 + 3x_2 \leq 480 \quad (3) \\
x_1 \geq 0 \\
x_2 \geq 0
\]

The area bounded by (1) : \(5x_1 + 3x_2 = 750\), (2) : \(6x_1 + 4x_2 = 800\), (3) : \(2x_1 + 3x_2 = 480\), (4) : \(x_1 = 0\) and (5) : \(x_2 = 0\) is called the feasible space, which is the shaded area shown in Fig. 1.1. Any point that lies within this feasible space will satisfy all the constraints and is called a feasible solution.

![Graphical Method](image)

Note: the \(x_1\) and \(x_2\) axes are not drawn on the same scale.

The optimal solution is a feasible solution which, on top of satisfying all constraints, also optimizes the objective function, that is, maximizes profit in this case. By using the slope of the objective function, \(-10/8\) in our case, a line can be drawn with such a slope which touches a point within the feasible space.
and is as far away as possible from the point of origin 0. This point is represented by A in Fig. 1.1 and is the optimal solution. From the graph, it can be seen that at optimum,

\[
\begin{align*}
  x_1 &= 48 \quad \text{(type I pipes)} \\
  x_2 &= 128 \quad \text{(type II pipes)} \\
  \max Z &= 1504 \quad \text{(profit in $), calculated from 10(48) + 8(128)}
\end{align*}
\]

From Fig. 1.1, one can also see whether or not the resources (i.e. storage space, raw materials, working time) are fully utilized.

Consider the storage space constraint (1). The optimal point A does not lie on line (1) and therefore does not satisfy the equation \(5x_1 + 3x_2 = 750\). If we substitute \(x_1 = 48\) and \(x_2 = 128\) into this equation, we obtain:

\[5(48) + 3(128) = 624 < 750\]

Therefore, at optimum, only 624 m\(^2\) of storage space are used and 126 m\(^2\) (i.e. 750 - 624) are not used.

By similar reasoning, we can see that the other two resources (raw materials and working time) are fully utilized.

If constraint (1) of the above problem is changed to \(5x_1 + 3x_2 \leq 624\), that is, the available storage space is 624 m\(^2\) instead of 750 m\(^2\), then line (1) will also touch the feasible space at point A. In this case, lines (1), (2) and (3) are concurrent at point A and all the three resources are fully utilized when the maximum profit is attained. There is a technical term called "optimal degenerate solution" used for such a situation.

### 1.2.2 Simplex Method

When there are three or more decision variables in a linear programming model, the graphical method is no more suitable for solving the model. Instead of the graphical method, the **simplex method** will be used.
As mentioned earlier, the main theme of this book is applications of linear programming, not mathematical theory behind linear programming. Therefore, no detail description of the mathematics of linear programming will be presented here. There are many well developed computer programs available in the market for solving linear programming models using the simplex method. One of them is QSB\(^+\) (Quantitative Systems for Business Plus) written by Y.L. Chang and R.S. Sullivan and can be obtained in any large bookshop world-wide. The author will use the QSB\(^+\) software to solve all the problems contained in the later chapters of this book.

Examples of the techniques employed in the simplex method will be illustrated in Appendix A at the end of this book. Some salient points of the method are summarized below.

First of all we introduce slack variables \(S_1, S_2\) and \(S_3\) \((S_1, S_2, S_3 \geq 0)\) for Example 1.1 to change the constraints from inequalities to equalities such that the model becomes:

\begin{align*}
Z - 10x_1 - 8x_2 &= 0 \quad \text{(0a)} \\
\text{subject to} \\
5x_1 + 3x_2 + S_1 &= 750 \quad \text{(1a)} \\
6x_1 + 4x_2 + S_2 &= 800 \quad \text{(2a)} \\
2x_1 + 3x_2 + S_3 &= 480 \quad \text{(3a)}
\end{align*}

The initial tableau of the simplex method is shown in Table 1.1. It is in fact a rewrite of equations (0a), (1a), (2a) and (3a) in a tableau format.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>(Z)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0a) (Z)</td>
<td>1</td>
<td>-10</td>
<td>-8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1a) (S_1)</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>(2a) (S_2)</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>(3a) (S_3)</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>480</td>
</tr>
</tbody>
</table>

*Table 1.1 Initial Simplex Tableau for Example 1.1*
After two iterations (see Appendix A), the final tableau will be obtained and is shown in Table 1.2 below.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Z</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0c) $Z$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.4</td>
<td>0.8</td>
<td>1504</td>
</tr>
<tr>
<td>(1c) $S_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-0.9</td>
<td>-0.2</td>
<td>126</td>
</tr>
<tr>
<td>(2c) $x_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>-0.4</td>
<td>48</td>
</tr>
<tr>
<td>(3c) $x_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-0.2</td>
<td>0.6</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 1.2 Final Simplex Tableau for Example 1.1

To obtain a solution from a simplex tableau, the basic variables are equal to the values in the RHS column. The non-basic variables (i.e. the decision variables or slack variables which are not in the basic variable column) are assigned the value zero. Therefore, from the final tableau, we can see that the optimal solution is:

$$Z = 1504$$
$$x_1 = 48$$
$$x_2 = 128$$
$$S_1 = 126$$
$$S_2 = 0$$
$$S_3 = 0$$

\{ \text{non-basic variables are equal to 0.} \}

It can be seen that this result is the same as that found by the graphical method. $S_1$ here is 126, which means that the slack variable for storage space is 126 and therefore 126 m$^2$ of storage space is not utilized. $S_1$ and $S_2$ are slack variables for the other two resources and are equal to 0. This means that the raw materials and the working time are fully utilized.

### 1.2.3 Revised Simplex Method

The revised simplex method is also called the modified simplex method. In this method, the objective function is usually written in the last row instead of the first. Examples of the technique are illustrated in Appendix B. QSB$^+$ uses the revised simplex method in solving linear programming models. The initial tableau for Example 1.1 is shown in Table 1.3.
After two iterations (see Appendix B), the final tableau will be obtained. It is shown in Table 1.4.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Cj</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(S_1)</th>
<th>(S_2)</th>
<th>(S_3)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1)</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>(x_1)</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-0.9</td>
<td>0.2</td>
<td>126</td>
</tr>
<tr>
<td>(x_2)</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>-0.4</td>
<td>48</td>
</tr>
<tr>
<td>(S_2)</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>800</td>
</tr>
<tr>
<td>(S_3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>750</td>
</tr>
<tr>
<td>(Z_j)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(C_j - Z_j)</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1504</td>
</tr>
</tbody>
</table>

Table 1.4 Final Simplex Tableau for Example 1.1 (Revised Simplex Method)
2.1 Shadow Price / Opportunity Cost

The shadow price (or called opportunity cost) of a resource is defined as the economic value (increase in profit) of an extra unit of resource at the optimal point. For example, the raw material available in Example 1.1 of Chapter 1 is 800 kg; the shadow price of it means the increase in profit (or the increase in $Z$, the objective function) if the raw material is increased by one unit, to 801 kg.

Now, let

\[ y_1 = \text{shadow price of storage space} ($/m^2) \]
\[ y_2 = \text{shadow price of raw material} ($/kg) \]
\[ y_3 = \text{shadow price of working time} ($/minute) \]

This means that one additional $m^2$ of storage space available (i.e. 751 $m^2$ is available instead of 750 $m^2$) will increase $Z$ by $y_1$ dollars; one additional kg of raw materials available will increase $Z$ by $y_2$ dollars; and one additional minute of working time available will increase $Z$ by $y_3$ dollars.

Based on the definition of shadow price, we can formulate another linear programming model for Example 1.1. This new model is called the dual model.

2.2 The Dual Model

Since the production of a type I pipe requires 5 $m^2$ of storage space, 6 kg of raw material and 2 minutes of working time, the shadow price of producing one extra type I pipe will be $5y_1 + 6y_2 + 2y_3$. This means that the increase in profit (i.e. $Z$) due to producing an additional type I pipe is $5y_1 + 6y_2 + 2y_3$, which should be greater than or at least equal to $10$, the profit level of selling one type I pipe, in order to justify the extra production. Hence, we can write the constraint that:

\[ 5y_1 + 6y_2 + 2y_3 \geq 10 \] \hspace{1cm} (1)

A similar argument applies to type II pipe, and we can write another constraint that:

\[ 3y_1 + 4y_2 + 3y_3 \geq 8 \] \hspace{1cm} (2)
It is impossible to have a decrease in profit due to an extra input of any resources. Therefore the shadow prices cannot have negative values. So, we can also write:

\[ y_1 \geq 0 \]
\[ y_2 \geq 0 \]
\[ y_3 \geq 0 \]

We can also interpret the shadow price as the amount of money that the pipe company can afford to pay for one additional unit of resource so that he can just break even on the use of that resource. In other words, the company can afford, to pay, say, \( y_2 \) for one extra kg of raw material. If it pays less than \( y_2 \) from the market to buy the raw material it will make a profit, and vice versa.

The objective this time is to minimize cost. The total price, \( P \), of the total resources employed in producing pipes is equal to \( 750y_1 + 800y_2 + 480y_3 \). In order to minimize \( P \), the objective function is written as:

\[
\text{Minimize } P = 750y_1 + 800y_2 + 480y_3
\]

We can now summarize the dual model as follows:

\[
\text{Min } P = 750y_1 + 800y_2 + 480y_3 \quad \text{(0)}
\]

subject to

\[
5y_1 + 6y_2 + 2y_3 \geq 10 \quad \text{(1)}
\]
\[
3y_1 + 4y_2 + 3y_3 \geq 8 \quad \text{(2)}
\]

\[ y_1 \geq 0 \]
\[ y_2 \geq 0 \]
\[ y_3 \geq 0 \]

The solution of this dual model (see Appendix A or Appendix B) is:

\[
\text{min } P = 1504
\]
\[ y_1 = 0 \]
\[ y_2 = 1.4 \]
\[ y_3 = 0.8 \]

We can observe that the optimal value of \( P \) is equal to the optimal value of \( Z \) found in Chapter 1. The shadow price of storage space, \( y_1 \), is equal to 0. This means that one additional \( \text{m}^2 \) of storage space will result in no increase in profit.
This is reasonable because there has already been unutilized storage space. The shadow price of raw material, $y_2$, is equal to 1.4. This means that one additional kg of raw material will increase the profit level by $1.4. The shadow price of working time, $y_3$, is equal to 0.8. This means that one additional minute of working time will increase the profit level by $0.8. It can also be interpreted that $0.8/\text{minute}$ is the amount which the company can afford to pay for the extra working time. If the company pays less than $0.8/\text{minute}$ for the workers it will make a profit, and vice versa.

### 2.3 Comparing Primal and Dual

The linear programming model given in Chapter 1 is referred to as a **primal model**. Its dual form has been discussed in Section 2.2. These two models are reproduced below for easy reference.

<table>
<thead>
<tr>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max ( Z = 10x_1 + 8x_2 )</td>
<td>Min ( P = 750y_1 + 800y_2 + 480y_3 )</td>
</tr>
<tr>
<td>subject to</td>
<td>subject to</td>
</tr>
<tr>
<td>( 5x_1 + 3x_2 \leq 750 )</td>
<td>( 5y_1 + 6y_2 + 2y_3 \geq 10 )</td>
</tr>
<tr>
<td>( 6x_1 + 4x_2 \leq 800 )</td>
<td>( 3y_1 + 4y_2 + 3y_3 \geq 8 )</td>
</tr>
<tr>
<td>( 2x_1 + 3x_2 \leq 480 )</td>
<td></td>
</tr>
<tr>
<td>( x_1 \geq 0 )</td>
<td>( y_1 \geq 0 )</td>
</tr>
<tr>
<td>( x_2 \geq 0 )</td>
<td>( y_2 \geq 0 )</td>
</tr>
</tbody>
</table>

It can be observed that:

(a) the coefficients of the objective function in the primal model are equal to the RHS constants of the constraints in the dual model,

(b) the RHS constants of the constraints of the primal model are the coefficient of the objective function of the dual model, and

(c) the coefficients of $y_1$, $y_2$ and $y_3$, when read row by row, for the two constraints of the dual model are equal to those of $x_1$ and $x_2$, when read column by column, in the primal model. In other words, the dual is the transpose of the primal if the coefficients of the constraints are imagined as a matrix.
The general form of a primal model is:

Max \( Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \)

subject to

\[ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \]
\[ \vdots \]
\[ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m \]
all \( x_j \geq 0 \)

The general form of the dual model will be:

Min \( P = b_1y_1 + b_2y_2 + \ldots + b_my_m \)

subject to

\[ a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \geq c_1 \]
\[ a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \geq c_2 \]
\[ \vdots \]
\[ a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \geq c_n \]
all \( y_i \geq 0 \)

The two models are related by:

(a) maximum \( Z = \) minimum \( P \), and

(b) \( y_i = \frac{\Delta Z}{\Delta b_i} \)

where \( y_i \) stand for the shadow price of the resource \( i \) and \( b_i \) is the amount of the \( i^{th} \) resource available.

It should be pointed out that it is not necessary to solve the dual model in order to find \( y_i \). In fact, \( y_i \) can be seen from the final simplex tableau of the primal model. Let us examine the final tableau of Example 1.1:
We can see that the coefficient of $S_1$, slack variable for storage space, in the first row (the row of the objective function $Z$) is equal to $y_1$, the shadow price of storage space in $$/m^2$. The coefficient of $S_2$, slack variable for raw material, in the first row is equal to $y_2$, the shadow price of raw material in $$/kg. Again, the coefficient of $S_3$, slack variable for working time, in the first row is equal to $y_3$, the shadow price of working time in $$/minute.

Therefore, it is not necessary to solve the dual model to find the values of the decision variables (ie. $y_1$, $y_2$ and $y_3$). They can be found from the primal model. The same occurs in the revised simplex method. The final tableau of the revised method for Example 1.1 is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$Z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.4</td>
<td>0.8</td>
<td>1504</td>
</tr>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-0.9</td>
<td>-0.2</td>
<td>126</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>-0.4</td>
<td>48</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-0.2</td>
<td>0.6</td>
<td>128</td>
</tr>
</tbody>
</table>

In a similar way, the values of $y_1$, $y_2$ and $y_3$ can be seen from the row of $Z_j$.

### 2.4 Algebraic Way to Find Shadow Prices

There is a simple algebraic way to find the shadow price without employing the simplex method. Let us use the same example, Example 1.1, again to illustrate how this can be done. The linear programming model is reproduced hereunder for easy reference:

$$\text{Max } Z = 10x_1 + 8x_2$$

(0)
subject to
\[5x_1 + 3x_2 \leq 750 \quad \text{--------------------------- (1)}\]
\[6x_1 + 4x_2 \leq 800 \quad \text{--------------------------- (2)}\]
\[2x_1 + 3x_2 \leq 480 \quad \text{--------------------------- (3)}\]

As can be seen from the graphical method, storage space has no effect on the optimal solution (see Section 1.2.1). Line (1) \(5x_1 + 3x_2 = 750\) therefore does not pass through the optimal point. Since raw material and working time both define the optimal point, the solution is therefore the intersection point of line (2) \(6x_1 + 4x_2 = 800\) and line (3) \(2x_1 + 3x_2 = 480\).

Assuming that the raw material available is increased by \(\Delta L\) kg, the optimal solution will then be obtained by solving:
\[6x_1 + 4x_2 = 800 + \Delta L\]
and \[2x_1 + 3x_2 = 480\]

Solving, we obtain:
\[x_1 = 48 + \frac{3}{10} \Delta L\]
and \[x_2 = 128 - \frac{1}{5} \Delta L\]

Substituting \(x_1\) and \(x_2\) into the objective function, we have:
\[Z + \Delta Z = 10(48 + \frac{3}{10} \Delta L) + 8(128 - \frac{1}{5} \Delta L)\]

Simplifying, we get
\[Z + \Delta Z = 10(48) + 8(128) + \frac{7}{5} \Delta L\]

Since \(Z = 10(48) + 8(128)\)
\[\therefore \Delta Z = \frac{7}{5} \Delta L\]

i.e. \[\frac{\Delta Z}{\Delta L} = 1.4\] (i.e. shadow price of raw material in $/kg)
Similarly, if we assume that the working time available is increased by \( \Delta M \) minutes, we can, by the same method, obtain that:

\[
\frac{\Delta Z}{\Delta M} = 0.8 \quad \text{(i.e. shadow price of working time in $/minute)}
\]

### 2.5 A Worked Example

Let us now see a practical example of the application of shadow prices.

**Example 2.1**

A company which manufactures table lamps has developed three models denoted the “Standard”, “Special” and “Deluxe”. The financial returns from the three models are $30, $40 and $50 respectively per unit produced and sold. The resource requirements per unit manufactured and the total capacity of resources available are given below:

<table>
<thead>
<tr>
<th></th>
<th>Machining (hours)</th>
<th>Assembly (hours)</th>
<th>Painting (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Special</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Deluxe</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Available Capacity</td>
<td>20,000</td>
<td>10,000</td>
<td>6,000</td>
</tr>
</tbody>
</table>

(a) Find the number of units of each type of lamp that should be produced such that the total financial return is maximized. (Assume all units produced are also sold.)

(b) At the optimal product mix, which resource is under-utilized?

(c) If the painting-hours resource can be increased to 6,500, what will be the effect on the total financial return?

**Solution 2.1**

(a) The problem can be represented by the following linear programming model:

\[
\text{Max } Z = 30x_1 + 40x_2 + 50x_3
\]
subject to
\[ 3x_1 + 4x_2 + 4x_3 \leq 20,000 \]
\[ 2x_1 + 2x_2 + 3x_3 \leq 10,000 \]
\[ x_1 + 2x_2 + 3x_3 \leq 6,000 \]
\[ x_1, x_2, x_3 \geq 0 \]

where \( x_1 \) = number of Standard lamps produced
\( x_2 \) = number of Special lamps produced
\( x_3 \) = number of Deluxe lamps produced

Introduce slack variables \( S_1, S_2 \) and \( S_3 \) such that:
\[ 3x_1 + 4x_2 + 4x_3 + S_1 = 20,000 \]
\[ 2x_1 + 2x_2 + 3x_3 + S_2 = 10,000 \]
\[ x_1 + 2x_2 + 3x_3 + S_3 = 6,000 \]

Using the simplex method to solve the model, the final tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Z</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>S_1</th>
<th>S_2</th>
<th>S_3</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td></td>
<td>160,000</td>
</tr>
<tr>
<td>S_1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>4,000</td>
</tr>
<tr>
<td>x_1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>4,000</td>
</tr>
<tr>
<td>x_2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>0</td>
<td>-0.5</td>
<td>1</td>
<td>1,000</td>
</tr>
</tbody>
</table>

The final tableau of the revised simplex method is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>C_j</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>S_1</th>
<th>S_2</th>
<th>S_3</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>4,000</td>
</tr>
<tr>
<td>x_1</td>
<td>30</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>4,000</td>
</tr>
<tr>
<td>x_2</td>
<td>40</td>
<td>0</td>
<td>1</td>
<td>1.5</td>
<td>0</td>
<td>-0.5</td>
<td>1</td>
<td>1,000</td>
</tr>
<tr>
<td>Z_i</td>
<td>30</td>
<td>40</td>
<td>60</td>
<td>0</td>
<td>10</td>
<td></td>
<td>10</td>
<td>160,000</td>
</tr>
</tbody>
</table>

From the final tableau of either method, we obtain the following optimal solution:
Primal and Dual Models

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>Non-basic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 = 4,000 )</td>
<td>( x_3 = 0 )</td>
</tr>
<tr>
<td>( x_1 = 4,000 )</td>
<td>( S_2 = 0 )</td>
</tr>
<tr>
<td>( x_2 = 1,000 )</td>
<td>( S_3 = 0 )</td>
</tr>
<tr>
<td>( Z = 160,000 )</td>
<td></td>
</tr>
</tbody>
</table>

Therefore, the company should manufacture 4,000 Standard lamps, 1,000 Special lamps and no Deluxe lamps. The maximum financial return is $160,000.

(b) Machining is under-utilized. Since \( S_1 = 4,000 \), therefore 4,000 machining hours are not used.

(c) The shadow price of painting-hours resource can be read from either tableau and is equal to $10/hour. This means that the financial return will increase by $10 if the painting-hour resource is increased by 1 hour. If the painting hour is increased by 500 (from 6,000 hours to 6,500 hours), then the total increase in financial return is $10 \times 500 = $5,000, and the overall financial return will be $160,000 + $5,000 = $165,000.

It should be noted that the shadow price $10/per hour may not be valid for infinite increase of painting hours. In this particular problem, it is valid until the painting hour is increased to 10,000. This involves post optimality analysis which is outside the scope of this book. The software QSB, however, shows the user the range of validity for each and every resource in its sensitivity analysis function.

It is also worthwhile to note that while the \( Z_j \) values of the final revised simplex tableau represent shadow prices of the corresponding resources, the \( C_j - Z_j \) values represent the reduced costs. The reduced cost of a non-basic decision variable is the reduction in profit of the objective function due to one unit increase of that non-basic decision variable. In Example 2.1, the non-basic decision variable is \( x_3 \), and so \( x_3 \) is equal to 0 when the total profit is maximized to $160,000. If we want to produce one Deluxe lamp in the
product mix (i.e. $x_3 = 1$) under the original available resources, then the total profit will be reduced to $159,990$ (i.e. $160,000 - 10$) because the reduced cost of $x_3$ is $-10$ (the $C_j - Z_j$ value under column $x_3$).

**Exercise**

1. A precast concrete subcontractor makes three types of panels. In the production the quantities of cement, coarse aggregates and fines aggregates required are as follows:

<table>
<thead>
<tr>
<th></th>
<th>Cement (m$^3$/panel)</th>
<th>Course aggregates (m$^3$/panel)</th>
<th>Fines aggregates (m$^3$/panel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel I</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Panel II</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Panel III</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

The subcontractor has the following quantities of cement, coarse aggregates and fines aggregates per week:

- Cement : 300 m$^3$
- Coarse aggregates : 500 m$^3$
- Fines aggregates : 620 m$^3$

The financial return for panel types I, II and III are 20, 18 and 25 respectively. Find the number of panels of each type that should be made so that the total financial return is maximized. Which resource is under-utilized and why? If the subcontractor can obtain some extra coarse aggregates, what is the maximum cost per m$^3$ the subcontractor can afford to pay for it?
In Chapter 1, we have seen how a linear programming model for maximizing profit under limited resources is formulated. In Chapter 2, we have also seen how its dual model is formulated. This chapter contains more examples on the formulation of linear programming problems.

3.1 Transportation Problem

Example 3.1

Goods have to be transported from three warehouses 1, 2 and 3 to two customers 1 and 2. The three warehouses have the following quantities of stock to be transported per week.

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Quantity of stock (tonnes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
</tr>
</tbody>
</table>

The requirements of the two customers per week are:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Quantity required (tonnes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
</tr>
</tbody>
</table>

The costs of transporting one tonne of stock from each warehouse to each customer is given in Table 3.1.

<table>
<thead>
<tr>
<th>Cost per stock from i to j</th>
<th>Customer 1</th>
<th>Customer 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warehouse</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 3.1 Cost matrix of the transportation problem

This means that it costs $9 to transport one tonne of stock from warehouse 1 to customer 1, $10 to customer 2, and so on.
The problem is to determine how many tonnes of stock to transport from each warehouse to each customer per week in order to minimize the overall transportation cost.

Solution 3.1
Let $x_{ij}$ be the decision variables so that $x_{ij}$ is the number of tonnes of stock transported from warehouse $i$ to customer $j$ per week as follows:

$$
\begin{array}{c}
\text{Warehouse} & \text{Customer} \\
1 & 1 & 2 \\
2 & 1 & 2 \\
3 & 1 & 2 \\
\end{array}
$$

It is best to represent the problem in the form of a transportation tableau (Fig. 3.1).

<table>
<thead>
<tr>
<th>Customer</th>
<th>1</th>
<th>2</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warehouse</td>
<td>9 $X_{11}$</td>
<td>10 $X_{12}$</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>6 $X_{21}$</td>
<td>7 $X_{22}$</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>8 $X_{31}$</td>
<td>11 $X_{32}$</td>
<td>30</td>
</tr>
<tr>
<td>Demand</td>
<td>29</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3.1 A transportation tableau

The objective function is to minimize overall transportation cost:

\[
\text{Minimize } P = 9X_{11} + 10X_{12} + 6X_{21} + 7X_{22} + 8X_{31} + 11X_{32}
\]

There are three constraints on the amount of stock transported from each warehouse; the amount must not exceed the supply available:
Formulating Linear Optimization Problems

Warehouse 1: \( x_{11} + x_{12} \leq 12 \) \hfill (1)

Warehouse 2: \( x_{21} + x_{22} \leq 20 \) \hfill (2)

Warehouse 3: \( x_{31} + x_{32} \leq 30 \) \hfill (3)

There are two constraints on the amount of stock demanded by the customers:

Customer 1: \( x_{11} + x_{21} + x_{31} = 29 \) \hfill (4)

Customer 2: \( x_{12} + x_{22} + x_{32} = 33 \) \hfill (5)

So, the linear programming model for this problem is summarized as follows:

\[
\text{Min } P = 9x_{11} + 10x_{12} + 6x_{21} + 7x_{22} + 8x_{31} + 11x_{32}
\]

subject to

\[
\begin{align*}
  x_{11} + x_{12} &\leq 12 \quad \text{(1)} \\
  x_{21} + x_{22} &\leq 20 \quad \text{(2)} \\
  x_{31} + x_{32} &\leq 30 \quad \text{(3)} \\
  x_{11} + x_{21} + x_{31} & = 29 \quad \text{(4)} \\
  x_{12} + x_{22} + x_{32} & = 33 \quad \text{(5)}
\end{align*}
\]

all \( x_{ij} \geq 0 \) \( (i = 1, 2, 3 \text{ and } j = 1, 2) \)

The solution for this model is:

Overall minimum transportation cost = 503

\[
\begin{align*}
  x_{11} &= 0 & x_{12} &= 12 \\
  x_{21} &= 0 & x_{22} &= 20 \\
  x_{31} &= 29 & x_{32} &= 1
\end{align*}
\]

The above example is usually called a **transportation problem**. A transportation problem can be solved by the simplex method of course, or by another algorithm technique which will be discussed in Chapter 4.
3.2 Transportation Problem With Distributors

Example 3.2

This example is related to but a little more complicated than Example 3.1. Assume that the three warehouses (i = 1, 2, 3) are transporting their stocks to the two customers (k = 1, 2) directly and/or through two distributors (j = 1, 2). Distributorships are offered at the discretion of the warehouses only when needed. Distributors 1 and 2 have weekly capacities to store and distribute 40 and 35 tonnes of stock respectively. The system can be represented by Fig. 3.2.

![Fig. 3.2] A transportation problem with distributors

The costs $C_{ik}$ of transporting a tonne of stock from i to k on route $z_{ik}$ (i.e. direct delivery without through the distributors) have already been given in Example 3.1. The cost of transporting a tonne of stock from each warehouse to each distributor (i.e. i to j) and from each distributor to each customer (i.e. j to k) are as follows:
Formulating Linear Optimization Problems

<table>
<thead>
<tr>
<th>Route i - j</th>
<th>Unit cost C_{ij}</th>
<th>Route j - k</th>
<th>Unit cost C_{jk}</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>4</td>
<td>1-1</td>
<td>3</td>
</tr>
<tr>
<td>1-2</td>
<td>6</td>
<td>1-2</td>
<td>5</td>
</tr>
<tr>
<td>2-1</td>
<td>5</td>
<td>2-1</td>
<td>6</td>
</tr>
<tr>
<td>2-2</td>
<td>4</td>
<td>2-2</td>
<td>4</td>
</tr>
<tr>
<td>3-1</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How should the stocks be distributed to the customers such that the overall distribution cost is minimized?

**Solution 3.2**

**Objective function:**

Minimize \( P = \sum C_{ij} x_{ij} + \sum C_{jk} y_{jk} + \sum C_{ik} z_{ik} \)

\[
= 4x_{11} + 6x_{12} + 5x_{21} + 4x_{22} + 5x_{31} + 3x_{32} + 3y_{11} + 5y_{12} + 6y_{21} + 4y_{22} + 9z_{11} + 10z_{12} + 6z_{21} + 7z_{22} + 8z_{31} + 11z_{32}
\]

**Constraints:**

The warehouse supply constraints:

\[
\sum_{j=1}^{2} x_{ij} + \sum_{k=1}^{2} z_{ik} \leq \text{Supply of warehouse} \quad \text{for } i = 1, 2, 3
\]

i.e.

\[
x_{11} + x_{12} + z_{11} + z_{12} \leq 12
\]

(1)

\[
x_{21} + x_{22} + z_{21} + z_{22} \leq 20
\]

(2)

\[
x_{31} + x_{32} + z_{31} + z_{32} \leq 30
\]

(3)

The distributor capacity constraints:

\[
\sum_{i=1}^{3} x_{ij} \leq \text{Capacity of distributor} \quad \text{for } j = 1, 2
\]

i.e.

\[
x_{11} + x_{21} + x_{31} \leq 40
\]

(4)

\[
x_{12} + x_{22} + x_{32} \leq 35
\]

(5)
The distributor input-output constraints:

\[ \sum_{i=1}^{3} x_{ij} \geq \sum_{k=1}^{2} y_{jk} \quad \text{for } j = 1, 2 \]

i.e. \( x_{11} + x_{21} + x_{31} \geq y_{11} + y_{12} \)
\( x_{12} + x_{22} + x_{32} \geq y_{21} + y_{22} \) (6)

(7)

The customer demand constraints:

\[ \sum_{i=1}^{3} z_{ik} + \sum_{j=1}^{2} y_{jk} = \text{Demand of customer} \quad \text{for } k = 1, 2 \]

i.e. \( z_{11} + z_{21} + z_{31} + y_{11} + y_{21} = 29 \)
\( z_{12} + z_{22} + z_{32} + y_{12} + y_{22} = 33 \) (8)

(9)

The final set of constraints are:

all \( x_{ij} \geq 0, \ y_{jk} \geq 0, \ z_{ik} \geq 0 \)

The solution for this model is:

Minimum overall distribution cost = 417

\[
\begin{align*}
x_{11} & = 12 & y_{11} & = 12 & z_{11} & = 0 \\
x_{12} & = 0 & y_{12} & = 0 & z_{12} & = 0 \\
x_{21} & = 0 & y_{21} & = 0 & z_{21} & = 17 \\
x_{22} & = 0 & y_{22} & = 30 & z_{22} & = 3 \\
x_{31} & = 0 & & & z_{31} & = 0 \\
x_{32} & = 30 & & & z_{32} & = 0
\end{align*}
\]

3.3 Trans-shipment Problem

In Example 3.1, we have seen what a transportation problem is. A transportation problem involves a set of source locations and another set of destinations. We have also seen an extension of the transportation problem in Example 3.2 with the introduction of a set of distributors in between the sources and the destinations. Now, we shall see another extension of the transportation problem, called the trans-shipment problem, in which stock can flow into and
out of intermediate points in a network, depending upon where they are needed most. Let us now see a trans-shipment example.

**Example 3.3**

![Diagram of a trans-shipment problem](image)

In Fig. 3.3, the warehouses 1 and 4 have positive numbers, which means that 18 and 12 tonnes of stock respectively are to be transported out from these two nodes in the network. Demand stations 3 and 6 have negative numbers, which means that 14 and 16 tonnes of stock respectively are required by these two nodes. Nodes 2 and 5 are intermediate nodes which neither supply nor demand any items of stock. These two nodes are called **trans-shipment nodes**. Sometimes, nodes 3 and 4 are also called trans-shipment points because stock can be both in and out in these nodes. The costs $C_{ij}$ are unit costs of transportation, similar to those of Examples 3.1 and 3.2. Note that delivery is possible in both directions between nodes 2 and 5. The unit costs $C_{25}$ and $C_{52}$ may be the same or may be different (in this example, they are different). Nodes 4 and 5 also allow travelling in both directions; $C_{45}$ and $C_{54}$ may, again, be the same or different (in this example, they are the same). The requirement is to relocate the stocks at minimum cost.

**Solution 3.3**

Let $x_{ij}$ be the tonnes of stock transported from i to j ($i \neq j$). The objective function is:
Minimize \( P = C_{12} x_{12} + C_{23} x_{23} + C_{34} x_{34} + C_{25} x_{25} + C_{52} x_{52} \)

\[ + C_{33} x_{33} + C_{45} x_{45} + C_{54} x_{54} + C_{46} x_{46} + C_{56} x_{56} \]

\[ = 10x_{12} + 20x_{23} + 15x_{34} + 22x_{25} + 18x_{52} \]

\[ + 12x_{33} + 11x_{43} + 11x_{54} + 13x_{46} + 16x_{56} \]

subject to

At node 1: \( x_{12} \leq 18 \)

At node 2: \(-x_{12} - x_{52} + x_{23} + x_{25} = 0\)

At node 3: \(-x_{23} - x_{33} + x_{34} = -14\)

At node 4: \(-x_{34} - x_{44} + x_{45} + x_{46} \leq 12\)

At node 5: \(-x_{25} - x_{45} + x_{52} + x_{53} + x_{54} + x_{56} = 0\)

At node 6: \(-x_{46} - x_{56} = -16\)

all \( x_{ij} \geq 0 \)

The optimal solution of the above linear programming model can never have \( x_{25} \) and \( x_{52} \) both positive, nor \( x_{45} \) and \( x_{54} \) both positive.

The solution of this model is given below:

Total minimum trans-shipment cost = 768

\( x_{12} = 18 \)

\( x_{23} = 14 \)

\( x_{25} = 4 \)

\( x_{46} = 12 \)

\( x_{56} = 4 \)

other \( x_{ij} = 0 \)

There are two modifications to a trans-shipment problem. The first modification is the addition of capacity constraints to the nodes. For example, if node 2 can only process the trans-shipment of a maximum of \( R_2 \) tonnes of stock, node 3 can only process \( R_3 \) tonnes, and node 5 can only process \( R_5 \) tonnes, then we have three additional constraints:
At node 2, \( x_{12} + x_{52} \leq R_2 \) \hspace{1cm} (7)

At node 3, \( x_{23} + x_{53} \leq R_3 \) \hspace{1cm} (8)

At node 5, \( x_{25} + x_{45} \leq R_5 \) \hspace{1cm} (9)

The second modification is the addition of capacity constraints to the routes. For example, if route 2-3 has only a flow capacity of \( L_{23} \), route 5-2 a flow capacity of \( L_{52} \), route 4-6 a flow capacity of \( L_{46} \), and route 5-4 a flow capacity of \( L_{54} \), then we have four additional constraints:

For route 2-3: \( x_{23} \leq L_{23} \) \hspace{1cm} (10)

For route 5-2: \( x_{52} \leq L_{52} \) \hspace{1cm} (11)

For route 4-6: \( x_{46} \leq L_{46} \) \hspace{1cm} (12)

For route 5-4: \( x_{54} \leq L_{54} \) \hspace{1cm} (13)

### 3.4 Earth Moving Optimization

**Example 3.4**

A highway contract requires a contractor to alter the terrain of a section of road work. The work involves cut and fill of earth so that the original profile will be much flatter. The original profile and the finished profile are shown in solid line and dotted line respectively in Fig. 3.4.

<table>
<thead>
<tr>
<th>Section No.</th>
<th>Cut ( (m^3 \times 10^3) )</th>
<th>Fill ( (m^3 \times 10^3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>11</td>
</tr>
</tbody>
</table>

**Fig. 3.4** Original and finished profiles of cut and fill
The roadway is divided into nine sections. For example, $10 \times 10^3$ m$^3$ of fill is required in section 1; $28 \times 10^3$ m$^3$ of cut in section 2; and so on. The cost for cutting including loading is $8$ per m$^3$ and that for filling including compaction is $12$ per m$^3$. The unit cost of transporting earth by trucks from one section to another is $2$ per m$^3$ per section.

Our task is to determine how much earth should be moved from where to where so as to optimize the earth moving operations.

**Solution 3.4**

Let $x_{ij}$ be the quantity of earth in thousand m$^3$ to be moved from section $i$ to section $j$ ($i \neq j$).

Therefore, there are 18 (i.e. $3 \times 6$) decision variables as follows:

<table>
<thead>
<tr>
<th>Fill section</th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut Section 2</td>
<td>$x_{21}$</td>
<td>$x_{24}$</td>
<td>$x_{25}$</td>
<td>$x_{26}$</td>
<td>$x_{28}$</td>
<td>$x_{29}$</td>
</tr>
<tr>
<td>Cut Section 3</td>
<td>$x_{31}$</td>
<td>$x_{34}$</td>
<td>$x_{35}$</td>
<td>$x_{36}$</td>
<td>$x_{38}$</td>
<td>$x_{39}$</td>
</tr>
<tr>
<td>Cut Section 7</td>
<td>$x_{71}$</td>
<td>$x_{74}$</td>
<td>$x_{75}$</td>
<td>$x_{76}$</td>
<td>$x_{78}$</td>
<td>$x_{79}$</td>
</tr>
</tbody>
</table>

$C_{ij}$ can be calculated as follows:

$C_{21} = \text{cost per m}^3 \text{ for moving earth from section 2 to section 1}$

$= 8 + 2 \times 1 + 12 = 22$

$\uparrow \quad \uparrow \quad \uparrow$

Cut  Transport  Fill

$C_{24} = \text{cost per m}^3 \text{ for moving earth from section 2 to section 4}$

$= 8 + 2 \times 2 + 12 = 24$

$\uparrow \quad \uparrow \quad \uparrow$

Cut  Transport  Fill

Other $C_{ij}$ can be calculated in the same way.
Now, we can write the objective function:

Minimize $P = \sum C_{ij} x_{ij}$

$$\begin{align*}
&= 22x_{21} + 24x_{24} + 26x_{25} + 28x_{26} + 32x_{28} + 34x_{29} \\
&\quad + 24x_{31} + 22x_{34} + 24x_{35} + 26x_{36} + 30x_{38} + 32x_{39} \\
&\quad + 32x_{71} + 26x_{74} + 24x_{75} + 22x_{76} + 22x_{78} + 24x_{79}
\end{align*}$$

Constraints:

The constraints for cuts:

$$\begin{align*}
x_{21} + x_{24} + x_{25} + x_{26} + x_{28} + x_{29} &\leq 28 \quad \text{(1)} \\
x_{31} + x_{34} + x_{35} + x_{36} + x_{38} + x_{39} &\leq 32 \quad \text{(2)} \\
x_{71} + x_{74} + x_{75} + x_{76} + x_{78} + x_{79} &\leq 13 \quad \text{(3)}
\end{align*}$$

The constraints for fills:

$$\begin{align*}
x_{21} + x_{31} + x_{71} & = 10 \quad \text{(4)} \\
x_{24} + x_{34} + x_{74} & = 14 \quad \text{(5)} \\
x_{25} + x_{35} + x_{75} & = 16 \quad \text{(6)} \\
x_{26} + x_{36} + x_{76} & = 10 \quad \text{(7)} \\
x_{28} + x_{38} + x_{78} & = 12 \quad \text{(8)} \\
x_{29} + x_{39} + x_{79} & = 11 \quad \text{(9)}
\end{align*}$$

all $x_{ij} \geq 0$ for $i = 2, 3, 7$ and $j = 1, 4, 5, 6, 8, 9$

The solution for this model is given below:

Minimum overall earth moving cost = $1,816 \times 10^3$

$$\begin{align*}
x_{21} & = 10 \\
x_{24} & = 8 \\
x_{28} & = 10 \\
x_{34} & = 6 \\
x_{35} & = 16 \\
x_{36} & = 10 \\
x_{78} & = 2 \\
x_{79} & = 11 \\
\text{other } x_{ij} & = 0
\end{align*}$$
We can observe that the model formulated for this earth moving problem is similar to the one for Example 3.1, the transportation problem. In fact, this example is indeed a transportation problem which can be solved by the simplex method, or an algorithm which will be discussed in Chapter 4.

3.5 Production Schedule Optimization

Example 3.5

A cement manufacturer provides cement for ready-mix concrete companies. In the next four months its volume of sales, production costs and available worker-hours are estimated as follows:

<table>
<thead>
<tr>
<th>Month</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cement required (m³ x 10³)</td>
<td>220</td>
<td>300</td>
<td>250</td>
<td>320</td>
</tr>
<tr>
<td>Cost of regular time production ($ per thousand m³)</td>
<td>2,500</td>
<td>2,600</td>
<td>2,700</td>
<td>2,800</td>
</tr>
<tr>
<td>Cost of overtime production ($ per thousand m³)</td>
<td>3,000</td>
<td>3,100</td>
<td>3,200</td>
<td>3,300</td>
</tr>
<tr>
<td>Regular time worker-hours</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td>Overtime worker-hours</td>
<td>800</td>
<td>800</td>
<td>800</td>
<td>800</td>
</tr>
</tbody>
</table>

There is no cement in stock initially. It takes 10 hours of production time to produce one thousand m³ of cement. It costs $150 to store one thousand m³ of cement from one month to the next.

Now, the cement company wishes to know the optimal production schedule.

Solution 3.5

Let \( x_1, x_2, x_3 \) and \( x_4 \) be the number of thousand m³ of cement produced in months 1, 2, 3 and 4 respectively on regular time.

Let \( y_1, y_2, y_3 \) and \( y_4 \) be the number of thousand m³ of cement produced in months 1, 2, 3 and 4 respectively on overtime.
Let \( z_1, z_2 \) and \( z_3 \) be the number of thousand \( m^3 \) of cement in stock at the end of months 1, 2 and 3 respectively which have to be carried to the next.

After defining the decision variables, the problem can be presented diagramatically as shown in Fig. 3.5.

Fig. 3.5 Cement company production model

The objective function is:

\[
\text{Minimize } P = 2500x_1 + 2600x_2 + 2700x_3 + 2800x_4 \\
+ 3000y_1 + 3100y_1 + 3200y_3 + 3300y_4 \\
+ 150z_1 + 150z_2 + 150z_3
\]

subject to

Monthly demand constraints:

Month 1: \( x_1 + y_1 - z_1 = 220 \) \hspace{2cm} (1)
Month 2: \( x_2 + y_2 + z_1 - z_2 = 300 \) \hspace{2cm} (2)
Month 3: \( x_3 + y_3 + z_2 - z_3 = 250 \) \hspace{2cm} (3)
Month 4: \( x_4 + y_4 + z_3 = 320 \) \hspace{2cm} (4)

Monthly regular worker-hours constraints:

\( 10x_1 \leq 2000 \) \hspace{2cm} (5)
\( 10x_2 \leq 2000 \) \hspace{2cm} (6)
\( 10x_3 \leq 2000 \) \hspace{2cm} (7)
\( 10x_4 \leq 2000 \) \hspace{2cm} (8)
Monthly overtime worker-hour constraints:

\[
\begin{align*}
10y_1 & \leq 800 & \quad (9) \\
10y_2 & \leq 800 & \quad (10) \\
10y_3 & \leq 800 & \quad (11) \\
10y_4 & \leq 800 & \quad (12) \\
\end{align*}
\]

All \( x_i \geq 0, \ y_i \geq 0 \) and \( z_i \geq 0 \)

The solution for this model is:

Total production cost = 2,458,500

\[
\begin{align*}
x_1 &= 200 & y_1 &= 50 & z_1 &= 30 \\
x_2 &= 200 & y_2 &= 80 & z_2 &= 10 \\
x_3 &= 200 & y_3 &= 80 & z_3 &= 40 \\
x_4 &= 200 & y_4 &= 80 \\
\end{align*}
\]

3.6 Aggregate Blending Problem

**Example 3.6**

The contract specifications of a building project require that the course aggregate grading for concrete mixing must be within the following limits:

<table>
<thead>
<tr>
<th>Sieve size</th>
<th>Percentage passing by weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>63.0 mm</td>
<td>100%</td>
</tr>
<tr>
<td>37.5 mm</td>
<td>95% to 100%</td>
</tr>
<tr>
<td>20.0 mm</td>
<td>35% to 70%</td>
</tr>
<tr>
<td>10.0 mm</td>
<td>10% to 40%</td>
</tr>
<tr>
<td>5.0 mm</td>
<td>0% to 5%</td>
</tr>
</tbody>
</table>

The contractor has four quarry sites to supply aggregates. These aggregates, however, do not individually satisfy the above specifications requirement. The aggregate grading from each of quarry sites is shown below:
The contractor needs to prepare 150 tonnes of aggregates for concrete mixing in the next two days. How should he obtain the aggregates from the quarry sites so that after blending them the aggregates will satisfy the specifications?

**Solution 3.6**

Let \( x_1, x_2, x_3 \) and \( x_4 \) be the number of tonnes of aggregates supplied from quarry sites 1, 2, 3 and 4 respectively.

The objective function:

Minimize \( P = 12x_1 + 14x_2 + 10x_3 + 11x_4 \)

Constraints:

The constraints for 37.5 mm sieve:

\[
0.95 \leq \frac{0.90x_1 + 0.95x_2 + 1.00x_3 + 1.00x_4}{x_1 + x_2 + x_3 + x_4} \leq 1.00
\]

This can be written as:

\[
0.05x_1 - 0.05x_3 - 0.05x_4 \leq 0 \quad \text{(1)}
\]

and

\[
0.10x_1 + 0.05x_2 \geq 0 \quad \text{(2)}
\]

The constraints for 20.0 mm sieve:

\[
0.35 \leq \frac{0.50x_1 + 0.65x_2 + 0.80x_3 + 1.00x_4}{x_1 + x_2 + x_3 + x_4} \leq 0.70
\]

This can be written as:

\[
0.15x_1 + 0.30x_2 + 0.45x_3 + 0.65x_4 \geq 0 \quad \text{(3)}
\]
and \(-0.20x_1 - 0.05x_2 + 0.10x_3 + 0.30x_4 \leq 0\) \hspace{1cm} (4)

The constraints for 10.0 mm sieve:
\[
0.10 \leq \frac{0.00x_1 + 0.25x_2 + 0.40x_3 + 0.90x_4}{x_1 + x_2 + x_3 + x_4} \leq 0.40
\]
This can be written as:
\[
-0.10x_1 + 0.15x_2 + 0.30x_3 + 0.80x_4 \geq 0 \hspace{1cm} (5)
\]
and
\[
-0.40x_1 - 0.15x_2 + 0.50x_4 \leq 0 \hspace{1cm} (6)
\]

The constraints for 5.0 mm sieve:
\[
0 \leq \frac{0.00x_1 + 0.00x_2 + 0.03x_3 + 0.30x_4}{x_1 + x_2 + x_3 + x_4} \leq 0.05
\]
This can be written as:
\[
0.03x_3 + 0.30x_4 \geq 0 \hspace{1cm} (7)
\]
and
\[
-0.05x_1 - 0.05x_2 + 0.02x_3 + 0.25x_4 \leq 0 \hspace{1cm} (8)
\]

The constraints for demand:
\[
x_1 + x_2 + x_3 + x_4 = 150 \hspace{1cm} \text{(9)}
\]
Also, \(x_1, x_2, x_3, x_4 \geq 0\)

The solution for this model is given below:
Total minimum costs = 1,690.16
\[
x_1 = 65.26 \text{ tonnes}
\]
\[
x_2 = 10.04 \text{ tonnes}
\]
\[
x_3 = 55.22 \text{ tonnes}
\]
\[
x_4 = 19.48 \text{ tonnes}
\]

3.7 Liquid Blending Problem

In Example 3.6, we have seen a problem involving aggregate blending. For blending liquids, the linear programming model can be formulated, in most cases, in the very same way. However, we will now see an example of
optimizing liquid blending problem in which the requirements are specified in a slightly different way.

**Example 3.7**

A paint manufacturing company produces three enamels types I, II and III by mixing acrylic polymers in three different formulations, types A, B and C.

Type A polymer contains 55% solids and 45% solvent, type B polymer 45% solids and 55% solvent, type C polymer 35% solids and 65% solvent. Polymers A, B and C cost $6 per litre, $7.5 per litre and $9 per litre respectively.

Type I enamel must contain at least 30% solids and at least 50% solvent, type II enamel at least 40% solids but no more than 60% solvent, type III enamel not more than 60% solids and not more than 70% solvent.

The paint manufacturing company has got a rush order for 800 litres of type I enamel, 950 litres of type II enamel and 650 litres of type III enamel. How many litres of each polymer should the company purchase for producing the required enamels?

**Solution 3.7**

The problem can be presented diagramatically as shown in Fig. 3.7.
The objective function:

\[
\text{Minimize } P = 6(x_{A1} + x_{A2} + x_{A3}) + 7.5(x_{B1} + x_{B2} + x_{B3}) + 9(x_{C1} + x_{C2} + x_{C3})
\]

Constraints:

The constraint for enamel I solids:

\[
\frac{0.55x_{A1} + 0.45x_{B1} + 0.35x_{C1}}{x_{A1} + x_{B1} + x_{C1}} \geq 0.30
\]

i.e. \(0.25x_{A1} + 0.15x_{B1} + 0.05x_{C1} \geq 0\) \hspace{1cm} (1)

The constraint for enamel I solvent:

\[
\frac{0.45x_{A1} + 0.55x_{B1} + 0.65x_{C1}}{x_{A1} + x_{B1} + x_{C1}} \geq 0.50
\]

i.e. \(-0.05x_{A1} + 0.05x_{B1} + 0.15x_{C1} \geq 0\) \hspace{1cm} (2)

The constraint for enamel II solids:

\[
\frac{0.55x_{A2} + 0.45x_{B2} + 0.35x_{C2}}{x_{A2} + x_{B2} + x_{C2}} \geq 0.40
\]

i.e. \(0.15x_{A2} + 0.05x_{B2} - 0.05x_{C2} \geq 0\) \hspace{1cm} (3)

The constraint for enamel II solvent:

\[
\frac{0.45x_{A2} + 0.55x_{B2} + 0.65x_{C2}}{x_{A2} + x_{B2} + x_{C2}} \leq 0.60
\]

i.e. \(-0.15x_{A2} - 0.05x_{B2} + 0.05x_{C2} \leq 0\) \hspace{1cm} (4)

The constraint for enamel III solids:

\[
\frac{0.55x_{A3} + 0.45x_{B3} + 0.35x_{C3}}{x_{A3} + x_{B3} + x_{C3}} \leq 0.60
\]

i.e. \(-0.05x_{A3} - 0.15x_{B3} - 0.25x_{C3} \leq 0\) \hspace{1cm} (5)
The constraint for enamel III solvent:

\[
\frac{0.45x_{A3} + 0.55x_{B3} + 0.65x_{C3}}{x_{A3} + x_{B3} + x_{C3}} \leq 0.70
\]

i.e. \(-0.25x_{A3} - 0.15x_{B3} - 0.05x_{C3} \leq 0\) \hspace{1cm} (6)

The constraints for quantities of enamels:

\[
x_{A1} + x_{B1} + x_{C1} = 800 \hspace{1cm} (7)
\]

\[
x_{A2} + x_{B2} + x_{C2} = 950 \hspace{1cm} (8)
\]

\[
x_{A3} + x_{B3} + x_{C3} = 650 \hspace{1cm} (9)
\]

and all \(x_{ij} \geq 0\) where \(i = A, B, C\)

and \(j = 1, 2, 3\)

The solution for this model is:

Total minimum purchasing costs = 15,000

\[
x_{A1} = 600 \quad x_{B1} = 0 \quad x_{C1} = 200
\]

\[
x_{A2} = 950 \quad x_{B2} = 0 \quad x_{C2} = 0
\]

\[
x_{A3} = 650 \quad x_{B3} = 0 \quad x_{C3} = 0
\]

3.8 Wastewater Treatment Optimization

Example 3.8

An industrial company has three factories located along three streams as shown in Fig. 3.8.

![Fig. 3.8 An industrial wastewater treatment problem](image)

Factory A generates a daily average of 1000 m³ industrial wastewater of 800 mg/litre of BOD₅ (biological oxygen demand at 5 days, a measure of degree of
pollution), factory B 1500 m$^3$ of wastewater of 600 mg/litre of BOD$_5$, and factory C 1800 m$^3$ of wastewater of 1000 mg/litre of BOD$_5$. Before the wastewater is discharged into the streams, the industrial company has to build in-house treatment plants at each of the factories so as to remove the pollutants to a level which is acceptable at the downstream waters.

The costs for treating 1 kg of BOD$_5$ at factory A is $100, factory B $110 and factory C $120. The rates of flow per day in the streams are $Q_1$, $Q_2$ and $Q_3$ which are 0.2 million m$^3$, 0.22 million m$^3$ and 0.25 million m$^3$ respectively. The flows in the streams are assumed to have no pollutants unless they are contaminated by the discharges from the factories.

The river requirement is that no water in any part of the streams will exceed an average standard of 2 mg/litre of BOD$_5$. It can also be assumed that 20% of the pollutants discharged at factory A will be removed by natural processes (e.g. sunshine, oxidation etc.) before they reach factory B, and that 15% of the pollutants be removed by natural processes between factory B and factory C.

The company wishes to know the optimal treatment capacities of the in-house treatment plants at the three factories.

**Solution 3.8**

Let $x_1$ = kg of BOD$_5$ to be removed daily at factory A

$x_2$ = kg of BOD$_5$ to be removed daily at factory B

$x_3$ = kg of BOD$_5$ to be removed daily at factory C

The objective function is:

Minimize $P = 100x_1 + 110x_2 + 120x_3$

subject to the following constraints:

The constraint for factory A:

Daily BOD$_5$ generated by factory A $= 800$ mg/L $\times 1000$ m$^3$

$= 800$ kg
Formulating Linear Optimization Problems

\[ \therefore \text{Daily BOD}_5 \text{ released into stream} = 800 - x_1 \]
\[ \text{River standard} = 2 \text{ mg/L and } Q_2 = 0.22 \text{ million m}^3 \]
\[ \therefore \text{Allowable pollutants in stream 2 after passing factory A} = 440 \text{ kg} \]

Hence, \[ 800 - x_1 \leq 440 \]  \hspace{1cm} (1)
and \[ x_1 \leq 800 \]  \hspace{1cm} (2)

The constraint for factory B:

\[ \text{Daily BOD}_5 \text{ generated by factory B} = 600 \text{ mg/L x 1500 m}^3 \]
\[ = 900 \text{ kg} \]
\[ \therefore \text{Daily BOD}_5 \text{ released into stream} = 900 - x_2 \]
\[ \text{River standard} = 2 \text{ mg/L and } Q_1 + Q_2 = 0.42 \text{ million m}^3 \]
\[ \therefore \text{Allowable pollutants in stream 1 after passing factory B} = 840 \text{ kg} \]
20% of pollutants are removed by natural processes

Hence, \[ 0.8(800 - x_1) + (900 - x_2) \leq 840 \] \hspace{1cm} (3)
and \[ x_2 \leq 900 \]  \hspace{1cm} (4)

The constraint for factory C:

\[ \text{Daily BOD}_5 \text{ generated by factory C} = 1000 \text{ mg/L x 1800 m}^3 \]
\[ = 1800 \text{ kg} \]
\[ \therefore \text{Daily BOD}_5 \text{ released into stream} = 1800 - x_3 \]
\[ \text{River standard} = 2 \text{ mg/L and } Q_1 + Q_2 + Q_3 = 0.67 \text{ million m}^3 \]
\[ \therefore \text{Allowable pollutants in stream 1 after passing factory C} = 1340 \text{ kg} \]
15% of pollutants are removed by natural processes

Hence,
\[ 0.85[0.8(800 - x_1) + (900 - x_2)] + (1800 - x_3) \leq 1340 \] \hspace{1cm} (5)
and \[ x_3 \leq 1800 \] \hspace{1cm} (6)

Other constraints:
\[ x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0 \]

The solution for this model is:

\[ \text{Total minimum treatment costs} = 222,200 \]
3.9 Critical Path of a Precedence Network

A precedence network is usually referred to as an activity-on-node network, that is, an activity is denoted by a node in a network diagram which shows the sequence of the activities in a logical manner. There are of course methods other than linear programming for tracing a critical path in a network. The following example, however, shows how linear programming can do such a job.

Example 3.9

![Diagram of an activity-on-node network]

Fig. 3.9 An activity-on-node network

The durations of the activities of a project are shown in the network diagram in Fig. 3.9 and are as follows:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Activity duration (weeks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>9</td>
</tr>
<tr>
<td>70</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
x_1 = 360 \\
x_2 = 412 \\
x_3 = 1174
\]
Find the shortest project duration (critical path) by the use of linear programming.

**Solution 3.9**

Let \( x_{10} = \) start time of activity 10
\( x_{20} = \) start time of activity 20
\( \ldots \)
\( x_{70} = \) start time of activity 70
\( x_{\text{END}} = \) project duration

The objective is to:

Minimize \( P = x_{\text{END}} \)

subject to

\[ x_{30} - x_{10} \geq 5 \quad \text{(1)} \]
\[ x_{40} - x_{20} \geq 4 \quad \text{(2)} \]
\[ x_{50} - x_{30} \geq 4 \quad \text{(3)} \]
\[ x_{60} - x_{30} \geq 3 \quad \text{(4)} \]
\[ x_{60} - x_{40} \geq 6 \quad \text{(5)} \]
\[ x_{70} - x_{50} \geq 2 \quad \text{(6)} \]
\[ x_{\text{END}} - x_{60} \geq 9 \quad \text{(7)} \]
\[ x_{\text{END}} - x_{70} \geq 3 \quad \text{(8)} \]

All \( x_i \geq 0 \) for \( i = 10, 20, 30, 40, 50, 60, 70, \text{END} \)

The solution for this model is:

Shortest project duration = minimum \( x_{\text{END}} = 19 \) weeks

\( x_{10} = 2 \)
\( x_{20} = 0 \)
\( x_{30} = 7 \)
\( x_{40} = 4 \)
\( x_{50} = 14 \)
\( x_{60} = 10 \)
\( x_{70} = 16 \)
\( x_{\text{END}} = 19 \)
Readers can always check whether the project duration is 19 weeks or not using the critical path method.

3.10 Time-Cost Optimization of a Project Network

In Example 3.9, we have seen how the critical path or the shortest duration in a project network can be found using the linear programming method. Now, we shall see how the time-cost optimization can be done if the normal duration & cost and the crash duration & cost of each activity in the network are given.

Example 3.10

The normal duration and cost and those of the crash for the activities of the network in the previous example (Fig. 3.9) are as follows:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Normal</th>
<th>Crash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Duration (R)</td>
<td>Cost (U)</td>
</tr>
<tr>
<td>10</td>
<td>5 weeks</td>
<td>$100</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>150</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>150</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>60</td>
<td>9</td>
<td>550</td>
</tr>
<tr>
<td>70</td>
<td>3</td>
<td>100</td>
</tr>
</tbody>
</table>

The indirect cost (overheads and so on) of the project is $60 per week. How should the activities be compressed so that an optimal project duration can be achieved with the minimum total direct and indirect costs of the project?

Solution 3.10

Before formulating the linear programming model, some preliminary theories of the normal point and the crash point of an activity must be discussed.
When additional resources (hence additional cost) is used to carry out an activity, it will usually be completed faster than its normal duration. The crash point is reached when the duration of the activity cannot be further shortened even if extra resources are input. We denote the normal duration and crash duration by R and Q respectively, and normal cost and crash cost by U and V respectively.

The cost slope of an activity between normal and crash = \( C = \frac{V - U}{R - Q} \)

Now, we can calculate C of each activity:

<table>
<thead>
<tr>
<th>Activity</th>
<th>( C = \frac{V - U}{R - Q} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>60</td>
<td>110</td>
</tr>
<tr>
<td>70</td>
<td>50</td>
</tr>
</tbody>
</table>

An activity can be compressed to any duration between R and Q. We assume that D is the activity duration after compressing (\( R \geq D \geq Q \)) such that the time and cost of the overall project is optimized. M is the corresponding cost associated with D.
By similar triangles, we can easily derive that

$$D = R - \frac{1}{C} (M - U)$$

Therefore, for each activity:

$$D_{10} = 5 - \frac{1}{20} (M_{10} - 100)$$

$$D_{20} = 4 - \frac{1}{20} (M_{20} - 150)$$

$$D_{30} = 3$$

$$D_{40} = 6 - \frac{1}{50} (M_{40} - 300)$$

$$D_{50} = 2$$

$$D_{60} = 9 - \frac{1}{110} (M_{60} - 550)$$

$$D_{70} = 3 - \frac{1}{50} (M_{70} - 100)$$

Using the formulation discussed in Example 3.9, the activity duration constraints are:

$$x_{30} - x_{10} \geq D_{10}$$  \hspace{1cm} (Remember that $x_i$ is the start time of activity $i$)

$$x_{40} - x_{20} \geq D_{20}$$

$$x_{50} - x_{40} \geq D_{40}$$

$$x_{60} - x_{50} \geq D_{50}$$

$$x_{60} - x_{40} \geq D_{40}$$

$$x_{70} - x_{60} \geq D_{60}$$

$$x_{70} - x_{50} \geq D_{50}$$

$$x_{\text{END}} - x_{60} \geq D_{60}$$

$$x_{\text{END}} - x_{70} \geq D_{70}$$

These can be written as:

$$x_{30} - x_{10} \geq 5 - \frac{1}{20} (M_{10} - 100) \hspace{1cm} (1)$$

$$x_{40} - x_{20} \geq 4 - \frac{1}{20} (M_{20} - 150) \hspace{1cm} (2)$$
Formulating Linear Optimization Problems

\[ x_{50} - x_{20} \geq 4 - \frac{1}{20} (M_{20} - 150) \]  
(3)

\[ x_{60} - x_{30} \geq 3 \]  
(4)

\[ x_{60} - x_{40} \geq 6 - \frac{1}{50} (M_{40} - 300) \]  
(5)

\[ x_{70} - x_{50} \geq 2 \]  
(6)

\[ x_{\text{END}} - x_{60} \geq 9 - \frac{1}{110} (M_{60} - 550) \]  
(7)

\[ x_{\text{END}} - x_{70} \geq 3 - \frac{1}{50} (M_{70} - 100) \]  
(8)

The second set of constraints is the minimum cost (or normal cost) constraints:
\[ M_{10} \geq 100 \]  
(9)

\[ M_{20} \geq 150 \]  
(10)

\[ M_{30} \geq 150 \]  
(11)

\[ M_{40} \geq 300 \]  
(12)

\[ M_{50} \geq 200 \]  
(13)

\[ M_{60} \geq 550 \]  
(14)

\[ M_{70} \geq 100 \]  
(15)

The third set of constraints is the maximum cost (or crash cost) constraints:
\[ M_{10} \leq 120 \]  
(16)

\[ M_{20} \leq 170 \]  
(17)

\[ M_{30} \leq 150 \]  
(18)

\[ M_{40} \leq 400 \]  
(19)

\[ M_{50} \leq 200 \]  
(20)

\[ M_{50} \leq 990 \]  
(21)

\[ M_{70} \leq 150 \]  
(22)
Other constraints are:

\[ M_i \geq 0 \quad \text{for } i = 1, 2, \ldots, 7 \]

and \[ x_i \geq 0 \quad \text{for } i = 1, 2, \ldots, 7, \text{ END} \]

The objective function is:

Minimize \[ P = M_{10} + M_{20} + M_{30} + M_{40} + M_{50} + M_{60} + M_{70} + 60x_{\text{END}} \]

The solution for this model is:

Total minimum project cost = 2,640

\[ x_{10} = 0 \quad M_{10} = 100 \]
\[ x_{20} = 0 \quad M_{20} = 170 \]
\[ x_{30} = 5 \quad M_{30} = 150 \]
\[ x_{40} = 3 \quad M_{40} = 350 \]
\[ x_{50} = 3 \quad M_{50} = 200 \]
\[ x_{60} = 8 \quad M_{60} = 550 \]
\[ x_{70} = 14 \quad M_{70} = 100 \]
\[ x_{\text{END}} = 17 \]

.: Project duration = 17 weeks

We can observe that only \( M_{20} \) and \( M_{40} \) in the result are different from their normal cost \( U_{20} \) and \( U_{40} \) respectively. By comparing \( M_{20} \) and \( U_{20} \), we can know that activity 20 is shortened by 1 week. Similarly, we can know that activity 40 is also shortened by 1 week. The normal (original) project duration is 19 weeks (see Example 3.9) and the optimal project duration is found to be 17 weeks, because the two activities (20 and 40) are each shortened by 1 week.
Exercise

1. Formulate the following trans-shipment problem as a linear programming model.

```
+12 ① → ③ → ⑤ - 4
          +10 ② → ④ → ⑦ - 8
```

The unit cost from route to route are as follows:

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>33</td>
<td>45</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>28</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>-</td>
<td>13</td>
<td>16</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>19</td>
<td>18</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

2. A manufacturing process involves 4 stages I, II, III and IV by inputting two different raw materials A and B. There are four final products 1, 2, 3 and 4 coming out from the processes as shown in the following diagram:
250 m$^3$ and 200 m$^3$ of raw materials A and B respectively are available per week. Raw material A costs $2.6 \times 10^3$ per m$^3$ and raw material B $3.2 \times 10^3$ per m$^3$.

Plant 1 has a capacity of 400 m$^3$ per week. In plant 1, processing costs for process I and process II respectively are $2.5$ per m$^3$ and $5.0$ per m$^3$.

Plant 2 has a capacity of 200 m$^3$ per week. In plant 2, processing costs for each final product and the yields associated with the processes are:

<table>
<thead>
<tr>
<th>Final product</th>
<th>Process III</th>
<th>Process IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP 2</td>
<td>0.6</td>
<td>-</td>
</tr>
<tr>
<td>FP 3</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>FP 4</td>
<td>-</td>
<td>0.4</td>
</tr>
<tr>
<td>Loss</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Processing cost/m$^3$</td>
<td>$2.5/m^3$</td>
<td>$10.5/m^3$</td>
</tr>
</tbody>
</table>

The company has to supply in the next week a minimum of 50 m$^3$ of final product 1, a minimum of 25 m$^3$ of final product 2, a minimum of 60 m$^3$ of final product 3, and a minimum of 40 m$^3$ of final product 4 to its customers. If the
production of final product 3 is in excess, it will cost the company to store it at $250 per m³. The selling price of the final products are:

<table>
<thead>
<tr>
<th>Final Product</th>
<th>Selling price</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP 1</td>
<td>$3.4 \times 10^3$ per m³</td>
</tr>
<tr>
<td>FP 2</td>
<td>$9.4 \times 10^3$ per m³</td>
</tr>
<tr>
<td>FP 3</td>
<td>$7.4 \times 10^3$ per m³</td>
</tr>
<tr>
<td>FP 4</td>
<td>$7.6 \times 10^3$ per m³</td>
</tr>
</tbody>
</table>

Formulate a linear programming model for the problem so that the financial return of the company is maximized.
In Example 3.1 of Chapter 3, we have seen what a typical transportation problem is. Examples 3.3, 3.4 and 3.5 are also transportation problems although they are less obvious than example 3.1. Transportation problems can be solved, of course, by the simplex method. However, there is an algorithm which can also solve transportation problems without using the techniques of the simplex method. In this chapter, we will discuss the techniques of this new algorithm and will see how to use it to solve Examples 3.1, 3.3, 3.4 and 3.5.

4.1 The General Form of a Transportation Problem

Before discussing the general form of a transportation problem, we shall look at the particular form again which has been illustrated in Example 3.1. This example is now reproduced below for easy reference. Its transportation tableau is as follows:

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Customer 1</th>
<th>Customer 2</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>$x_{11}$</td>
<td>$x_{12}$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>$x_{21}$</td>
<td>$x_{22}$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>11</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>$x_{31}$</td>
<td>$x_{32}$</td>
<td></td>
</tr>
<tr>
<td>Demand</td>
<td>29</td>
<td>33</td>
<td></td>
</tr>
</tbody>
</table>

The warehouses 1, 2 and 3 supply 12, 20 and 30 tonnes of stock per week respectively to customers 1 and 2 who demand 29 and 33 tonnes of stock per week respectively. The cost of transporting one tonne of stock from warehouse
i to customer j is given in the small box at the left top corner of the decision variable $x_{ij}$. As already explained in Example 1.3, the linear programming model for this typical transportation problem is:

Minimize $P = 9x_{11} + 10x_{12} + 6x_{21} + 7x_{22} + 8x_{31} + 11x_{32}$

subject to

\[
\begin{align*}
    x_{11} + x_{12} & \leq 12 \\
    x_{21} + x_{22} & \leq 20 \\
    x_{31} + x_{32} & \leq 30 \\
    x_{11} + x_{21} + x_{31} & = 29 \\
    x_{12} + x_{22} + x_{32} & = 33
\end{align*}
\]

Subject to constraints

All $x_{ij} \geq 0$ for $i = 1, 2, 3$ and $j = 1, 2$

The above is a particular case of transportation problems. We can generalize it and a general transportation tableau is shown in Fig. 4.1 below.

We can see that there are $m$ supply stations and $n$ demand stations. $C_{ij}$ is the unit cost of transportation from supply station $i$ to demand station $j$. $S_i$ is the quantity available at supply station $i$ and $D_j$ is the quantity required at demand
station j. $C_{ij}$, $S_i$ and $D_j$ are given quantities in the problem. The decision variables $x_{ij}$ are the ones which we want to find out by solving the problem. The linear programming model can be formulated as:

$$\text{Minimize } P = C_{11}x_{11} + C_{12}x_{12} + \ldots + C_{ij}x_{ij} + \ldots + C_{mn}x_{mn}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}x_{ij} \quad \text{(in a more compact form)}$$

subject to

$$x_{11} + x_{12} + \ldots + x_{1n} \leq S_1 \quad \text{(1)}$$
$$x_{21} + x_{22} + \ldots + x_{2n} \leq S_2 \quad \text{(2)}$$
$$\vdots$$
$$x_{m1} + x_{m2} + \ldots + x_{mn} \leq S_m \quad \text{(m)}$$

$$x_{11} + x_{21} + \ldots + x_{m1} = D_1 \quad \text{(m+1)}$$
$$x_{12} + x_{22} + \ldots + x_{m2} = D_2 \quad \text{(m+2)}$$
$$\vdots$$
$$x_{1n} + x_{2n} + \ldots + x_{mn} = D_n \quad \text{(m+n)}$$

and all $x_{ij} \geq 0$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

4.2 The Algorithm
As mentioned before, there is an algorithm other than the simplex method to solve transportation problems. The algorithm is usually referred to as transportation algorithm. It will be described in this section using examples.

Example 4.1
Use the transportation algorithm to solve the transportation problem in Example 3.1.
Solution 4.1
There are four steps in the algorithm.

Step 1
Draw the transportation tableau as shown in Fig. 4.2.

<table>
<thead>
<tr>
<th></th>
<th>Customer 1</th>
<th>Customer 2</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warehouse 1</td>
<td>9</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Warehouse 2</td>
<td>6</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>Warehouse 3</td>
<td>8</td>
<td>11</td>
<td>30</td>
</tr>
<tr>
<td>Demand</td>
<td>29</td>
<td>33</td>
<td>62 (total)</td>
</tr>
</tbody>
</table>

Fig. 4.2 Transportation tableau for Example 4.1

It should be observed that in this example, the total supply (12 + 20 + 30 = 62) is equal to the total demand (29 + 33 = 62). This will make our problem simpler. Cases where total supply is not equal to total demand will be discussed later.

We must firstly find an initial feasible allocation. It is reasonable to start by allocating as many tonnes of stock as possible to the route i-j with the lowest $C_{ij}$, and in our example, this is route 2-1 ($C_{21} = 6$). The maximum quantity that can be allocated to this route is 20, because warehouse 2 can supply only 20 tonnes. This is followed by the allocation of the remaining demand to customer 1, who will demand a further 9 tonnes, making a total of 29 tonnes. The most inexpensive route to do so is through route 3-1 ($C_{31} = 8$).

After allocating 9 tonnes to route 3-1, the items of stock remained in warehouse 3 will be reduced to 21 (i.e. 30-9). Then, allocate 21 tonnes to route 3-2 so all 30 tonnes of stock in warehouse 3 will be used up. Then, allocate 12 tonnes to
route 1-2 so that the total demand of customer 2 can be satisfied. Note that the supply of warehouse 1 is also automatically satisfied because in this problem total supply is equal to total demand.

Now, we have finished the initial allocation and this initial feasible solution is shown in the transportation tableau in Fig. 4.3.

![Fig. 4.3 Initial feasible solution](image)

The total transportation costs for this initial feasible solution is

\[
20 \times 6 + 9 \times 8 + 12 \times 10 + 21 \times 11
\]

\[= 543\]

**Step 2**

Next, we have to see if this total costs can be further reduced by using the unused routes. So we examine the unused routes (1-1 and 2-2 in our example) one by one.

For route 1-1: if one tonne is allocated to route 1-1, then one tonne must be subtracted from route 3-1, one tonne added to route 3-2 and one tonne subtracted from route 1-2 in order to maintain the supply and demand to be satisfied (see Fig. 4.4).
The change in cost when 1 tonne of stock is allocated to route 1-1

\[ C_{11} - C_{31} + C_{32} - C_{12} \]

\[ = 9 - 8 + 11 - 10 \]

\[ = +2 \quad \text{(a positive cost change)} \]

We usually call such cost change (+2 in this case) the \textit{improvement index} of the unused route 1-1. A positive improvement index means that the use of route 1-1 will increase the transportation cost rather than reducing it. Therefore, we reject using route 1-1.

For route 2-2: if one tonne is allocated to route 2-1, then one tonne must be subtracted from route 2-1, one tonne added to route 3-1 and one tonne subtracted from route 3-2. Fig. 4.5 shows this modification.
The improvement index of the unused route 2-2

\[ = C_{22} - C_{21} + C_{31} - C_{32} \]

\[ = 7 - 6 + 8 - 11 \]

\[ = -2 \] (a negative cost change)

A negative improvement index indicates that a reduction in transportation cost is possible by using the unused route 2-2.

**Step 3**

Now, we try to utilize route 2-2 and use it as fully as possible. Let \( x \) be the maximum number of tonnes of stock that route 2-2 can be allocated (see Fig. 4.6).
By inspection, the maximum $x$ is 20 since route 2-1 cannot take a negative value if $x$ is large than 20. So we allocate 20 tonnes of stock to route 2-2. The new (or second) feasible solution is shown in Fig. 4.7.

The total transportation costs for this second feasible solution is

$$29 \times 8 + 12 \times 10 + 20 \times 7 + 1 \times 11$$

$$= 503$$
We can see that 503 now is lower than 543 which is calculated from the initial feasible solution.

**Step 4**

We now repeat step 2 to see if the unused routes 1-1 or 2-1 have negative improvement index.

Improvement index of route 1-1

\[ I_{1-1} = C_{11} - C_{31} + C_{32} - C_{12} = 9 - 8 + 11 - 10 = +2 \]

Improvement index of route 1-2

\[ I_{1-2} = C_{21} - C_{31} + C_{32} - C_{22} = 6 - 8 + 11 - 7 = +2 \]

There is no negative improvement index. Therefore, the second feasible solution is the optimal solution with total transportation costs equal to 503.

Readers should now compare this result with that of Example 3.1 in Chapter 3.

### 4.3 A Further Example

In Example 4.1, the total supply of the three warehouses is equal to the total demand of the two customers. We shall now see an example with different total supply and total demand.

**Example 4.2**

Redo Example 4.1 if the supply of stock in warehouses 1, 2 and 3 per week are 15, 35 and 32 tonnes respectively while the weekly demands of the customers remain unchanged.
Solution 4.2

Step 1

Since the total supply is greater than the total demand, we introduce a dummy customer, customer 3, and so a new column (column 3) is added to the transportation tableau. The $C_{3}$ in the column is equal to zero. This is shown in Fig. 4.8.

We can observe that the demand of customer 3 (the dummy customer) is 20 tonnes, so that the total demand is made equal to the total supply.

Step 2

Next, we use the same method as described in step 1 of the solution of Example 4.1 to obtain an initial feasible solution. In this case, the first allocation is to use route 3-3 and put 20 tonnes in this route. Then the remaining procedures are the same. The initial allocation is shown Fig. 4.9 below.
The unused routes are 1-1, 1-3, 2-3 and 3-2. So we have to find the improvement index of these routes.

**Improvement index of route 1-1**

\[
\text{Im}_{1-1} = C_{11} - C_{21} + C_{22} - C_{12}
\]

\[
= 9 - 6 + 7 - 10
\]

\[
= 0
\]

For finding the improvement index of route 1-3, the method is a little more complicated. The loop for it is not rectangular but irregular, as shown in Fig. 4.10.
The reason for using such a strange loop is that for finding the improvement index of a particular unused route, no other unused routes should be involved. Readers should have observed this point in the previous examples of calculating improvement indices.

Now, the improvement index of route 1-3 can be calculated by:

\[ C_{13} - C_{12} + C_{22} - C_{21} + C_{31} - C_{33} \]
\[ = 0 - 10 + 7 - 6 + 8 - 0 \]
\[ = -1 \]

The calculations of improvement indices of routes 2-3 and 3-2 are rather simple, and are shown as follows:

**Improvement index of route 2-3**

\[ = C_{23} - C_{21} + C_{31} - C_{33} \]
\[ = 0 - 6 + 8 - 0 \]
\[ = +2 \]
Improvement index of route 3-2
\[ = C_{32} - C_{22} + C_{21} - C_{31} \]
\[ = 11 - 7 + 6 - 8 \]
\[ = +2 \]

**Step 3**
Now, we know that the improvement indices of unused routes 1-1, 1-3, 2-3 and 3-2 are 0, -1, +2 and +2 respectively. We should use the unused route with the most negative improvement index, and in our case it is route 1-3. So, we use route 1-3 as fully as possible. Let \( x \) be the maximum number of tonnes of stock that route 1-3 can be allocated (see Fig. 4.11).

<table>
<thead>
<tr>
<th>Warehouse</th>
<th>Customer</th>
<th>1</th>
<th>2</th>
<th>3 (dummy)</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>9</td>
<td>10</td>
<td>0</td>
<td>+x</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>6</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>8</td>
<td>11</td>
<td>0</td>
<td>20 - x</td>
</tr>
<tr>
<td>Demand</td>
<td></td>
<td>29</td>
<td>33</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4.11 Allocating \( x \) to route 1-3

The maximum \( x \) is 15. So we allocate 15 tonnes of stock to route 1-3. The second feasible solution is shown in Fig. 4.12.
In the second feasible solution (Fig. 4.12), the unused routes are routes 1-1, 1-2, 2-3 and 3-2. The improvement indices of these unused routes are calculated as follows.

**Improvement index of route 1-1**

\[
\begin{align*}
\text{Improvement index of route 1-1} & = C_{11} - C_{31} + C_{33} - C_{13} \\
& = 9 - 8 + 0 - 0 \\
& = +1
\end{align*}
\]

**Improvement index of route 1-2**

\[
\begin{align*}
\text{Improvement index of route 1-2} & = C_{12} - C_{22} + C_{21} - C_{31} + C_{33} - C_{13} \\
& = 10 - 7 + 6 - 8 + 0 - 0 \\
& = +1
\end{align*}
\]

**Improvement index of route 2-3**

\[
\begin{align*}
\text{Improvement index of route 2-3} & = C_{23} - C_{21} + C_{31} - C_{33} \\
& = 0 - 6 + 8 - 0 \\
& = +2
\end{align*}
\]
Improvement index of route 3-2
\[ = C_{32} - C_{22} + C_{21} - C_{31} \]
\[ = 11 - 7 + 6 - 8 \]
\[ = +2 \]

Since all the improvement indices are positive, there cannot be further improvement. Therefore we stop here and conclude that the second feasible solution is the optimal solution. Curious readers may use the simplex method to check the solution.

We should have observed that there are usually \( m+n-1 \) used routes in a feasible solution. If fewer routes are used and the supply and demand conditions are satisfied, we call such a case **degeneracy**. When a non-optimal degenerate feasible solution occurs, we have to add a zero allocation to a route and treat it as a used route such that the total number of used routes is equal to \( m+n-1 \) in order that the algorithm can be continued.

### 4.4 More Applications of Transportation Algorithm

In Sections 4.2 and 4.3, we have seen how to use the transportation algorithm to solve a typical transportation problem. However, some linear programming problems which are less obvious compared with these two examples can also be solved by the algorithm. They are problems in Examples 3.3, 3.4 and 3.5 of Chapter 3.

#### 4.4.1 Trans-shipment Problem

Firstly, let us see Example 3.3 of Chapter 3. It is a trans-shipment problem. The problem is reproduced below for easy reference.
We can construct a transportation tableau for this problem. In the tableau, non-feasible routes (such as 2-1 or 4-3) are excluded by assigning $C_{ij}$ as a big-M unit cost. Routes from a node to the same node are assigned zero $C_{ij}$ where $i = j$. Fig.4.13 shows the required transportation tableau for this trans-shipment problem (the problem without modification, see Solution 3.3).

<table>
<thead>
<tr>
<th>Sources</th>
<th>Destinations</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>20</td>
<td>M</td>
<td>22</td>
<td>M</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>M</td>
<td>0</td>
<td>15</td>
<td>M</td>
<td>M</td>
<td>B - 14</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>M</td>
<td>M</td>
<td>0</td>
<td>11</td>
<td>13</td>
<td>B + 12</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>18</td>
<td>12</td>
<td>11</td>
<td>0</td>
<td>16</td>
<td>B</td>
</tr>
<tr>
<td>Demand</td>
<td></td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>16</td>
<td>4B + 16 (total)</td>
</tr>
</tbody>
</table>

Fig. 4.13 Transportation tableau for trans-shipment problem (Example 3.3)

In this problem, only node 1 is strictly a source and node 6 strictly a destination. All other nodes have arrows both in and out so that they can be both sources and destinations and are trans-shipment nodes. Node 1 has a supply of 18 and node 6 has a demand of 16 since they are strictly source and destination. A buffer
stock (i.e. B) that is sufficiently large for feasible moves must be introduced at the trans-shipment nodes. In this problem, the buffer stock B is equal to 30 because the total tonnes to be moved is 30 and is big enough for its purpose. For demands, all the nodes except node 6 (a strict demand node) have been given demands of B items (or 30 tonnes). But for supplies, they are a little more complicated. Those nodes with no net increase or decrease in stock (nodes 2 and 5) have demand equal to supply, are assigned value B. However, node 3, which has a net demand of 14, has been given a supply of B - 14 (or 30 - 14 = 16) tonnes. Node 4, which has a net supply of 12 tonnes, has been given a figure of B + 12 (or 30 + 12 = 42) tonnes.

Readers may use the transportation algorithm to solve this trans-shipment problem. The optimal solution is shown in Fig. 4.14.

<table>
<thead>
<tr>
<th>Destinations</th>
<th>Sources</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>12</td>
<td>20</td>
<td>M</td>
<td>22</td>
<td>M</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>0</td>
<td>16</td>
<td>15</td>
<td>M</td>
<td>M</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>M</td>
<td>M</td>
<td>0</td>
<td>30</td>
<td>11</td>
<td>M</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>12</td>
<td>11</td>
<td>0</td>
<td>26</td>
<td>M</td>
<td>4</td>
</tr>
<tr>
<td>Demand</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>16</td>
<td>136 (total)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4.14 Optimal solution for Example 3.3

Readers may compare this solution with the one given in Chapter 3 (Solution 3.3).
4.4.2 Earth Moving Problem

Now, let us look at Example 3.4 of Chapter 3. It is an earth moving optimization problem. In fact, it is very similar to a typical transportation problem. Its transportation tableau is shown in Fig. 4.15.

<table>
<thead>
<tr>
<th>Cut section</th>
<th>Fill Section</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>22 24 26 28 32 34</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>24 22 24 26 30 32</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>32 26 24 22 22 24</td>
<td>13</td>
</tr>
<tr>
<td>Demand</td>
<td>10 14 16 10 12 11</td>
<td>73 (total)</td>
</tr>
</tbody>
</table>

Fig. 4.15 Transportation tableau for earth moving problem (Example 3.4)

Readers may try solving this problem using the transportation algorithm and compare the result with that of Solution 3.4 in Chapter 3. The optimal solution is shown in Fig. 4.16.

<table>
<thead>
<tr>
<th>Cut section</th>
<th>Fill Section</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>22 10 24 8 26 28 32 34 10</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>24 22 6 24 16 24 30 32 10</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>32 26 24 22 22 24 2 11</td>
<td>13</td>
</tr>
<tr>
<td>Demand</td>
<td>10 14 16 10 12 11</td>
<td>73 (total)</td>
</tr>
</tbody>
</table>

Fig. 4.16 Optimal solution for Example 3.4
4.4.3 Production Schedule Problem

Example 3.5 of Chapter 3 is a production schedule problem. Apparently it is hard to do with transportation problem. However, it can actually be solved by using the transportation algorithm. The transportation tableau is shown in Fig. 4.17.

<table>
<thead>
<tr>
<th>Month (production)</th>
<th>Month (usage)</th>
<th>Excess capacity</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular time</td>
<td>2500</td>
<td>2650</td>
</tr>
<tr>
<td></td>
<td>Overtime</td>
<td>3000</td>
<td>3150</td>
</tr>
<tr>
<td>2</td>
<td>Regular time</td>
<td>M</td>
<td>2600</td>
</tr>
<tr>
<td></td>
<td>Overtime</td>
<td>M</td>
<td>3100</td>
</tr>
<tr>
<td>3</td>
<td>Regular time</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>Overtime</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>4</td>
<td>Regular time</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>Overtime</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>Demand</td>
<td>220</td>
<td>300</td>
</tr>
</tbody>
</table>

Fig. 4.17 Transportation tableau for production schedule problem (Example 3.5)

The optimal solution is shown in Fig. 4.18.
The optimal solution can be interpreted as follows:

**Month 1**: regular time production $= 200$

overtime production $= 50$ (30 of which is stored for month 2)

**Month 2**: regular time production $= 200$

overtime production $= 80$ (10 of which is stored for month 3)

**Month 3**: regular time production $= 200$

overtime production $= 80$ (40 of which is stored for month 4)

**Month 4**: regular time production $= 200$

overtime production $= 80$
4.5 An Interesting Example Using Transportation Algorithm

Before completing the chapter, let us see an interesting example involving the techniques of the transportation algorithm.

Example 4.3

A plant hire chain has excess stock of adjustable props at two stores and shortages at two others. The objective is to redistribute the stock at minimum transportation cost. The following tables give stock position and unit transportation cost respectively:

<table>
<thead>
<tr>
<th>Stores</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excess</td>
<td>40</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shortage</td>
<td>35</td>
<td>50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stock Position

<table>
<thead>
<tr>
<th>From store</th>
<th>To store</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Transportation Cost Matrix

(a) Find the solution that minimizes the total transportation costs of transporting excess stocks from Stores 1 and 2 to Stores 3 and 4.

(b) It is expected that at some time in the near future the unit transportation costs will increase. By letting the unit costs from store 1 increased by an amount $\alpha$ and those from store 2 increased by an amount $\beta$, show that there will be a new optimal solution if

$$(\alpha - \beta) + 10 < 0$$

(c) Find the relation between $\alpha$ and $\beta$ when the problem has more than one optimal solution.
Solution 4.3

(a) The problem can be presented as a transportation tableau in Fig. 4.19.

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>4</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80</td>
<td>70</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>Demand</td>
<td>35</td>
<td>50</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 4.19 Transportation tableau for Example 4.3

The optimal feasible solution is shown in Fig. 4.20.

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>4</th>
<th>(Dummy)</th>
<th>5</th>
<th>Supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80</td>
<td>70</td>
<td>0</td>
<td>15</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>60</td>
<td>0</td>
<td>25</td>
<td>60</td>
</tr>
<tr>
<td>Demand</td>
<td>35</td>
<td>50</td>
<td>15</td>
<td>100</td>
<td>(total)</td>
</tr>
</tbody>
</table>

Fig. 4.20 Optimal feasible solution

(II. = improvement index)

(I.I.)_{1-3} = C_{13} - C_{23} + C_{24} - C_{14}
= 80 - 50 + 60 - 70 = +20

(I.I.)_{2-5} = C_{25} - C_{15} + C_{14} - C_{24}
= 0 - 0 + 70 - 60 = +10

Since both I.I.s are positive, it is an optimal solution.
(b) In the near future, the unit cost from Store 1 will increase by $\alpha$, and that from Store 2 will increase by $\beta$. The transportation tableau will become the one as shown in Fig. 4.21.

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>4</th>
<th>(Dummy)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80+\alpha</td>
<td>70+\alpha</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>50+\beta</td>
<td>60+\beta</td>
<td>0</td>
<td>60</td>
</tr>
</tbody>
</table>

Demand | 35 | 50 | 15 | 100 (total)

Fig. 4.21 New transportation tableau

(I.I)_{1,3} = (80+\alpha) - (50+\beta) + (60+\beta) - (70+\alpha) = +20

(I.I)_{2,5} = (0) - (0) + (70+\alpha) - (60+\beta) = (\alpha-\beta) +10

There will be a new optimal solution if (I.I)_{2,5} = negative value

ie. $(\alpha-\beta) +10 < 0$

(c) The problem has more than one optimal solutions when (I.I)_{2,5} = 0

ie. $(\alpha-\beta) +10 = 0$

ie. $\alpha + 10 = \beta$

Exercise

1. Use the transportation algorithm to solve the trans-shipment problem given in Exercise 1 of Chapter 3. Hints: use a destination node “S” to equalize the total supply and total demand (see the following diagram). Unit costs of 1-S and 2-S are zero.
2. Example 3.2 of Chapter 3 can be treated as a trans-shipment problem and solved by the transportation algorithm if capacity constraints of distributors 1 and 2 (i.e. 40 and 35 respectively) are eliminated. Assuming that distributors 1 and 2 can store and distribute unlimited quantity of stocks, use the transportation algorithm to solve the problem.
In many linear optimization problems, we require that the decision variables be integers. Strictly speaking, Example 1.1 in Chapter 1 is an integer programming problem, because the optimal number of pipes must be whole numbers. Example 1.1, fortunately, has an optimal solution with decision variables in integers, and so we did not worry about the process of converting non-integers to whole numbers. However, we do not face every problem as lucky as Example 1.1. In this chapter, we shall look into linear optimization problems which require optimal solutions to have integer values, and, furthermore, the problems which require the decision variables to have either the values of 0 or 1 (called zero-one variables). This chapter will only concentrate on the formulation of such models. The next chapter will discuss the method of solution for such problems. Now, let us see a number of examples on integer programming formulation.

5.1 An Integer Programming Example

Example 5.1

A company is planning to invest a maximum amount of $1,000,000 in purchasing vehicles to deliver goods to customers from a central store. Details of the three types of vehicles available for purchase are given below:

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Capacity m$^3$</th>
<th>Price $</th>
<th>Average speed (km/hr)</th>
<th>Annual repair cost ($/year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>100,000</td>
<td>30</td>
<td>5,000</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>80,000</td>
<td>35</td>
<td>6,000</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>60,000</td>
<td>40</td>
<td>7,000</td>
</tr>
</tbody>
</table>

The total repair costs should not exceed $65,000 in a year. No more than 12 vehicles are required. Determine the number of vehicles of each type to be purchased so that the rate of delivery can be optimized.
Solution 5.1

Let $x_1$ = number of type 1 vehicles
$x_2$ = number of type 2 vehicles
$x_3$ = number of type 3 vehicles

Since the number of vehicles must be a whole number, therefore, $x_1$, $x_2$ and $x_3$ must be integers.

Rate of delivery is proportional to capacity, and rate of delivery is also proportional to average speed, ie.

- rate of delivery $\propto$ capacity
- rate of delivery $\propto$ average speed

$\therefore$ rate of delivery $\propto$ capacity x average speed

In order to optimize the rate of delivery, our objective function is:

Maximize $Z = (6 \times 30)x_1 + (5 \times 35)x_2 + (4 \times 40)x_3$

$= 180x_1 + 175x_2 + 160x_3$

subject to:

The capital constraint:

$100,000x_1 + 80,000x_2 + 60,000x_3 \leq 1,000,000 \quad \text{(1)}$

The repair cost constraint:

$5,000x_1 + 6,000x_2 + 7,000x_3 \leq 65,000 \quad \text{(2)}$

The total number of vehicles constraint:

$x_1 + x_2 + x_3 \leq 12 \quad \text{(3)}$

Therefore, the model can be summarized as follows:

Max $Z = 180x_1 + 175x_2 + 160x_3$

subject to

$100,000x_1 + 80,000x_2 + 60,000x_3 \leq 1,000,000 \quad \text{(1)}$

$5,000x_1 + 6,000x_2 + 7,000x_3 \leq 65,000 \quad \text{(2)}$

$x_1 + x_2 + x_3 \leq 12 \quad \text{(3)}$
\[ x_1, x_2, x_3 \geq 0 \] and are integers.

The solution for this model is:

\[
\text{Rate of delivery} = \sum (\text{capacity} \times \text{speed}) = 1955
\]

\[
x_1 = 6
\]
\[
x_2 = 5
\]
\[
x_3 = 0
\]

Note that if \( x_1 \) and \( x_2 \) were not constrained to be integers, the solution would be:

\[
\text{Rate of delivery} = \sum (\text{capacity} \times \text{speed}) = 2032.5
\]

\[
x_1 = 4
\]
\[
x_2 = 7.5
\]
\[
x_3 = 0
\]

### 5.2 Use of Zero-One Variables

In the beginning of this chapter, we have come across the term "zero-one variables". Some decision variables of some linear programming problems can only take the value of either 1 or 0. It is extremely powerful in the formulation of linear optimization models to use zero-one variable so that one can make a decision as yes or no by referring to whether the variable takes the value 1 or 0. Readers will see how powerful and useful these zero-one integer decision variables are in the following examples.

**Example 5.2**

A product can be manufactured by five different plant/equipment. The set-up costs (fixed costs), production costs (variables costs) per unit product processed, and the capacity of each plant/equipment per period are given as follows:
### Table of Fixed Set-up, Variable Production Cost, and Capacity per Period

<table>
<thead>
<tr>
<th>Plant/equipment</th>
<th>Fixed set-up cost ($)</th>
<th>Variable Production cost ($/unit)</th>
<th>Capacity per period (units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5500</td>
<td>90</td>
<td>700</td>
</tr>
<tr>
<td>2</td>
<td>5000</td>
<td>85</td>
<td>650</td>
</tr>
<tr>
<td>3</td>
<td>4500</td>
<td>91</td>
<td>600</td>
</tr>
<tr>
<td>4</td>
<td>4200</td>
<td>93</td>
<td>550</td>
</tr>
<tr>
<td>5</td>
<td>3800</td>
<td>95</td>
<td>500</td>
</tr>
</tbody>
</table>

1500 units of the product is demanded for a given period. Determine the optimal combination of plant/equipment to be used.

### Solution 5.2

Let

- \( x_1 \) = number of units produced by plant 1
- \( x_2 \) = number of units produced by plant 2
- \( x_3 \) = number of units produced by plant 3
- \( x_4 \) = number of units produced by plant 4
- \( x_5 \) = number of units produced by plant 5

Let \( d_i \) (i = 1, 2, 3, 4, 5) be zero-one variables such that \( d_i = 1 \) when plant \( i \) is set up and \( d_i = 0 \) when plant \( i \) is not set up so that:

\[
d_i = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \end{cases} \quad \text{for } i = 1, 2, 3, 4, 5
\]

The objective function is:

\[
\text{Minimize } P = 90x_1 + 85x_2 + 91x_3 + 93x_4 + 95x_5 \\
+ 5500d_1 + 5000d_2 + 4500d_3 + 4200d_4 + 3800d_5
\]

subject to

\[
x_1 + x_2 + x_3 + x_4 + x_5 = 1500 \quad \text{(1)}
\]

\[
x_1 \leq 700d_1 \quad \text{(2)}
\]

\[
x_2 \leq 650d_2 \quad \text{(3)}
\]

\[
x_3 \leq 600d_3 \quad \text{(4)}
\]
\[ x_4 \leq 550d_4 \]  \hspace{1cm} (5)
\[ x_5 \leq 500d_5 \]  \hspace{1cm} (6)

\[ x_1, x_2, x_3, x_4, x_5 \geq 0 \]

\[ d_1, d_2, d_3, d_4, d_5 \text{ are } 0-1 \text{ variables} \]

We call this a mixed-integer linear programming model because \( x_i \) are continuous variables and \( d_i \) are integer (zero-one) variables.

The solution for this model is:

Total production costs = 146,800

- \( x_1 = 0 \) \hspace{1cm} \( d_1 = 0 \)
- \( x_2 = 650 \) \hspace{1cm} \( d_2 = 1 \)
- \( x_3 = 600 \) \hspace{1cm} \( d_3 = 1 \)
- \( x_4 = 250 \) \hspace{1cm} \( d_4 = 1 \)
- \( x_5 = 0 \) \hspace{1cm} \( d_5 = 0 \)

So, only plant 2, 3 and 4 should be set up.

### 5.3 Transportation Problem With Warehouse Renting

**Example 5.3**

This example is an extension of Example 4.2 of Chapter 4. It is reproduced (with amendments) below diagramatically (Fig. 5.1) for easy reference.
Assume that all the three warehouses can be rented. One can rent all the warehouses or rent only part of them. The customer demand must be satisfied of course. The weekly rental of warehouses 1, 2 and 3 are $8,000, $10,000 and $9,500 respectively. Determine which warehouses should be rented and how many tonnes of stock from each warehouse should be transported to each customer.

**Solution 5.3**

This example is an extension of a standard transportation problem. The new element is to introduce zero-one variables to represent "to rent" or "not to rent" the warehouses. Now, let

\[
d_i = \begin{cases} 
1 & \text{if warehouse } i \text{ is rented} \\
0 & \text{if warehouse } i \text{ is not rented}
\end{cases} \quad \text{for } i = 1, 2, 3
\]

The objective function is:

Minimize \( P = 9x_{11} + 10x_{12} + 6x_{21} + 7x_{22} + 8x_{31} + 11x_{32} + 8000d_1 + 10000d_2 + 9500d_3 \)

subject to

The demand constraints:

\[
x_{11} + x_{21} + x_{31} = 29 \quad (1)
\]

\[
x_{12} + x_{22} + x_{32} = 33 \quad (2)
\]
The supply constraints:

\[ x_{11} + x_{12} \leq 15d_1 \]  \hspace{2cm} (3)
\[ x_{21} + x_{22} \leq 35d_2 \]  \hspace{2cm} (4)
\[ x_{31} + x_{32} \leq 32d_3 \]  \hspace{2cm} (5)

\[ x_{ij} \geq 0 \]
for \( i = 1, 2, 3 \) and \( j = 1, 2 \)

\[ d_i = \begin{cases} 
1 & \text{if } i = 1, 2, 3 \\
0 & \text{otherwise}
\end{cases} \]

Similar to Example 5.2, we call this a mixed-integer linear programming model because \( x_{ij} \) are continuous variables and \( d_i \) are integer (zero-one) variables.

The solution for this model is:

Total costs (transportation + rental) = 19,959

\[ x_{21} = 2 \quad d_1 = 0 \]
\[ x_{22} = 33 \quad d_2 = 1 \]
\[ x_{31} = 27 \quad d_3 = 1 \]

all other \( x_{ij} = 0 \)

So, only warehouses 2 and 3 are rented.

\section*{5.4 Transportation Problem With Additional Distributor}

\textbf{Example 5.4}

This example is an extension of Example 3.2 of Chapter 3. It is reproduced (with amendment) below (Fig. 5.2) for easy reference. Readers should refer to Example 3.2 for the definition of variables.
Assume that the management is considering whether or not to open a new distributor 3, and/or to expand distributor 1, and/or to close distributor 2 so that a maximum of two distributors will remain after replanning. The unit transportation costs from distributor 3 to customers 1 and 2 are as follows:

<table>
<thead>
<tr>
<th>Route i-j</th>
<th>Unit cost $C_{ij}$</th>
<th>Route j-k</th>
<th>Unit cost $C_{jk}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 3</td>
<td>5</td>
<td>3 - 1</td>
<td>4</td>
</tr>
<tr>
<td>2 - 3</td>
<td>4.5</td>
<td>3 - 2</td>
<td>5</td>
</tr>
<tr>
<td>3 - 3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The weekly storage capacity of the new distributor is 30 tonnes and this new distributor will pose an additional cost of $60 per week to the company. An expansion of distributor 1 from a weekly capacity of 40 tonnes to 55 tonnes will cost $30 per week. Closing distributor 2, on the other hand, will give the company a saving of $70 per week.

The management of the company wishes to know the optimal strategy.
Solution 5.4

Let \( d_1 = \begin{cases} 1 & \text{if distributor 1 is expanded} \\ 0 & \text{if distributor 1 is not expanded} \end{cases} \)

\[ d_2 = \begin{cases} 1 & \text{if distributor 2 is not closed} \\ 0 & \text{if distributor 2 is closed} \end{cases} \]

\[ d_3 = \begin{cases} 1 & \text{if distributor 3 is opened} \\ 0 & \text{if distributor 3 is not opened} \end{cases} \]

The objective function is:

\[
\text{Minimize } P = 4x_{11} + 6x_{12} + 5x_{13} + 5x_{21} + 4x_{22} + 4.5x_{23} + 5x_{31} + 3x_{32} + 4x_{33} + 3y_{11} + 5y_{12} + 6y_{21} + 4y_{22} + 4y_{31} + 5y_{32} + 9z_{11} + 10z_{12} + 6z_{21} + 7z_{22} + 8z_{31} + 11z_{32} + 30d_1 + 70d_2 + 60d_3
\]

subject to

The warehouse supply constraints:

\[
x_{11} + x_{12} + x_{13} + z_{11} + z_{12} \leq 12
\]

\[
x_{21} + x_{22} + x_{23} + z_{21} + z_{22} \leq 20
\]

\[
x_{31} + x_{32} + x_{33} + z_{31} + z_{32} \leq 30
\]

The distributor capacity constraints:

\[
x_{11} + x_{21} + x_{31} \leq 40 + 15d_1
\]

\[
x_{12} + x_{22} + x_{32} \leq 35d_2
\]

\[
x_{13} + x_{23} + x_{33} \leq 30d_3
\]

Note that when \( d_1 \) is equal to 0 (i.e. distributor 1 is not expanded), the capacity of distributor 1 remains to be 40. When \( d_2 \) is 0 (i.e. distributor 2 is closed), then \( x_{12}, x_{22}, \) and \( x_{32} \) will be 0. Also, when \( d_3 \) is 0 (i.e. distributor 3 is not opened), then \( x_{13}, x_{23} \) and \( x_{33} \) will be 0 too.

The distributor input-output constraints:

\[
x_{11} + x_{21} + x_{31} \geq y_{11} + y_{12}
\]

\[
x_{12} + x_{22} + x_{32} \geq y_{21} + y_{22}
\]

\[
x_{13} + x_{23} + x_{33} \geq y_{31} + y_{32}
\]
The customer demand constraints:
\[ z_{11} + z_{21} + z_{31} + y_{11} + y_{21} + y_{31} = 29 \]  \hspace{1cm} (10)
\[ z_{12} + z_{22} + z_{32} + y_{12} + y_{22} + y_{32} = 33 \]  \hspace{1cm} (11)

The maximum number of distributors constraint:
\[ d_1 + d_2 + d_3 \leq 2 \]  \hspace{1cm} (12)

all \( x_{ij} \geq 0, y_{jk} \geq 0, \quad z_{ik} \geq 0, \) and \( d_j = \begin{cases} 1 \\ 0 \end{cases} \) \( i = 1, 2, 3 \)
\( j = 1, 2, 3 \)
\( k = 1, 2 \)

The solution for this model is:
Total costs = 487
\[ x_{11} = 12 \quad y_{11} = 12 \quad z_{21} = 17 \quad d_1 = 0 \]
\[ x_{32} = 30 \quad y_{22} = 30 \quad z_{22} = 3 \quad d_2 = 1 \]
other \( x_{ij} = 0 \) other \( y_{jk} = 0 \) other \( z_{ik} = 0 \) \( d_3 = 0 \)

So, distributor 1 is not to be expanded and remains to be of capacity 40. Distributor 2 is not closed but distributor 3 is not to be opened.

5.5 Assignment Problem
An assignment problem is a special kind of transportation problem with \( m = n \), \( S_i = 1 \) and \( D_j = 1 \) (see Section 4.1 and Fig. 4.1 of Chapter 4). In the following we shall see an example of assignment problem with \( m = n = 5 \).

Example 5.5
A project manager is at the initial stage of undertaking a construction project which consists of five jobs, namely, job 1: tunnelling, job 2: building, job 3: bridge work, job 4: road work and job 5: earth work. There are five foremen reporting to him and he does not know which job should be assigned to which foreman. Therefore, he sets a test paper which consists of five questions, each question related to each job. The foremen take the test and answer all the five
questions (each question carries 20 marks). The scores obtained by the foremen on each question are shown in Fig. 5.3.

<table>
<thead>
<tr>
<th>Job</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>7</td>
<td>12</td>
<td>17</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>10</td>
<td>18</td>
<td>19</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>12</td>
<td>19</td>
<td>18</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>13</td>
<td>10</td>
<td>17</td>
<td>14</td>
</tr>
</tbody>
</table>

Fig. 5.3 Test results of five foremen on each question

How should the project manager assign jobs to the foremen based on the result of the test?

**Solution 5.5**

Our objective is to maximize the overall test scores. Since transportation problems are usually minimization problems, therefore we must convert this maximization problem into a minimization one. This can be done by using 20 (the full mark of each question) to subtract each score given in each box of Fig. 5.3, yielding the transportation tableau as shown in Fig. 5.4.
The objective function is:

Minimize \( P = \sum_{i=1}^{5} \sum_{j=1}^{5} C_{ij} x_{ij} \)

\[
= 5x_{11} + 13x_{12} + \cdots + 6x_{55}
\]

altogether 25 terms

subject to

The supply constraints:

\[
x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \quad (1)
\]

\[
x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1 \quad (2)
\]

\[
x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 1 \quad (3)
\]

\[
x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 1 \quad (4)
\]

\[
x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 1 \quad (5)
\]
The demand constraints:

\[
x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \quad (6)
\]

\[
x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \quad (7)
\]

\[
x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 1 \quad (8)
\]

\[
x_{14} + x_{24} + x_{34} + x_{44} + x_{54} = 1 \quad (9)
\]

\[
x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \quad (10)
\]

and all \( x_{ij} = \begin{cases} 1 & \text{for } i = 1, 2, 3, 4, 5 \text{ and } j = 1, 2, 3, 4, 5 \\ 0 & \text{otherwise} \end{cases} \)

Note that an assignment problem is an integer linear programming model because all the decision variables are zero-one variables.

The solution for this model is:

Minimum value of the objective function = 27

\[ \Rightarrow \text{total overall scores obtained} = 100 - 27 = 73 \]

\[
x_{11} = 1 \\
x_{24} = 1 \\
x_{33} = 1 \\
x_{42} = 1 \\
x_{55} = 1 
\]

So, job 1 should be assigned to foreman 1, job 2 to foreman 4, job 3 to foreman 3, job 4 to foreman 1, and job 5 to foreman 5.

5.6 Knapsack Problem

Example 5.6

A truck can be loaded with a maximum capacity of 12 tonnes. There are seven items of cargo with different weights and values as shown below:
How should the truck carry the cargo so that the total value of the cargo loaded is maximized and the maximum capacity of the truck is not exceeded?

**Solution 5.6**

This problem is a typical **knapsack problem** with the following model formulation.

Maximize \( Z = 3d_1 + 6d_2 + 2d_3 + 9d_4 + 4d_5 + 7d_6 + 5d_7 \)

subject to

\[
2d_1 + 3d_2 + d_3 + 4d_4 + 6d_5 + 5d_6 + 2d_7 \leq 12
\]

\[
d_i = \begin{cases} 
1 & \text{for } i = 1, 2, \ldots, 7 \\
0 & \text{otherwise}
\end{cases}
\]

The solution for this model is:

Total value of cargo on truck = 25 \times 10^3

\[
d_1 = 1 \\
d_2 = 1 \\
d_3 = 1 \\
d_4 = 1 \\
d_5 = 0 \\
d_6 = 0 \\
d_7 = 1
\]
5.7 Set-Covering Problem

Example 5.7

The headmaster of a secondary school is considering to employ teachers for English, Mathematics, Physics, Chemistry, Geography and Music. There are five teachers applying for the teaching posts. Applicant 1 can teach Mathematics, Physics, Chemistry and Geography; applicant 2 can teach English and Music; applicant 3 can teach Mathematics, Physics, Chemistry and Music; applicant 4 can teach Mathematics, Chemistry and Geography; applicant 5 can teach English, Mathematics and Music. The headmaster wants to select applicants so that:

(a) the number of teachers is minimized;
(b) the overall cost is minimized if the costs of employing applicants 1, 2, 3, 4 and 5 are 30, 20, 35, 25 and 18 (in thousand dollars) respectively.

Solution 5.7

The data for the problem are summarized in Fig. 5.5.

| Teacher  
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td></td>
<td>√</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Maths</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Physics</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Chemistry</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Geography</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Music</td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>30</td>
<td>20</td>
<td>35</td>
<td>25</td>
</tr>
</tbody>
</table>

Fig. 5.5 A set-covering problem

(a) Let \( d_i \) = \( \begin{cases} 1 & \text{applicant } i \text{ is employed} \\ 0 & \text{applicant } i \text{ is not employed} \end{cases} \) for \( i = 1, 2, 3, 4, 5 \)

If the number of teachers is to be minimized, the integer programming model will be:

Minimize \( P = d_1 + d_2 + d_3 + d_4 + d_5 \)

subject to

English: \( d_2 + d_5 \geq 1 \) \hspace{1cm} (1)
Maths: \( d_1 + d_3 + d_4 + d_5 \geq 1 \) \hspace{1cm} (2)

Physics: \( d_1 + d_3 \geq 1 \) \hspace{1cm} (3)

Chemistry: \( d_1 + d_3 + d_4 \geq 1 \) \hspace{1cm} (4)

 Geography: \( d_1 + d_4 \geq 1 \) \hspace{1cm} (5)

Music: \( d_2 + d_3 + d_5 \geq 1 \) \hspace{1cm} (6)

The solution for this model is:

Objective function = 2 \hspace{1cm} (i.e. only 2 applicants are employed)

\[
\begin{align*}
    d_1 &= 1 \\
    d_2 &= 1 \\
    d_3 &= 0 \\
    d_4 &= 0 \\
    d_5 &= 0
\end{align*}
\]

(b) If the overall cost is to be minimized, then the objective function will change to:

Minimize \( P = 30d_1 + 20d_2 + 35d_3 + 25d_4 + 18d_5 \)

All the constraints will be the same as in part (a).

The solution for this model is:

Minimum overall cost = \( 48 \times 10^3 \)

\[
\begin{align*}
    d_1 &= 1 \\
    d_2 &= 0 \\
    d_3 &= 0 \\
    d_4 &= 0 \\
    d_5 &= 1
\end{align*}
\]

5.8 Set-Packing Problem

A set-packing problem is a modification of the set-covering problem. The latter, as we have seen, has an objective that at least one teacher can teach one subject. The headmaster wants to ensure that all subjects are covered even if there is overlapping. In a set-packing problem, however, the headmaster has a different
objective. This time he wants to have as many of the subjects to be taught as possible but for each subject there cannot be more than one teacher. Let us see the following example.

**Example 5.8**

Reconsider example 5.7. This time the headmaster wishes to have as many subjects covered as possible but there must be no overlapping, that is, each subject can be taught by only one teacher. How should the headmaster select the applicants?

**Solution 5.8**

We have to assume the number of subjects known as the "value" of each teacher.

The objective function is:

Maximize $Z = 4d_1 + 2d_2 + 4d_3 + 3d_4 + 3d_5$

subject to

<table>
<thead>
<tr>
<th>Subject</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>$d_2 + d_5 \leq 1$</td>
</tr>
<tr>
<td>Maths</td>
<td>$d_1 + d_3 + d_4 + d_5 \leq 1$</td>
</tr>
<tr>
<td>Physics</td>
<td>$d_1 + d_3 \leq 1$</td>
</tr>
<tr>
<td>Chemistry</td>
<td>$d_1 + d_3 + d_4 \leq 1$</td>
</tr>
<tr>
<td>Geography</td>
<td>$d_1 + d_4 \leq 1$</td>
</tr>
<tr>
<td>Music</td>
<td>$d_2 + d_3 + d_5 \leq 1$</td>
</tr>
</tbody>
</table>

all $d_i = \begin{cases} 1 & \text{for } i = 1, 2, \ldots, 5 \\ 0 & \text{else} \end{cases}$

The solution for this model is:

Objective function = 6

$d_1 = 1$
$d_2 = 1$
$d_3 = 0$
$d_4 = 0$
$d_5 = 0$
5.9 Either-or Constraint (Resource Scheduling Problem)

In the following example (a resource scheduling problem), we shall see how zero-one variables can be used to model "either-or" conditions.

Example 5.9

There are three earthwork sites adjacent to each other undertaken by one contractor. Since the sites are so near to each other the transportation time for moving equipment from one site to another is assumed negligible. The contractor has only one backhoe, one bulldozer and one roller available and these plant items have to be shared among the three sites.

Site 1 needs the backhoe first, then the bulldozer and then the roller to work with. Site 2 needs the backhoe first and then the roller. Site 3 needs the bulldozer first and then the roller. The time durations for which they need the plant are shown below:

<table>
<thead>
<tr>
<th></th>
<th>Backhoe</th>
<th>Bulldozer</th>
<th>Roller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site 1</td>
<td>3 days</td>
<td>3 days</td>
<td>4 days</td>
</tr>
<tr>
<td>Site 2</td>
<td>4 days</td>
<td></td>
<td>5 days</td>
</tr>
<tr>
<td>Site 3</td>
<td></td>
<td>5 days</td>
<td>3 days</td>
</tr>
</tbody>
</table>

Help the contractor to find a schedule which will minimize the overall duration of completing the three jobs.

Solution 5.9

There are two sets of constraints for this problem. The first set is for ensuring the correct sequence of operations, and the second set is for ensuring that each plant is not scheduled on different sites at the same time.

Let us look at the first set of constraints.

In order to model a correct sequence of operations, let $x_{ij}$ be the start time at which plant $j$ is scheduled to be on site $i$, and $x_{\text{END}}$ be the total time required to complete all the three jobs as shown in Fig. 5.6.
In the above figure looks like a critical path problem, and in fact, it is. We have already discussed how to find the minimum overall duration of such a problem in Example 3.9 of Chapter 3. Using the method as described in Example 3.9, we have the following objective function and constraints:

Minimize \( P = x_{\text{END}} \)

subject to

\[
\begin{align*}
    x_{12} - x_{11} & \geq 3 \\
    x_{13} - x_{12} & \geq 3 \\
    x_{23} - x_{21} & \geq 4 \\
    x_{33} - x_{32} & \geq 5 \\
    x_{\text{END}} - x_{13} & \geq 4 \\
    x_{\text{END}} - x_{23} & \geq 5 \\
    x_{\text{END}} - x_{33} & \geq 3 \\
\end{align*}
\]

all \( x_{ij} \geq 0 \) (\( x_{ij} \) are continuous variables)

The second set of constraints is to ensure that each plant is not scheduled on different sites at the same time. In order to do so, we have to make use of "either-or" constraints. For example, plant 1 (i.e. backhoe) can only be on site 1 or site 2 at any given time. So, either plant 1 is on site 1 first and then on site 2, or vice versa. That is:
either \( x_{11} + 3 \leq x_{21} \) \\
or \( x_{21} + 4 \leq x_{11} \)

To model this, we have the following constraints:

\[
\begin{align*}
& x_{11} + 3 - x_{21} \leq M\delta_1 \quad \text{(i)} \\
& x_{21} + 4 - x_{11} \leq M\delta_2 \quad \text{(ii)} \\
& \delta_1 + \delta_2 = 1 \quad \text{(iii)}
\end{align*}
\]

where \( \delta_1 \) and \( \delta_2 \) are zero-one variables and \( M \) is a large positive number. When \( \delta_1 = 1 \) (i.e. \( \delta_2 = 0 \)), constraint (i) will become \( x_{11} + 3 - x_{21} \leq M \) and is inactive, but at the same time constraint (ii) will become \( x_{21} + 4 - x_{11} \leq 0 \), implying that plant 1 will be on site 2 first and then on site 1. Similarly, if \( \delta_2 = 1 \) (i.e. \( \delta_1 = 0 \)), it implies that plant 1 will be on site 1 first and then on site 2.

The three constraints (i), (ii) and (iii) appear to be a little clumsy and can be reduced to two constraints by reducing a zero-one variable as follows:

\[
\begin{align*}
& x_{11} + 3 - x_{21} \leq Md_1 \quad \text{(8)} \\
& x_{21} + 4 - x_{11} \leq M(1 - d_1) \quad \text{(9)}
\end{align*}
\]

In (8) and (9), \( d_1 \) is a zero-one variable. When \( d_1 = 1 \), plant 1 will be on site 2 first and then on site 1, and vice versa if \( d_1 = 0 \).

Similarly, we can now write down all the other constraints of the second set:

Plant 2:

\[
\begin{align*}
& x_{12} + 3 - x_{22} \leq Md_2 \quad \text{(10)} \\
& x_{22} + 5 - x_{12} \leq M(1 - d_2) \quad \text{(11)}
\end{align*}
\]

Plant 3:

\[
\begin{align*}
& x_{13} + 4 - x_{23} \leqMd_3 \quad \text{(12)} \\
& x_{23} + 5 - x_{13} \leq M(1 - d_3) \quad \text{(13)} \\
& x_{33} + 5 - x_{23} \leq Md_4 \quad \text{(14)}
\end{align*}
\]
\[ x_{33} + 3 - x_{23} \leq M(1 - d_4) \]  \hspace{1cm} (15)
\[ x_{13} + 4 - x_{33} \leq M d_5 \]  \hspace{1cm} (16)
\[ x_{33} + 3 - x_{13} \leq M(1 - d_5) \]  \hspace{1cm} (17)

all \( x_{ij} \geq 0 \) \hspace{0.5cm} i = 1, 2, 3 \hspace{0.5cm} j = 1, 2, 3

all \( d_k = \begin{cases} 1 \\ 0 \end{cases} \) \hspace{0.5cm} k = 1, 2, 3, 4, 5

\( M \) in this example can be taken as the sum of durations of all plant on all sites, that is 27, or any number larger than that.

The solution for this model is:
\[ x_{\text{END}} = \text{total duration} = 16 \text{ days} \]
\[ x_{11} = 4 \hspace{0.5cm} d_1 = 1 \]
\[ x_{12} = 9 \hspace{0.5cm} d_2 = 1 \]
\[ x_{13} = 12 \hspace{0.5cm} d_3 = 1 \]
\[ x_{21} = 0 \hspace{0.5cm} d_4 = 0 \]
\[ x_{23} = 4 \hspace{0.5cm} d_5 = 1 \]
\[ x_{32} = 4 \]
\[ x_{33} = 9 \]
\[ x_{\text{END}} = 16 \]

So, plant 1 (backhoe) will be on site 2 first and then on site 1. Plant 2 (bulldozer) will be on site 3 first and then on site 1. Plant 3 (roller) will be on site 2 first, then on site 3, and then on site 1.

5.10 Project Scheduling Problem

Example 5.10

A building developer is considering a development plan involving three office buildings. Block 1 will take three years to construct, and 105 workers will have to be assigned to it full-time. This block will contribute \( $100 \times 10^6 \) per year net to the developer after completion in the form of rental income. Block 2 will take two years to construct, will need 80 full-time workers, and will contribute \( $85 \times 10^6 \) per year net once completed. Block 3 will take four years to
construct, will need 120 full-time workers, and will contribute $115 \times 10^6$ per year net when finished. Once the construction of a block commences, the process will continue until it is completed, that is, it does not allow the construction to be stopped and then continued afterwards.

There are only 250 full-time workers available. The monetary constraints of the developer are such that he can start at most one building in any one year. All the three buildings must be completed by the end of year 5. Determine for the developer which office building(s) he should build and when he should start to build each of them in order to maximize the rental income.

**Solution 5.10**

Let $d_{ij}$ be zero-one variables such that $i$ represents the block number and $j$ the year which the construction of the block commences. All possible ways of constructing the blocks so that they can all be completed within five years are shown in Fig. 5.7.

\[
\begin{array}{ccc}
\text{Year 1} & d_{11} & d_{21} & d_{31} \\
\text{Year 2} & d_{11} & d_{12} & d_{21} & d_{22} & d_{31} & d_{32} \\
\text{Year 3} & d_{11} & d_{12} & d_{13} & d_{22} & d_{23} & d_{31} & d_{32} \\
\text{Year 4} & d_{12} & d_{13} & d_{22} & d_{24} & d_{31} & d_{32} \\
\text{Year 5} & d_{13} & d_{24} & d_{32} \\
\end{array}
\]

Fig. 5.7 Project scheduling problem

The objective is to maximize rental income and so we can write the objective function as:

\[
\text{Maximize } Z = (100 \times 2)d_{11} + 100d_{12} + (85 \times 3)d_{21} + (85 \times 2)d_{22} + 85d_{23} + 115d_{31}
\]

subject to:
Constraints ensuring that one building block is only built at most once:

\[
\begin{align*}
    d_{11} + d_{12} + d_{13} & \leq 1 \\
    d_{21} + d_{22} + d_{23} + d_{24} & \leq 1 \\
    d_{31} + d_{32} & \leq 1
\end{align*}
\]  

(1) (2) (3)

Constraints ensuring that the developer can start at most one building block in any one year:

\[
\begin{align*}
    d_{11} + d_{21} + d_{31} & \leq 1 \\
    d_{11} + d_{12} + d_{21} + d_{22} + d_{31} + d_{32} & \leq 2 \\
    d_{11} + d_{12} + d_{13} + d_{22} + d_{23} + d_{31} + d_{32} & \leq 3 \\
    d_{12} + d_{13} + d_{23} + d_{24} + d_{31} + d_{32} & \leq 3 \\
    d_{13} + d_{24} + d_{32} & \leq 3
\end{align*}
\]  

(4) (5) (6) (7) (8)

Constraints for labour:

\[
\begin{align*}
    105d_{11} + 80d_{21} + 120d_{31} & \leq 250 \\
    105d_{11} + 105d_{12} + 80d_{21} + 80d_{22} + 120d_{31} + 120d_{32} & \leq 250 \\
    105d_{11} + 105d_{12} + 105d_{13} + 80d_{22} + 80d_{23} + 120d_{31} + 120d_{32} & \leq 250 \\
    105d_{12} + 105d_{13} + 80d_{23} + 80d_{24} + 120d_{31} + 120d_{32} & \leq 250 \\
    105d_{13} + 80d_{24} + 120d_{32} & \leq 250
\end{align*}
\]  

(9) (10) (11) (12) (13)

\[
\begin{align*}
    \text{all } d_{ij} = \begin{cases} 
    1 \\
    0
\end{cases}
\end{align*}
\]

The solution for this model is:

\[
\begin{align*}
    \text{Total rental income} & = 370 \times 10^6 \\
    d_{11} &= 1 \\
    d_{21} &= 0 \\
    d_{31} &= 0 \\
    d_{12} &= 0 \\
    d_{22} &= 1 \\
    d_{32} &= 0 \\
    d_{13} &= 0 \\
    d_{23} &= 0 \\
    d_{24} &= 0
\end{align*}
\]

So, construct building block 1 in the first year and building block 2 in the second year. Block 3 is never constructed because of insufficient workers.
5.11 Travelling Salesman Problem

Example 5.11
A man is at town 1 but wishes to travel to towns 2, 3, 4, 5 and then come back to town 1. The distances between towns are shown in Fig. 5.8.

In what order should the salesman travel from town to town so that the total distance of his journey is minimized?

Solution 5.11
The distances between towns can be summarized in a matrix as shown in Fig. 5.9.

```
From          Town 1  Town 2  Town 3  Town 4  Town 5
------------- ------------- ------------- ------------- ------------- ------------- 
  1            0       7       3       8       6
  2            7       0       6.5     5.5     10.5
  3            3       6.5     0       5       4
  4            8       5.5     5       0       7.5
  5            6       10.5    4       7.5     0
```

Fig. 5.9 Distances between towns ($C_{ij}$)

Let $d_{ij} = \begin{cases} 1 & \text{route } i-j \text{ is taken} \\ 0 & \text{route } i-j \text{ is not taken} \end{cases}$
Note that: (a) $i \neq j$

(b) $d_{ij}$ and $d_{ji}$ are different variables

The objective is to minimize total distance travelled, i.e.

$$\text{Minimize } P = \sum_{i=1}^{5} \sum_{j=1}^{5} C_{ij} \cdot d_{ij} \quad \text{for } i \neq j$$

$$= 7d_{12} + 3d_{13} + 8d_{14} + 6d_{15} + 7d_{21} + 6.5d_{23} + 5.5d_{24} + 10.5d_{25}$$

$$+ 3d_{31} + 6.5d_{32} + 5d_{34} + 8d_{41} + 5.5d_{42} + 5d_{43} + 7.5d_{45}$$

$$+ 6d_{51} + 10.5d_{52} + 4d_{53} + 7.5d_{54}$$

subject to

Constraints that ensuring one route must begin at each town:

$$d_{12} + d_{13} + d_{14} + d_{15} = 1 \quad \text{(1)}$$

$$d_{21} + d_{23} + d_{24} + d_{25} = 1 \quad \text{(2)}$$

$$d_{31} + d_{32} + d_{34} + d_{35} = 1 \quad \text{(3)}$$

$$d_{41} + d_{42} + d_{43} + d_{45} = 1 \quad \text{(4)}$$

$$d_{51} + d_{52} + d_{53} + d_{54} = 1 \quad \text{(5)}$$

Constraints that ensuring one route must end at each town:

$$d_{21} + d_{31} + d_{41} + d_{51} = 1 \quad \text{(6)}$$

$$d_{12} + d_{32} + d_{42} + d_{52} = 1 \quad \text{(7)}$$

$$d_{13} + d_{23} + d_{43} + d_{53} = 1 \quad \text{(8)}$$

$$d_{14} + d_{24} + d_{34} + d_{54} = 1 \quad \text{(9)}$$

$$d_{15} + d_{25} + d_{35} + d_{45} = 1 \quad \text{(10)}$$

$$\text{all } d_{ij} = \begin{cases} 1 & \text{for } i = 1, 2, \ldots, 5 \\ 0 & \text{for } j = 1, 2, \ldots, 5 \\ \text{and } i \neq j \end{cases}$$

The solution for this model is:

Total distance travelled $= 24$

$$d_{15} = 1$$

$$d_{24} = 1$$
We can observe that the answer gives two discontinuous subtours. One subtour is town 1 to town 5, town 5 to town 3, and town 3 to town 1. Another subtour is town 2 to town 4 and town 4 to town 2. These are two disconnected loops. However, what we require is not that but a complete and connected tour of all towns. Therefore, constraints (1) through (10) alone are not adequate to model what we want. We have to add some more constraints.

If there are \( n \) towns to be visited (starting from town 1), then introduce an integer variable \( x_i \) (\( i = 1, 2, \ldots, n \)) for each town such that if town \( i \) is the \( k \)th town the salesman visits then \( x_i = k \). For example, if he starts from town 1, then \( x_1 = 1 \). If the next town he visits is town 2, then \( x_2 = 2 \) and \( d_{12} = 1 \). If town 4 is the one he visits after town 2, then \( x_4 = 3 \) and \( d_{24} = 1 \), and so on. Therefore, if town \( j \) is the town he visits immediately after town \( i \), then \( d_{ij} = 1 \), and if town \( j \) is not the town he visits immediately after town \( i \), then \( d_{ij} = 0 \), but no matter what, \( |x_j - x_i| \leq n - 1 \) because \( x_i \) and \( x_j \) must take the values from 1 to \( n \).

So, we have the conditions that:

(i) \( x_1 = 1 \) because the salesman starts at town 1
(ii) If \( x_j - x_i = 1 \) then \( d_{ij} = 1 \), else \( d_{ij} = 0 \)
(iii) \(|x_j - x_i| \leq n - 1\)

The above conditions can be modelled by the following additional constraints:

\[
x_1 = 1
\]

and \( x_j - x_i \geq nd_{ij} - (n - 1) \) for \( i = 1, 2, \ldots, n \)
\[
j = 1, 2, \ldots, n
\]
\[
i \neq j
\]

Now, let us go back to our Example 5.11. The extra constraints we need are:
For $i = 1, 2, \ldots, 5$

\[
x_i \geq 0 \text{ and are integers for } i = 1, 2, \ldots, 5
\]

\[
d_{ij} = \begin{cases} 1 & \text{for } i = 1, 2, \ldots, 5, j = 1, 2, \ldots, 5, i \neq j \\ 0 & \end{cases}
\]

The solution for this complete model is:

Total distance travelled = 27.5

\[
d_{12} = 1 \quad x_1 = 1
\]

\[
d_{24} = 1 \quad x_2 = 2
\]

\[
d_{31} = 1 \quad x_3 = 5
\]

\[
d_{45} = 1 \quad x_4 = 3
\]

\[
d_{53} = 1 \quad x_5 = 4
\]

This time we have a complete tour. The salesman goes to town 2 first from town 1, then from town 2 to town 4, then to town 5, then to town 3, and then back to town 1.
Exercise

1. A site manager has a gang of carpenters, a gang of steel fixers and a gang of concretors working under him. There are three jobs on the site. The first job requires the carpenters gang and then the concretors gang. The second job requires the steel fixers gang and then the concretors gang. The third job requires the carpenters gang, then the steel fixers gang and then the concretors gang. One gang can only work on one job at one time. The time required by the gangs on each job is given by:

<table>
<thead>
<tr>
<th></th>
<th>carpenters gang</th>
<th>steel fixers gang</th>
<th>concretors gang</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1</td>
<td>2 days</td>
<td></td>
<td>1/2 day</td>
</tr>
<tr>
<td>Job 2</td>
<td></td>
<td>3 days</td>
<td>1 day</td>
</tr>
<tr>
<td>Job 3</td>
<td>4 days</td>
<td>5 days</td>
<td>1 1/2 days</td>
</tr>
</tbody>
</table>

Help the site manager to minimize the completion time of these three jobs by formulating the problem as a mixed integer programming model.

2. A contractor is looking for more jobs. Four jobs are available and they are: (1) an office building, (2) a shopping complex, (3) a high rise apartment, and (4) a low rise residential development consisting of 10 houses. If low rise housing development is selected the contractor must inform the developer how many houses he is able to undertake so that the remaining one will be given to another contractor by the developer. The estimated profit from each job is estimated as follows:

(1) office building --- $300 \times 10^3$
(2) shopping complex --- $360 \times 10^3$
(3) high rise apartment --- $450 \times 10^3$
(4) low rise houses --- $22 \times 10^3$ per house

Although the contractor would like to do all of them, he has certain resource constraints. Each job will need a project manager, but he has only three managers available. Jobs 1, 2 and 3 each need a crane, but he has only two cranes in hand. During the period of construction of the 4 jobs, the contractor is only confident in finding $160 \times 10^3$ man-hours of skilled labour and $360 \times 10^3$
man-hours of unskilled labour for the jobs in the construction market. The man-hours of skilled and unskilled labour required by the four jobs are as follows:

<table>
<thead>
<tr>
<th>Job Description</th>
<th>Skilled Labour</th>
<th>Unskilled Labour</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Office building</td>
<td>40,000</td>
<td>80,000</td>
</tr>
<tr>
<td>(2) Shopping complex</td>
<td>55,000</td>
<td>125,000</td>
</tr>
<tr>
<td>(3) High rise apartment</td>
<td>70,000</td>
<td>165,000</td>
</tr>
<tr>
<td>(4) Low rise houses (per house)</td>
<td>6,000</td>
<td>14,000</td>
</tr>
</tbody>
</table>

The problems are:
(a) How should the contractor select the jobs in order to maximize the total profit? If the low rise houses are selected, how many houses should he build?
(b) If the contractor can hire an additional crane at $15,000 for the period of construction, should he hire it?
(c) If he can pay the skilled and unskilled labourers to do overtime work at extra costs of $12 and $8 per hour respectively, what will be his change of decisions?

3. A man is now in Hong Kong. He wants to travel by air to six other cities, namely, Bangkok, Beijing, Jakarta, Manila, Singapore and Tokyo. The flight distances in km from city to city are:

<table>
<thead>
<tr>
<th></th>
<th>Hong Kong</th>
<th>Bangkok</th>
<th>Beijing</th>
<th>Jakarta</th>
<th>Manila</th>
<th>Singapore</th>
<th>Tokyo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hong Kong</td>
<td>0</td>
<td>1720</td>
<td>1990</td>
<td>3270</td>
<td>1130</td>
<td>2590</td>
<td>3210</td>
</tr>
<tr>
<td>Bangkok</td>
<td>1720</td>
<td>0</td>
<td>3300</td>
<td>2340</td>
<td>2200</td>
<td>1440</td>
<td>4610</td>
</tr>
<tr>
<td>Beijing</td>
<td>1990</td>
<td>3300</td>
<td>0</td>
<td>5390</td>
<td>2880</td>
<td>4490</td>
<td>2110</td>
</tr>
<tr>
<td>Jakarta</td>
<td>3270</td>
<td>2340</td>
<td>5390</td>
<td>0</td>
<td>2780</td>
<td>900</td>
<td>5800</td>
</tr>
<tr>
<td>Manila</td>
<td>1130</td>
<td>2200</td>
<td>2880</td>
<td>2780</td>
<td>0</td>
<td>2380</td>
<td>3020</td>
</tr>
<tr>
<td>Singapore</td>
<td>2590</td>
<td>1440</td>
<td>4490</td>
<td>900</td>
<td>2380</td>
<td>0</td>
<td>5330</td>
</tr>
<tr>
<td>Tokyo</td>
<td>3210</td>
<td>4610</td>
<td>2110</td>
<td>5800</td>
<td>3020</td>
<td>5330</td>
<td>0</td>
</tr>
</tbody>
</table>

In what order should the man travel to the cities and finally come back to Hong Kong so that the total distance travelled is minimized?
The most powerful method of finding solutions for integer linear programming problems is the branch and bound method. In this chapter, we will consider two examples using the branch and bound method. One example is a problem with decision variables greater than zero but which must be integers. Another example is a problem with zero-one variables.

6.1 An Example of Integer Linear Programming Solutioning

Example 6.1

Solve, using the branch and bound method, the following integer linear programming model:

\[
\begin{align*}
\text{Max } Z &= 6x_1 + 8x_2 \\
\text{subject to} & \\
2x_1 + x_2 & \leq 6 \\
2x_1 + 3x_2 & \leq 9 \\
x_1, x_2 & \geq 0 \text{ and integers}
\end{align*}
\]

Solution 6.1

There are 4 steps in solving the model

Step 1

First of all, solve the model as if \(x_1\) and \(x_2\) are not restricted to be integers. We call this problem 1 (or P1):

\[
\begin{align*}
\text{Max } Z &= 6x_1 + 8x_2 \\
\text{subject to} & \\
2x_1 + x_2 & \leq 6 \\
2x_1 + 3x_2 & \leq 9 \\
x_1, x_2 & \geq 0
\end{align*}
\]

The optimal solution, found by the simplex method or the graphical method (see Fig. 6.1), of P1 is:
Z = 25.5
\( x_1 = 2.25 \)
\( x_2 = 1.5 \)

Fig. 6.1 Graphical solution of P1

Since \( x_1 \) and \( x_2 \) are not integers, we must go to step 2.

**Step 2**

Divide P1 into two problems, P2 and P3. We branch on \( x_2 \) by adding the bound \( x_2 \leq 1 \) to P1 forming P2, and adding the bound \( x_2 \geq 2 \) to P1 forming P3. The branching variable \( x_2 \) is chosen because it is less near to an integer than \( x_1 \). The following shows the two branched and bound problems P2 and P3:

**P2**

\[
\begin{align*}
\text{Max } Z &= 6x_1 + 8x_2 \\
\text{subject to} \\
2x_1 + x_2 &\leq 6 \\
2x_1 + 3x_2 &\leq 9 \\
x_2 &\leq 1 \\
x_1, x_2 &\geq 0 
\end{align*}
\]

**P3**

\[
\begin{align*}
\text{Max } Z &= 6x_1 + 8x_2 \\
\text{subject to} \\
2x_1 + x_2 &\leq 6 \\
2x_1 + 3x_2 &\leq 9 \\
x_2 &\geq 2 \\
x_1, x_2 &\geq 0 
\end{align*}
\]

As a result of the branch and bound, the feasible spaces for P2 and P3 are separated out from the original feasible space of P1, as shown in Fig. 6.2.
The optimal solutions, found by the simplex method or the graphical method, of P2 and P3 are:

**P2**

\[
\begin{align*}
Z &= 23 \\
x_1 &= 2.5 \\
x_2 &= 1
\end{align*}
\]

**P3**

\[
\begin{align*}
Z &= 25 \\
x_1 &= 1.5 \\
x_2 &= 2
\end{align*}
\]

We can see that the solutions of P2 and P3, although still not all integers, are closer to what we want after the first branch and bound. The branch and bound process intentionally changes the non-integral \( x_2 \) of P1 into an integer.

Usually, we use a tree diagram to represent the branch and bound process as shown in Fig. 6.3.
Fig. 6.3 Tree diagram for branch and bound process (a)

Now, we go to step 3.

Step 3

Since neither P2 nor P3 has an all-integer solution, we have to do further branching. We now face the question of which one we should branch first, P2 or P3? The answer is easy. Since the Z value of P3 (i.e. 25) is larger than that of P2 (i.e. 23), we branch P3 first as it has a higher chance of obtaining an all-integer solution with larger Z than P2. So, we divide P3 into two problems, P4 and P5. This time, we branch on $x_1$ by adding the bounds $x_1 \leq 1$ and $x_1 \geq 2$ to P3 forming P4 and P5 respectively as follows:

<table>
<thead>
<tr>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Z = 6$x_1$ + 8$x_2$</td>
<td>Max Z = 6$x_1$ + 8$x_2$</td>
</tr>
<tr>
<td>subject to</td>
<td>subject to</td>
</tr>
<tr>
<td>2$x_1$ + $x_2$ $\leq$ 6</td>
<td>2$x_1$ + $x_2$ $\leq$ 6</td>
</tr>
<tr>
<td>2$x_1$ + 3$x_2$ $\leq$ 9</td>
<td>2$x_1$ + 3$x_2$ $\leq$ 9</td>
</tr>
<tr>
<td>$x_2$ $\geq$ 2</td>
<td>$x_2$ $\geq$ 2</td>
</tr>
<tr>
<td>$x_1$ $\leq$ 1</td>
<td>$x_1$ $\geq$ 2</td>
</tr>
<tr>
<td>$x_1$, $x_2$ $\geq$ 0</td>
<td>$x_1$, $x_2$ $\geq$ 0</td>
</tr>
</tbody>
</table>
The optimal solutions, found by the simplex method, of P4 and P5 are:

**P4**
\[ Z = 24.67 \]
\[ x_1 = 1 \]
\[ x_2 = 2.33 \]

**P5**
No feasible solution

P5 has no feasible solution. To denote this, we put a "T" sign against it. Once a "T" sign is seen, we know that the branch is terminated there, as shown in Fig. 6.4.

![Tree diagram for branch and bound process (b)](image)
Step 4

Divide P4 into P6 and P7 with branching on \( x_2 \). P6 and P7 are as follows:

**P6**  
Max \( Z = 6x_1 + 8x_2 \)  
subject to  
\[
\begin{align*}
2x_1 + x_2 & \leq 6 \\
2x_1 + 3x_2 & \leq 9 \\
x_2 & \geq 2 \\
x_1 & \leq 1 \\
x_2 & \leq 2 \\
x_1, x_2 & \geq 0
\end{align*}
\]

**P7**  
Max \( Z = 6x_1 + 8x_2 \)  
subject to  
\[
\begin{align*}
2x_1 + x_2 & \leq 6 \\
2x_1 + 3x_2 & \leq 9 \\
x_2 & \geq 2 \\
x_1 & \leq 1 \\
x_2 & \geq 3 \\
x_1, x_2 & \geq 0
\end{align*}
\]

The optimal solutions, found by the simplex method, of P6 and P7 are:

**P6**  
\[
\begin{align*}
Z &= 22 \\
x_1 &= 1 \\
x_2 &= 2
\end{align*}
\]

**P7**  
\[
\begin{align*}
Z &= 24 \\
x_1 &= 0 \\
x_2 &= 3
\end{align*}
\]

The tree diagram now becomes the one as shown in Fig. 6.5.
If we compare P2, P6 and P7, P7 has the largest Z value. Since P7 is already an all-integer solution, so we have already found a better solution than we can possibly find from P2 and P6. Further branching for P2 or P6, therefore, is not necessary. Hence, we ignore P2 and P6 and terminate them by putting a "T" sign against each of them. The optimal solution is P7 for this example.

The QSB+ (Quantitative Systems for Business Plus) software has the function of solving integer linear programming problems. The user only has to enter P1
into the computer and also define that \( x_1 \) and \( x_2 \) are integers. The optimal solution (i.e. result of P7) will be shown after a series of calculations are automatically done by the computer.

### 6.2 Solutioning for Models With Zero-One Variables

**Example 6.2**

Solve the following knapsack problem:

Max \( Z = 20d_1 + 12d_2 + 14d_3 + 3d_4 + d_5 \)

subject to

\[
45d_1 + 20d_2 + 30d_3 + 13d_4 + 6d_5 \leq 51
\]

\[
d_i = \begin{cases} 
1 & \text{for } i = 1, 2, \ldots, 5 \\
0 & \text{otherwise}
\end{cases}
\]

**Solution 6.2**

**Step 1**

First of all, solve P1 where P1 is:

Max \( Z = 20d_1 + 12d_2 + 14d_3 + 3d_4 + d_5 \)

subject to

\[
45d_1 + 20d_2 + 30d_3 + 13d_4 + 6d_5 \leq 51
\]

\[
d_1 \leq 1
\]

\[
d_2 \leq 1
\]

\[
d_3 \leq 1
\]

\[
d_4 \leq 1
\]

\[
d_5 \leq 1
\]

all \( d_i \geq 0 \) for \( i = 1, 2, \ldots, 5 \)

The solution of P1 is

\[
Z = 26.44
\]

\[
d_1 = 0.022
\]
\[ d_2 = 1 \]
\[ d_3 = 1 \]
\[ d_4 = 0 \]
\[ d_5 = 0 \]

**Step 2**

So, we divide P1 into P2 and P3 and solve them as shown in Fig. 6.6.

![Tree diagram](image)

Fig. 6.6  Tree diagram for the knapsack problem (a)

**Step 3**

We tackle P2 first because it has a larger Z value. Divide P2 into P4 and P5 and solve them as shown in Fig. 6.7.
Comparing P3, P4 and P5, P4 has the largest Z value. Therefore, divide P4 into P6 and P7 as shown in the tree diagram in Fig. 6.8.

**Step 4**
<table>
<thead>
<tr>
<th>P1</th>
<th>Z = 26.44</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0.022</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 1</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0</td>
</tr>
</tbody>
</table>

d₁ ≤ 0 → P2

<table>
<thead>
<tr>
<th>P2</th>
<th>Z = 26.23</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 1</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0.077</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0</td>
</tr>
</tbody>
</table>

d₁ ≥ 1 → P3

<table>
<thead>
<tr>
<th>P3</th>
<th>Z = 23.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 1</td>
</tr>
<tr>
<td></td>
<td>d₂ = 0.3</td>
</tr>
<tr>
<td></td>
<td>d₃ = 0</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P4</th>
<th>Z = 26.17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 1</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0.17</td>
</tr>
</tbody>
</table>

d₄ ≤ 0 → P4

<table>
<thead>
<tr>
<th>P5</th>
<th>Z = 23.4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 0.6</td>
</tr>
<tr>
<td></td>
<td>d₄ = 1</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0</td>
</tr>
</tbody>
</table>

d₄ ≥ 1 → P5

<table>
<thead>
<tr>
<th>P6</th>
<th>Z = 26</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 1</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0</td>
</tr>
<tr>
<td></td>
<td>d₅ = 0</td>
</tr>
</tbody>
</table>

Optimal solution

<table>
<thead>
<tr>
<th>P7</th>
<th>Z = 24.67</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d₁ = 0</td>
</tr>
<tr>
<td></td>
<td>d₂ = 1</td>
</tr>
<tr>
<td></td>
<td>d₃ = 0.83</td>
</tr>
<tr>
<td></td>
<td>d₄ = 0</td>
</tr>
<tr>
<td></td>
<td>d₅ = 1</td>
</tr>
</tbody>
</table>

|   | T |

Fig. 6.8 Tree diagram for the knapsack problem (c)
P6 yields an all-integer solution and therefore is a feasible solution. Comparing P6 with P3, P5 and P7, it has the largest Z value. Hence, we can conclude that P6 is the optimal solution and put a terminal sign "T" against each of P3, P5 and P7.

Note that P6 is in fact the solution of the following non-integer linear programming model:

\[
\text{Max } Z = 20d_1 + 12d_2 + 14d_3 + 3d_4 + d_5 \\
\text{subject to} \\
45d_1 + 20d_2 + 30d_3 + 13d_4 + 6d_5 \leq 51 \\
d_1 \leq 1 \\
d_2 \leq 1 \\
d_3 \leq 1 \\
d_4 \leq 1 \\
d_5 \leq 1 \\
d_1 \geq 0 \\
d_4 \geq 0 \\
d_5 \geq 0 \\
\text{all } d_i \geq 0 \text{ for } i = 1, 2, \ldots, 5
\]

**Exercise**

Solve the following integer programming model and draw the tree diagram of the branch and bound process.

\[
\text{Max } Z = 100x_1 + 60x_2 + 70x_3 + 15x_4 + 15x_5 \\
\text{subject to} \\
52x_1 + 23x_2 + 35x_3 + 15x_4 + 7x_5 \leq 60 \\
x_i = \begin{cases} 
1 & \text{for } i = 1, 2, \ldots, 5 \\
0 & \text{otherwise}
\end{cases}
\]
7.1 Linear Programming Versus Goal Programming

In previous chapters, all examples involved the maximization or minimization of one single objective under constraints. In goal programming, it is possible for us to optimize more than one objective in a problem. In fact, an ordinary linear programming model (with only one objective) can also be formulated as a goal programming model. The first example in this chapter will illustrate this. From the second example onward, we will see how multiple objectives are optimized under a number of constraints.

Example 7.1
A furniture company sells two types of armchairs, the standard and the deluxe. Each of these armchairs requires time in the woodwork and the painting departments. The marketing manager of the company indicates that no more than 50 deluxe armchairs can be sold in the next week and is not sure about the situation for standard armchairs. The unit profit for the standard and deluxe armchairs are $400 and $500 respectively. A standard armchair requires 1 man-hour in the woodwork department and 1 man-hour in the painting department. A deluxe armchair, on the other hand, requires 2 and 1 man-hours in these respective departments. The number of man-hours in the woodwork department is limited to 120 and that in painting to 80 in the week. These are regular working hours available. Formulate the problem as:

(a) a linear programming model, and
(b) a goal programming model

to maximize the company's profit in that week.

Solution 7.1
Let \( x_1 \) = number of standard armchairs produced
\[ x_2 \] = number of deluxe armchairs produced
(a) Linear programming model:

Max \( 400x_1 + 500x_2 \)

subject to

\[
\begin{align*}
\text{man-hours, woodwork} & \\
\text{man-hours, painting} & \\
\text{market} & \\
\end{align*}
\]

\[
\begin{align*}
x_1 + 2x_2 & \leq 120 \\
x_1 + x_2 & \leq 80 \\
x_2 & \leq 50 \\
x_1, x_2 & \geq 0
\end{align*}
\]

The above model has a solution that

\[
\begin{align*}
x_1 &= 40 \\
x_2 &= 40 \\
\text{maximum profit} &= 36,000
\end{align*}
\]

(b) Goal programming model:

Instead of formulating a linear programming model as shown in (a) above, the problem can be formulated as a goal programming model. In order to do so, we must set a target for our profit goal (i.e. the only goal in this problem). This target is an arbitrary large figure, say, \$100,000 in this case. We can now define our goal in the form of a constraint as follows:

\[
400x_1 + 500x_2 + d^-_1 - d^+_1 = 100,000
\]

where

\[
\begin{align*}
d^-_1 &= \text{under-achievement of profit goal in } \$ \\
d^+_1 &= \text{over-achievement of profit goal in } \$
\end{align*}
\]

Our objective is to minimize the amount of under-achievement of the profit goal and we do not care about the over-achievement (see notes below). We have:

\[
\begin{align*}
\text{Min } d^-_1 \\
\text{subject to}
\end{align*}
\]

\[
\begin{align*}
400x_1 + 500x_2 + d^-_1 - d^+_1 &= 100,000 \\
x_1 + 2x_2 & \leq 120
\end{align*}
\]
The solution for this goal programming model is:

\[ \begin{align*}
& d_1^- = 64,000 \\
& x_1 = 40 \\
& x_2 = 40 \\
& d_1^+ = 0 
\end{align*} \]

Readers should compare this solution with the one in part (a).

**Notes**

Two important notes in formulating the objective function of a goal programming model are:

1. If over-achievement of a certain goal is acceptable, the positive deviation variable \((d_i^+)\) from the goal should not be included in the objective function. The above example is of such a case.

2. If under-achievement of a certain goal is acceptable, the negative deviation variable \((d_i^-)\) from the goal should not be included in the objective function. We will see this in later examples.

Examples 7.1 (b) has shown the formulation of a simple goal programming problem. However, the usual practice in goal programming is that we always write "=" signs in constraints instead of "≤" or "≥" signs. The model in example 7.1 (b) can therefore be written as:

Min \( d_i^- \)

subject to

\[ \begin{align*}
& 400x_1 + 500x_2 + d_1^- - d_1^+ = 100,000 \\
& x_1 + 2x_2 + d_2^- = 120 \\
& x_1 + x_2 + d_3^- = 80 \\
& x_2 + d_4^- = 50 
\end{align*} \]
where \( d_2^- \) = under-achievement of woodwork man-hours
\( d_3^- \) = under-achievement of painting man-hours
\( d_4^- \) = under-achievement of the sales of deluxe armchairs

The solution for this model is exactly the same as that for the previous model (i.e. the one without \( d_2^- \), \( d_3^- \) and \( d_4^- \)). It should be noted that \( d_2^+, d_3^+ \) and \( d_4^+ \) are not in the constraints because 120, 80 and 50 are maximum possible values and there cannot be over-achievements. These are called "absolute goals".

### 7.2 Multiple Goal Problems

In the previous examples, there was only one goal to be optimized. Now, let us see how multiple goals can be optimized.

**Example 7.2**

This example is a continuation of Example 7.1. It is possible to increase the number of man-hours available by working overtime. The management of the company, however, wishes that the overtime hours should be within 80 hours in the two departments as far as possible. Furthermore, the management establishes a list, in the order of decreasing priority, on goal attainment as follows:

Priority 1 total overtime hours should be within 80 as far as possible;
Priority 2 production of deluxe armchair should be limited to the sales forecast but this limited market should be as saturated as possible;
Priority 3 a target profit of $100,000 should be achieved as far as possible;
Priority 4 overtime hours should be avoided as far as possible.

Formulate this problem as a prioritized goal programming model.

**Solution 7.2**

**Priority 1 (1st goal)**: total overtime hours should be within 80 as far as possible.

So, we must add the following constraint to the set of constraints in Example 7.1:

\[
x_1, x_2, d_1^+, d_2^+, d_3^+, d_4^+ \geq 0
\]
\[ d_2^+ + d_3^+ + d_4^- - d_5^- = 80 \]  
\[ d_3^- = \text{over-achievement of woodwork man-hours (i.e. overtime for woodwork)} \]
\[ d_4^- = \text{over-achievement of painting man-hours (i.e. overtime for painting)} \]
\[ d_5^- = \text{under-achievement of the 1st goal} \]
\[ d_5^+ = \text{over-achievement of the 1st goal} \]

And constraints (2) and (3) will change to:
\[ x_1 + 2x_2 + d_2^+ - d_5^+ = 120 \]
\[ x_1 + x_2 + d_4^- - d_3^- = 80 \]

Note that these two are no more absolute goals because overtime is now possible.

Moreover, the 1st goal is to minimize \( d_3^+ \). We will put this as priority 1 in the objective function.

**Priority 2 (2nd goal):** production of deluxe armchair should be limited to the sales forecast but this limited market should be as saturated as possible.

For the 2nd goal, no additional constraint is necessary. However, since the 2nd goal is to minimize \( d_4^- \), we will include it as priority 2 in the objective function.

**Priority 3 (3rd goal):** achieve profit target of $100,000.

The 3rd goal also needs no additional constraint, but we have to include a priority 3 that \( d_5^- \) be minimized in the objective function.

**Priority 4 (4th goal):** avoid overtime.

Again, the 4th goal needs no additional constraint, but we have to minimize \( (d_2^+ + d_3^-) \) and include it as priority 4 in the objective function.
We can now summarize the goal programming model as follows:

\[
\text{Min } P_1d_5^+ + P_2d_4^- + P_3d_1^- + P_4(d_2^+ + d_3^-)
\]

subject to

\[
400x_1 + 500x_2 + d_1^- - d_1^+ = 100,000 \quad (1)
\]
\[
x_1 + 2x_2 + d_2^- - d_2^+ = 120 \quad (2)
\]
\[
x_1 + x_2 + d_3^- - d_3^+ = 80 \quad (3)
\]
\[
x_2 + d_4^- = 50 \quad (4)
\]
\[
d_2^+ + d_3^+ + d_5^- - d_5^+ = 80 \quad (5)
\]
\[
x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_5^-, d_5^+ \geq 0
\]

The solution for this model is:

\[
x_1 = 65
\]
\[
x_2 = 50
\]
\[
d_1^- = 49000
\]
\[
d_2^+ = 45
\]
\[
d_3^+ = 35
\]
other deviation variables = 0

In solving a prioritized goal programming model, the goal of the highest priority must firstly be optimized to the fullest possible extent. When no further improvement is possible in the highest goal, we then optimize the second highest goal, and so on. Goals of lower priorities must not be optimized at the expenses of goals of higher priorities. The detailed method for solving a prioritized goal programming model will be discussed in Chapter 8.

### 7.3 Additivity of Deviation Variables

When two goals are assigned the same priority, we must make sure that the units of measure of the goals are commensurable. In other words, goals can be assigned to a same priority level only if they can be expressed in terms of a common unit of measure like priority 4 in Example 7.2; if not, some conversion has to be devised so that the addition will make sense. This can be
accomplished by multiplying the priority coefficient by a relative weighting factor. The following example will illustrate this.

**Example 7.3**
This example is a continuation of Example 7.2. Assume that the number of priority levels is reduced from four to three such that the profit goal (originally goal 3) and the avoidance of overtime goal (originally goal 4) are both assigned a priority of 3. Moreover, suppose that the marginal cost of working one hour overtime in the woodwork department is $75 and that in the painting department is $65. Modify the model formulated in solution 7.2 to a new model according to the new conditions.

**Solution 7.3**
So, priorities 1 and 2 remain the same. Priority 3 now is to achieve a profit target of $100,000 and to avoid overtime hours. These two things are not of a common unit of measure, that is, dollars cannot be added to hours in a meaningful sense. So, some conversion is necessary before these items can be added.

Since the marginal cost of working one hour overtime in the woodwork department is $75 and that in the painting department is $65, one hour overtime in woodwork and that in painting are equivalent to $75 and $65 respectively in profits. The coefficients in the objective function for goal 3 will change to \((d_1^- + 75d_2^+ + 65d_3^+)\) because it is sensible to have profit added to profits. Therefore, the new goal programming model is:

\[
\begin{align*}
\text{Min } P_1d_1^+ + P_2d_2^- + P_3(d_1^- + 75d_2^+ + 65d_3^+) \\
\text{subject to } \\
400x_1 + 500x_2 + d_1^- - d_1^+ = 100,000 \\
x_1 + 2x_2 + d_2^- - d_2^+ = 120 \\
x_1 + x_2 + d_3^- - d_3^+ = 80 \\
x_2 + d_4^- = 50
\end{align*}
\]
\[ d_2^+ + d_3^+ + d_5^- - d_5^+ = 80 \]  \hspace{1cm} (5)

The solution for this model is:

\[ x_1 = 65 \]
\[ x_2 = 50 \]
\[ d_1^- = 49,000 \]
\[ d_2^+ = 45 \]
\[ d_3^+ = 35 \]
\[ \text{other deviation variables} = 0 \]

The additivity problem is not confined to different units of measure in a given priority. Sometimes, conversions have to be made even if deviation variables are of the same units of measure in the same priority. The next example (i.e. Example 7.4) will illustrate this and readers should note priority I of the example.

**Example 7.4**

A company produces two products I and II. The estimated sales for the next month are 24 units and 8 units of products I and II respectively. Each unit of products I and II contributes a profit of $4,000 and 5,000 respectively. The man-hours needed to produce each unit of products in two work centres A and B are given in the following table. The man-hours available of the two work centres are also given in the table.

<table>
<thead>
<tr>
<th>Work centre</th>
<th>Man-hours per unit</th>
<th>Man-hours available</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I</td>
<td>II</td>
</tr>
<tr>
<td>Work centre A</td>
<td>32</td>
<td>56</td>
</tr>
<tr>
<td>Work centre B</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

The management of the company has the following priority of goals:
P_1: limit the production to the sales forecast but the limited market should be as saturated as possible;

P_2: avoid the under-utilization of regular man-hour capacity to maintain stable employment;

P_3: limit overtime operation as far as possible.

Formulate the problem as a goal programming model for determining the optimal number of each product to be produced in the next month.

**Solution 7.4**

Let \( x_1 = \) number of product I produced for the next month

\( x_2 = \) number of product II produced for the next month

**Goal 1**: limit the production to the sales forecast but the limited market should be as saturated as possible.

Goal 1 is an absolute goal and can be expressed by the following two constraints:

\[
x_1 + d_1^- = 24 \quad \text{(1)}
\]

\[
x_2 + d_2^- = 8 \quad \text{(2)}
\]

where

\( d_1^- = \) under-achievement in monthly production of product I

\( d_2^- = \) under-achievement in monthly production of product II

The objective is to minimize \( P_3(4d_1^- + 5d_2^-) \). Weightages 4 and 5 are used because the management has a greater desire to minimize \( d_2^- \) than \( d_1^- \) since the profits for each unit of product I and that for product II are in the ratio of 4:5.

**Goal 2**: avoid the under-utilization of regular man-hour capacity.

Goal 2 can be expressed by the following two constraints:

\[
32x_1 + 56x_2 + d_3^- - d_3^+ = 768 \quad \text{(3)}
\]
\[ 10x_1 + 15x_2 + d_4^- - d_4^+ = 300 \]  \hspace{1cm} (4)

where

\[ d_3^- = \text{under-utilization of regular man-hour capacity of work centre A} \]
\[ d_3^+ = \text{man-hours of overtime in work centre A} \]
\[ d_4^- = \text{under-utilization of regular man-hour capacity of work centre B} \]
\[ d_4^+ = \text{man-hours of overtime in work centre B} \]

The objective is to minimize \( P_2(d_3^- + d_4^-) \).

**Goal 3**: limit overtime operation as far as possible.

Goal 3 can be expressed by the following two constraints:

\[ d_3^+ + d_5^- - d_5^+ = 376 \]  \hspace{1cm} (5)
\[ d_4^+ + d_6^- - d_6^+ = 80 \]  \hspace{1cm} (6)

where

\[ d_5^- = \text{under-utilization of overtime capacity in work centre A} \]
\[ d_5^+ = \text{over-utilization of overtime capacity in work centre A} \]
\[ d_6^- = \text{under-utilization of overtime capacity in work centre B} \]
\[ d_6^+ = \text{over-utilization of overtime capacity in work centre B} \]

The objective is to minimize \( P_3(d_5^+ + d_6^+) \).

The following is a summary of the goal programming model:

\[
\text{Min} \quad P_1(4d_1^- + 5d_2^-) + P_2(d_3^- + d_4^-) + P_3(d_5^+ + d_6^+) \]  \hspace{1cm} (0)

subject to

\[ x_1 + d_1^- = 24 \]  \hspace{1cm} (1)
\[ x_2 + d_2^- = 8 \]  \hspace{1cm} (2)
\[ 32x_1 + 56x_2 + d_3^- - d_3^+ = 768 \]  \hspace{1cm} (3)
\[ 10x_1 + 15x_2 + d_4^- - d_4^+ = 300 \]  \hspace{1cm} (4)
\[ d_3^+ + d_5^- - d_5^+ = 376 \]  \hspace{1cm} (5)
\[ d_4^+ + d_6^- - d_6^+ = 80 \]  \hspace{1cm} (6)

\[ x_1, x_2, d_1^-, d_2^-, d_3^-, d_4^-, d_5^+, d_6^+, d_5^-, d_6^- \geq 0 \]
The solution for this model is:

\[ x_1 = 24 \]
\[ x_2 = 8 \]
\[ d_1^+ = 448 \]
\[ d_4^+ = 60 \]
\[ d_5^+ = 72 \]
\[ d_6^+ = 20 \]
all other deviation variables = 0

### 7.4 Integer Goal Programming

As in linear programming models, goal programming may involve decision variables which are of integers or zero-one values. The next example will show the formulation of such kind of problems.

**Example 7.5**

A company’s management is considering undertaking three plans I, II and III. A plan must be either selected or rejected (i.e. partial adoption of a plan is not allowed). The plans are of two years duration and the budget allowed for each year appears to be insufficient to support all the three plans. The net profit, market share and the cost of each plan are given in the following table:

<table>
<thead>
<tr>
<th>Plan</th>
<th>Net profit (units)</th>
<th>Market share (units)</th>
<th>Cost in Year 1 (units)</th>
<th>Cost in Year 2 (units)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>II</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>III</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

The allowable total budget for year 1 is 10 units, while that for year 2 is 9 units. It is required that these budget levels should not be exceeded. The company’s management considers that profit maximization is most important. The second important consideration is that the market shares should be within the limits as
far as possible. Formulate the problem as an integer goal programming model for determining which plan(s) should be adopted.

**Solution 7.5**

Let

\[
\begin{align*}
    x_1 &= \begin{cases} 1 & \text{plan I is adopted} \\ 0 & \text{plan I is not adopted} \end{cases} \\
    x_2 &= \begin{cases} 1 & \text{plan II is adopted} \\ 0 & \text{plan II is not adopted} \end{cases} \\
    x_3 &= \begin{cases} 1 & \text{plan III is adopted} \\ 0 & \text{plan III is not adopted} \end{cases}
\end{align*}
\]

**Budget**

There are two absolute goals:

\[
\begin{align*}
    7x_1 + 2x_2 + 5x_3 + d_1^- &= 10 \quad \text{(1)} \\
    5x_1 + 3x_2 + 4x_3 + d_2^- &= 9 \quad \text{(2)}
\end{align*}
\]

where

- \(d_1^-\) = under-achievement of total budget in year 1
- \(d_2^-\) = under-achievement of total budget in year 2

**Profit**

The profit goal can be expressed as:

\[
8x_1 + 4x_2 + 9x_3 + d_3^- - d_3^+ = 21 \quad \text{(3)}
\]

where

- \(d_3^-\) = under-achievement of profit goal
- \(d_3^+\) = over-achievement of profit goal

The objective is to minimize \(P_1d_3^-\). The RHS value of 21 in constraint (3) is the sum of the net profits of the three plans.

**Market share**

The market share can be expressed as:

\[
5x_1 + 3x_2 + 3x_3 + d_4^- - d_4^+ = 11 \quad \text{(4)}
\]

where
\[ d_4^- = \text{under-achievement of market share} \]
\[ d_4^+ = \text{over-achievement of market share} \]

The objective is to minimize \( P_2d_4^+ \). The RHS value of 11 in constraint (4) is the sum of the market shares of the three plans.

The following is a summary of the integer goal programming model:

\[
\begin{align*}
\text{Min} & \quad P_1d_3^- + P_2d_4^+ & \quad (0) \\
\text{subject to} & \\
7x_1 + 2x_2 + 5x_3 + d_1^- &= 10 & \quad (1) \\
5x_1 + 3x_2 + 4x_3 + d_2^- &= 9 & \quad (2) \\
8x_1 + 4x_2 + 9x_3 + d_3^- - d_3^+ &= 21 & \quad (3) \\
5x_1 + 3x_2 + 3x_3 + d_4^- - d_4^+ &= 11 & \quad (4) \\
\end{align*}
\]

\[ x_1, x_2, x_3 \in \{0, 1\}; \quad d_1^-, d_2^-, d_3^-, d_3^+, d_4^-, d_4^+ \geq 0 \]

The solution for this model is:
\[ x_1 = 0 \]
\[ x_2 = 1 \]
\[ x_3 = 1 \]
\[ d_1^- = 3 \]
\[ d_2^- = 2 \]
\[ d_3^- = 8 \]
\[ d_4^- = 5 \]
all other deviation variables = 0

Exercise

1. A company runs a production line with 1000 man-hours of regular production time available per day. Overtime may be worked. Two products, I and II, are to be produced. Each product requires 2 man-hours of production time per item. Sales forecast reviews that a maximum of 250
items of product I and 350 items of product II can be sold in a day. The contribution of I is $20 per item and that of II is $15 per item.

The management has set three goals in order of priority:
P₁ : avoid under-utilization of regular production time;
P₂ : sell as many items as possible;
P₃ : reduce overtime.

Formulate a goal programming model to optimize the production of I and II using decision variables x₁ and x₂ as the number of products I and II respectively produced per day.

2. A factory manufactures two types of products I and II. Type I requires 2 man-hours in the assembly line while type II requires 4 man-hours. Marketing surveys predict that no more than 30 units of type I and 20 units of type II should be produced in the next week. The net profit from type I is $800 each and that from type II is $1,200 each. The regular-time assembly operation is limited to 96 man-hours in the next week.

The factory manager has the following goals arranged in the order of decreasing priority:
P₁ : the number of products should never exceed the predicted market demand;
P₂ : the total profit should be maximized;
P₃ : the overtime operation of the assembly line should be minimized;
P₄ : the limited market should be as saturated as possible (i.e. sell as many products as possible).

Formulate a goal programming model which will satisfy the factory manager’s goals and optimize the number of products produced in the next week.
8 GOAL PROGRAMMING SOLUTION

8.1 The Revised Simplex Method as a Tool for Solving Goal Programming Models

In this chapter we will see how a goal programming model is solved. The revised simplex method (Appendix B) will be used as the tool. The example given below will illustrate the solutioning process. The model formulated in Example 7.2 of the previous chapter is used.

Example 8.1

Solve the following goal programming model:

\[
\text{Min } P_1 d_5^- + P_2 d_4^- + P_3 d_1^- + P_4 d_2^- + P_5 d_3^- \tag{0}
\]

subject to:

\[
400x_1 + 500x_2 + d_1^- - d_1^+ = 100,000 \tag{1}
\]
\[
x_1 + 2x_2 + d_2^- - d_2^+ = 120 \tag{2}
\]
\[
x_1 + x_2 + d_3^- - d_3^+ = 80 \tag{3}
\]
\[
x_2 + d_4^- = 50 \tag{4}
\]
\[
d_2^+ + d_3^+ + d_5^- - d_5^+ = 80 \tag{5}
\]
\[
x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_5^-, d_5^+ \geq 0
\]

Solution 8.1

We will use the revised simplex method (Appendix B) to solve this model. The solutioning process will be discussed step by step.

Step 1

As there are five constraints, there will be five basic variables. The negative deviation variables, \(d_i^-\), are analogous to slack variables (for absolute goals) or artificial variables (for non-absolute goals) in linear programming models (see Appendix C). In other words, \(d_i^-\) always appear as the basic variables in the initial goal programming tableau. In this example, the initial tableau is shown in Fig. 8.1.
<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^{-}$</th>
<th>$d_1^{+}$</th>
<th>$d_2^{-}$</th>
<th>$d_2^{+}$</th>
<th>$d_3^{-}$</th>
<th>$d_3^{+}$</th>
<th>$d_4^{-}$</th>
<th>$d_4^{+}$</th>
<th>$d_5^{-}$</th>
<th>$d_5^{+}$</th>
<th>RHS</th>
<th>( a_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_1 )</td>
<td>( p_4 )</td>
<td>400</td>
<td>500</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100000</td>
<td></td>
</tr>
<tr>
<td>( d_2 )</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>( d_3 )</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>( d_4 )</td>
<td>( p_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>( d_5 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>( p_4 )</td>
<td>( p_1 )</td>
<td>( p_2 )</td>
<td>( p_1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8.1 Initial goal programming tableau (a)

In ordinary linear programming simplex tableau, we have $Z_j$ and $C_j - Z_j$ rows at the bottom (see Appendix B). In goal programming, however, we omit the $Z_j$ row in order to simplify the tableau. It requires a little more calculation in our heads. Moreover, instead of having one $C_j - Z_j$ row as in linear programming, we now have four rows $P_4$, $P_3$, $P_2$ and $P_1$ of $C_j - Z_j$ values arranged in increasing order of priority.

**Step 2**

We now calculate the $C_j - Z_j$ values at the bottom of the initial tableau.

$Z_j$ values, as explained in Appendix B, are products of the sum of $C_j$ times coefficient. Thus, $Z_j$ value in the $x_1$ column is $400P_3$. $C_j$ value in the $x_1$ column is 0. Therefore, $C_j - Z_j$ value for the $x_1$ column is $-400P_3$. Hence, we put -400 at the $P_3$ row in the $x_1$ column (see Fig. 8.2).

$Z_j$ value in the $x_2$ column is $500P_3 + P_2$. $C_j$ value in the $x_2$ column is 0. Therefore, $C_j - Z_j$ for the $x_2$ column is $-500P_3 - P_2$. Since $P_2$ and $P_3$ denote different priorities, we must list them separately in the $P_2$ and $P_3$ rows for $C_j - Z_j$. Consequently, $C_j - Z_j$ value is -1 at the $P_2$ row and -500 at the $P_3$ row in the $x_2$ column (see Fig. 8.2).
For column $d_1^-$, $Z_j$ value is $P_3$. $C_j$ value in the $d_1^-$ column is also $P_3$. Therefore, $C_j - Z_j$ is 0. Hence, we put 0 value for all the four rows under this column.

For column $d_1^+$, $Z_j$ value is $-P_3$. Since $C_j$ is 0, $C_j - Z_j$ is therefore $P_3$. So, we put 1 at the $P_3$ row in the $d_1^+$ column.

By the similar methodology, we can obtain $C_j - Z_j$ values of $P_1$, $P_2$, $P_3$ and $P_4$ rows for all the remaining columns.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>RHS $a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^-$</td>
<td>$P_3$</td>
<td>100000</td>
</tr>
<tr>
<td>$d_2^-$</td>
<td>0</td>
<td>120</td>
</tr>
<tr>
<td>$d_3^-$</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>$d_4^-$</td>
<td>$P_2$</td>
<td>50</td>
</tr>
<tr>
<td>$d_5^-$</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>$P_4$</td>
<td>0</td>
</tr>
<tr>
<td>$P_2$</td>
<td>-400</td>
<td>100000</td>
</tr>
<tr>
<td>$P_1$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 8.2 Initial goal programming tableau (b)

In the RHS column, we simply put down the $Z_j$ value and not the $C_j - Z_j$ value. In this column, $Z_j$ is $100000P_3 + 50P_2$. Hence, we put 50 at the $P_2$ row and 100000 at the $P_3$ row.

In goal programming, we must firstly achieve $P_1$ to the fullest possible extent and then consider $P_2$, and so forth. Therefore, in determining the entering variable at this stage, we must examine row $P_1$ first. Since there is no negative coefficient in this row, we move to row $P_2$. In this row, there is a negative coefficient under the $x_2$ column. $x_2$ is the entering variable as there is no positive coefficient in the row below $P_2$ (or more exactly row $P_1$ in this case). If there is a positive coefficient below $P_2$ under the $x_2$ column, then $x_2$ cannot be
chosen as the entering variable as it would mean $P_2$ is optimized at the expense of $P_1$. This is not allowed in goal programming.

Then, we calculate the RHS/$a_{ij}$ values and determine the leaving variable. Since 50 is the smallest positive RHS/$a_{ij}$ value, $d_4^-$ is the leaving variable.

**Step 3**

As $d_4^-$ is leaving and $x_2$ is entering, the basic variables now are $d_1^-, d_2^-, d_3^-, x_2$ and $d_5^-$. They must be expressed in terms of the non-basic variables $x_1$, $d_1^+$, $d_2^+$, $d_3^+$, $d_4^+$ and $d_5^+$. To do so, readers should revise the examples given in Appendices A and B. The second goal programming tableau is shown in Fig. 8.3.

![Fig. 8.3 Second goal programming tableau](image)

**Step 4**

In Fig. 8.3, there is no negative coefficient in rows $P_1$ and $P_2$. We therefore move to row $P_3$. The most negative coefficient is -400 and there is no positive coefficient below it (i.e. no positive coefficient in rows $P_1$ and $P_2$ under this column). So $x_1$ is the entering variable. Then we calculate the RHS/$a_{ij}$ values. The smallest positive RHS/$a_{ij}$ value is 20 and so the leaving variable is $d_2^-$. 
Then, we express the basic variables $d_{i}^{-}$, $x_{1}$, $d_{3}^{-}$, $x_{2}$ and $d_{5}^{-}$ in terms of the non-basic variables $d_{1}^{+}$, $d_{2}^{-}$, $d_{3}^{+}$, $d_{4}^{-}$ and $d_{5}^{+}$. The third goal programming tableau is shown in Fig. 8.4.

<table>
<thead>
<tr>
<th>Basic</th>
<th>Variable</th>
<th>$x_{1}$</th>
<th>$x_{2}$</th>
<th>$d_{1}^{-}$</th>
<th>$d_{1}^{+}$</th>
<th>$d_{2}^{-}$</th>
<th>$d_{2}^{+}$</th>
<th>$d_{3}^{-}$</th>
<th>$d_{3}^{+}$</th>
<th>$d_{4}^{-}$</th>
<th>$d_{4}^{+}$</th>
<th>$d_{5}^{-}$</th>
<th>$d_{5}^{+}$</th>
<th>RHS</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{i}^{+}$</td>
<td>$P_{3}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-400</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>0</td>
<td>0</td>
<td>67000</td>
<td>167.5</td>
<td></td>
</tr>
<tr>
<td>$x_{1}$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>-20</td>
</tr>
<tr>
<td>$d_{3}^{+}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>(10)</td>
<td></td>
</tr>
<tr>
<td>$x_{2}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$d_{5}^{-}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8.4 Third goal programming tableau

**Step 5**

By similar reasoning, $d_{3}^{-}$ is leaving and $d_{2}^{+}$ is entering. We can obtain the fourth goal programming tableau as shown in Fig. 8.5.

<table>
<thead>
<tr>
<th>Basic</th>
<th>Variable</th>
<th>$x_{1}$</th>
<th>$x_{2}$</th>
<th>$d_{1}^{-}$</th>
<th>$d_{1}^{+}$</th>
<th>$d_{2}^{-}$</th>
<th>$d_{2}^{+}$</th>
<th>$d_{3}^{-}$</th>
<th>$d_{3}^{+}$</th>
<th>$d_{4}^{-}$</th>
<th>$d_{4}^{+}$</th>
<th>$d_{5}^{-}$</th>
<th>$d_{5}^{+}$</th>
<th>RHS</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{i}^{+}$</td>
<td>$P_{3}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-400</td>
<td>400</td>
<td>-100</td>
<td>0</td>
<td>0</td>
<td>63000</td>
<td>157.5</td>
<td></td>
</tr>
<tr>
<td>$x_{1}$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>-30</td>
<td></td>
</tr>
<tr>
<td>$d_{2}^{+}$</td>
<td>$P_{4}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>-10</td>
<td></td>
</tr>
<tr>
<td>$x_{2}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$d_{5}^{+}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>70</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8.5 Fourth goal programming tableau
Step 6

This time, $d_5^-$ is leaving and $d_3^+$ is entering. The fifth goal programming tableau is shown in Fig. 8.6.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_1^+$</th>
<th>$d_2^-$</th>
<th>$d_2^+$</th>
<th>$d_3$</th>
<th>$d_3^+$</th>
<th>$d_4$</th>
<th>$d_4^+$</th>
<th>RHS $a_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^-$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-200</td>
<td>0</td>
<td>-200</td>
<td>0</td>
<td>100</td>
<td>-200</td>
<td>200</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>-2/2</td>
<td>-1/2</td>
</tr>
<tr>
<td>$d_1^+$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$d_3^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
<td>2/2</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

| RHS            | 0     | 0     | 0       | 0       | 0       | 0       | 0     | 0       | 1     | 0       | 0         | 49000     |

Fig. 8.6 Fifth (final) goal programming tableau

Fig. 8.6 is the final tableau. If we examine rows $P_1$ and $P_2$, there is no negative coefficient. Row $P_3$ has the most negative coefficient -200 in column $d_3^+$. However, under -200, we have a positive coefficient in row $P_1$. We therefore cannot further optimize $P_3$ at the expense of $P_1$. The next most negative coefficient in row $P_3$ is -100 (in column $d_4$). Under -100 we have a positive coefficient in row $P_2$ and so we cannot further optimize $P_3$ at the expense of $P_2$. Then, we move to row $P_4$. There is a negative coefficient in this row under column $d_5^-$. However, a positive coefficient is under it (in row $P_3$). Hence we cannot further optimize $P_4$. Consequently, the optimal (final) solution is achieved. The answer is as follows:

$$d_1^- = 49,000$$
$$x_1 = 65$$
$$d_2^- = 45$$
$$x_2 = 50$$
$$d_3^- = 35$$

other deviation variables (non-basic) = 0
8.2 A Further Example

Example 8.2

Solve the following goal programming model:

\[
\text{Min } P_1d_1^- + P_2d_4^+ + 3P_3d_2^- + P_3d_3^- + P_4d_2^+ + 4P_4d_3^+ \quad (0)
\]

subject to

\[
\begin{align*}
5x_1 + 2x_2 + (d_1^- - d_1^+) &= 6400 \quad (1) \\
x_1 + (d_1^- - d_2^+) &= 800 \quad (2) \\
x_2 + (d_3^- - d_3^+) &= 400 \quad (3) \\
(d_2^+ + d_4^- - d_4^+) &= 100 \quad (4)
\end{align*}
\]

\[x_1, x_2, d_1^-, d_1^+, d_2^-, d_2^+, d_3^-, d_3^+, d_4^-, d_4^+ \geq 0\]

Solution 8.2

The initial tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( d_1^- )</th>
<th>( d_1^+ )</th>
<th>( d_2^- )</th>
<th>( d_2^+ )</th>
<th>( d_3^- )</th>
<th>( d_3^+ )</th>
<th>( d_4^- )</th>
<th>( d_4^+ )</th>
<th>RHS</th>
<th>( a_{j} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_1^- )</td>
<td>( P_1 )</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6400</td>
<td>6400 = 1280</td>
</tr>
<tr>
<td>( d_2^- )</td>
<td>( 3P_3 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>800</td>
<td>800 = 800</td>
</tr>
<tr>
<td>( d_3^- )</td>
<td>( P_3 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>400 = \infty</td>
</tr>
<tr>
<td>( d_4^- )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>100</td>
<td>100 = \infty</td>
</tr>
</tbody>
</table>

| \( C_j - Z_j \) | \( P_4 \) | 0           | 0           | 0           | 0           | 0           | 1           | 0           | 4           | 0           | 0           | 0     | 0          |
|                | \( P_3 \) | -3          | -1          | 0           | 0           | 0           | 3           | 0           | 1           | 0           | 0           | 2800  |             |
|                | \( P_2 \) | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 1           | 0     |             |
|                | \( P_1 \) | -5          | -2          | 0           | 1           | 0           | 0           | 0           | 0           | 0           | 0           | 6400  |             |

Fig. 8.7 Initial tableau

In the initial tableau (Fig. 8.7), \( Z_j \) value for the \( x_1 \) column is \( 5P_1 + 3P_3 \). \( C_j \) value in the \( x_1 \) column is 0. Hence, \( C_j - Z_j \) value for this column is -5\( P_1 \) - 3\( P_3 \). So, we put -5 at the \( P_1 \) row and -3 at the \( P_3 \) row in this column.

\( Z_j \) value for the \( d_3^+ \) column is -\( P_3 \). \( C_j \) value in this column is \( 4P_4 \). Therefore, \( C_j - Z_j \) value is \( 4P_4 \) - (-\( P_3 \)) or \( 4P_4 + P_3 \). Hence, we put 1 at the \( P_3 \) row and 4 at the \( P_4 \) row in the \( d_3^+ \) column.
In this way, we can compute the $C_j - Z_j$ values for all rows $P_1$, $P_2$, $P_3$ and $P_4$ in all columns. $x_i$ is the entering variable because it corresponds to the most negative value -5 in row $P_1$. $d_j^-$ is the leaving variable because it corresponds to the smallest positive RHS/$a_{ij}$ value 800. The second tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$d_4$</th>
<th>$d_5$</th>
<th>$d_6$</th>
<th>$d_7$</th>
<th>$d_8$</th>
<th>RHS</th>
<th>RHS $a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2400</td>
<td>2400/5  = 480</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>800</td>
<td>800/1  = -800</td>
</tr>
<tr>
<td>$d_2^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>400</td>
<td>400/0  = -∞</td>
</tr>
<tr>
<td>$d_3^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>100</td>
<td>100/1  = 100</td>
</tr>
<tr>
<td>$C_1 - Z_1$</td>
<td>$P_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_3$</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_1$</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2400</td>
<td>-----------</td>
</tr>
</tbody>
</table>

Fig. 8.8 Second tableau

The third tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$d_4$</th>
<th>$d_5$</th>
<th>$d_6$</th>
<th>$d_7$</th>
<th>$d_8$</th>
<th>RHS</th>
<th>RHS $a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1900</td>
<td>380/5  = 760</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>900</td>
<td>-900/1  = -900</td>
</tr>
<tr>
<td>$d_2^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>400</td>
<td>-800/0  = -∞</td>
</tr>
<tr>
<td>$d_3^-$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>100</td>
<td>-100/1  = -100</td>
</tr>
<tr>
<td>$C_1 - Z_1$</td>
<td>$P_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_3$</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-----------</td>
</tr>
<tr>
<td>$P_1$</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1900</td>
<td>-----------</td>
</tr>
</tbody>
</table>

Fig. 8.9 Third tableau
The fourth tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^*$</th>
<th>$d_1^*$</th>
<th>$d_2^*$</th>
<th>$d_2^*$</th>
<th>$d_3^*$</th>
<th>$d_4^*$</th>
<th>RHS</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_4^*$</td>
<td>P₂</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>3P₃</td>
<td>P₄</td>
<td>P₃</td>
<td>4P₄</td>
<td>0</td>
<td>P₂</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>380</td>
<td>950</td>
</tr>
<tr>
<td>$x_1$</td>
<td>P₃</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$d_2^*$</td>
<td>P₄</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>1200</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>480</td>
<td>1200</td>
</tr>
</tbody>
</table>

Fig. 8.10 Fourth tableau

The fifth tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^*$</th>
<th>$d_1^*$</th>
<th>$d_2^*$</th>
<th>$d_2^*$</th>
<th>$d_3^*$</th>
<th>$d_4^*$</th>
<th>RHS</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_4^*$</td>
<td>P₂</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
<td>-1</td>
<td>0</td>
<td>$\frac{2}{5}$</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>3P₃</td>
<td>P₄</td>
<td>P₃</td>
<td>4P₄</td>
<td>0</td>
<td>P₂</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>220</td>
<td>550</td>
</tr>
<tr>
<td>$x_1$</td>
<td>P₃</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
<td>-1</td>
<td>0</td>
<td>$\frac{2}{5}$</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>P₄</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
<td>-1</td>
<td>1</td>
<td>$\frac{2}{5}$</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>320</td>
<td>800</td>
</tr>
<tr>
<td>$d_2^*$</td>
<td>P₄</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$\frac{1}{5}$</td>
<td>-1</td>
<td>1</td>
<td>$\frac{2}{5}$</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>320</td>
<td>800</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>320</td>
<td>800</td>
</tr>
</tbody>
</table>

Fig. 8.11 Fifth tableau
The sixth (final) tableau is:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^+$</th>
<th>$d_1^-$</th>
<th>$d_3^+$</th>
<th>$d_3^-$</th>
<th>$d_4^+$</th>
<th>$d_4^-$</th>
<th>RHS $a_{i, j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1^+$</td>
<td>4P_1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>550</td>
</tr>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>900</td>
</tr>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>950</td>
</tr>
<tr>
<td>$d_3^-$</td>
<td>P_4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>P_4</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2300</td>
</tr>
<tr>
<td>P_3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P_2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>P_1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 8.12 Sixth (final) tableau

The sixth tableau (Fig. 8.12) is the final tableau because there is no negative value in any row ($P_1$, $P_2$, $P_3$ or $P_4$) without having a positive value below it. The optimal solution is:

- $d_1^+ = 550$
- $x_1 = 900$
- $x_2 = 950$
- $d_2^- = 100$
- other deviation variables (non-basic) = 0

### 8.3 Solving Goal Programming Models Using Linear Programming Software Packages

Besides the technique described in the above two sections, we can also use a linear programming software package (such as QSB') to solve a goal programming model. The solutioning process is illustrated by the following example.
**Example 8.3**

Use a linear programming software package to solve the goal programming model of Example 8.2.

**Solution 8.3**

The goal programming model is:

\[\text{Min } P_1d_1^- + P_2d_4^+ + 3P_3d_2^- + P_3d_3^- + P_4d_2^+ + 4P_4d_3^+ \quad (0)\]

subject to

\[
\begin{align*}
5x_1 + 2x_2 + d_1^- - d_1^+ &= 6400 \quad (1) \\
x_1 + d_2^- - d_2^+ &= 800 \quad (2) \\
x_2 + d_3^- - d_3^+ &= 400 \quad (3) \\
d_2^+ + d_4^- - d_4^+ &= 100 \quad (4)
\end{align*}
\]

We divide the solutioning process into 4 steps.

**Step 1**

We optimize \(P_1\) first. The **linear programming** model in step 1 is:

\[\text{Min } d_1^- \quad (0)\]

subject to

\[
\begin{align*}
5x_1 + 2x_2 + d_1^- - d_1^+ &= 6400 \quad (1) \\
x_1 + d_2^- - d_2^+ &= 800 \quad (2) \\
x_2 + d_3^- - d_3^+ &= 400 \quad (3) \\
d_2^+ + d_4^- - d_4^+ &= 100 \quad (4)
\end{align*}
\]

The result of the objective function obtained by using a linear programming software to solve the model is:

\[d_1^- = 0\]

So, 0 is the minimum (optimal) value of \(d_1^-\). This is the highest priority and we cannot optimize other lower priorities by sacrificing the optimal value of \(d_1^-\). Hence, we add an additional constraint \((d_1^- = 0)\) to the linear programming model to fix the value of \(d_1^-\) and then optimize \(P_2\). This is illustrated in step 2.
Step 2

Then, we optimize P₂. The linear programming model in this step is:

\[
\begin{align*}
\text{Min} & \quad d_4^+ \\
\text{subject to} & \quad 5x_1 + 2x_2 + d_1^- - d_1^+ = 6400 \\
& \quad x_1 + d_2^- - d_2^+ = 800 \\
& \quad x_2 + d_3^- - d_3^+ = 400 \\
& \quad d_2^- + d_4^- - d_4^+ = 100 \\
& \quad d_i^+ = 0 \\
& \quad d_i^- = 0
\end{align*}
\]

(0) \hspace{1cm} (1) \hspace{1cm} (2) \hspace{1cm} (3) \hspace{1cm} (4) \hspace{1cm} (5)

The result is that the minimum value of the objective function is 0. Similarly, we now add an additional constraint \((d_4^- = 0)\) to the model and then optimize P₃.

Step 3

We now optimize P₃ by solving the following model:

\[
\begin{align*}
\text{Min} & \quad 3d_2^+ + d_3^+ \\
\text{subject to} & \quad 5x_1 + 2x_2 + d_1^- - d_1^+ = 6400 \\
& \quad x_1 + d_2^- - d_2^+ = 800 \\
& \quad x_2 + d_3^- - d_3^+ = 400 \\
& \quad d_2^- + d_4^- - d_4^+ = 100 \\
& \quad d_i^+ = 0 \\
& \quad d_i^- = 0
\end{align*}
\]

(0) \hspace{1cm} (1) \hspace{1cm} (2) \hspace{1cm} (3) \hspace{1cm} (4) \hspace{1cm} (5) \hspace{1cm} (6)

The optimal value of the objective function of this model is 0. So we further add a constraint \((3d_2^- + d_3^- = 0)\) to the model and then optimize P₄. This is illustrated in step 4.

Step 4

Lastly, we optimize P₄. The model is:
Goal Programming Solution

Min $d_2^* + 4d_3^*$  

subject to

\begin{align*}
5x_1 + 2x_2 + d_1^- - d_1^+ &= 6400 \\
x_1 + d_2^- - d_2^+ &= 800 \\
x_2 + d_3^- - d_3^+ &= 400 \\
d_2^+ + d_4^- - d_4^+ &= 100 \\
d_1^- &= 0 \\
d_4^+ &= 0 \\
3d_2^+ + d_3^- &= 0
\end{align*}

The solution of the above model is the optimal solution for our goal programming model. The result is:

\begin{align*}
d_2^+ + 4d_3^* &= 2300 \quad \text{(i.e. value of $P_4$)} \\
x_1 &= 900 \\
x_2 &= 950 \\
d_2^+ &= 100 \\
d_3^- &= 550
\end{align*}

all other deviation variables = 0

Readers can compare this result with the one (Fig. 8.12) obtained in Example 8.2.

**Exercise**

1. Solve, using goal programming tableaus, the following model:

Min $P_1d_1^- + P_2d_4^+ + 5P_3d_2^- + 3P_3d_2^- + P_4d_1^*$

subject to

\begin{align*}
x_1 + x_2 + d_1^- - d_1^+ &= 90 \\
x_1 + d_2^- &= 80 \\
x_2 + d_3^- &= 50 \\
d_1^+ + d_4^- - d_4^+ &= 10
\end{align*}

2. Use a linear programming software package to solve exercise no. 1.
Example 1: A Maximization Problem

Maximize

\[
Z = 10x_1 + 8x_2
\]  

subject to

\[
\begin{align*}
5x_1 + 3x_2 & \leq 750 \\
6x_1 + 4x_2 & \leq 800 \\
2x_1 + 3x_2 & \leq 480 \\
x_1, x_2 & \geq 0
\end{align*}
\]  

Step 1

Introduce slack variables \( S_1, S_2 \) and \( S_3 \) \((S_1, S_2, S_3 \geq 0)\) such that:

\[
\begin{align*}
Z - 10x_1 - 8x_2 &= 0 \\
5x_1 + 3x_2 + S_1 &= 750 \\
6x_1 + 4x_2 + S_2 &= 800 \\
2x_1 + 3x_2 + S_3 &= 480
\end{align*}
\]

A basic variable is the variable which can be expressed in terms of non-basic variables. The most convenient initial step is to let \( Z, S_1, S_2 \) and \( S_3 \) be the basic variables and \( x_1, x_2 \) be the non-basic variables because \( Z, S_1, S_2 \) and \( S_3 \) can easily be expressed in terms of \( x_1 \) and \( x_2 \):

\[
\begin{align*}
Z &= 10x_1 + 8x_2 \\
S_1 &= 750 - 5x_1 - 3x_2 \\
S_2 &= 800 - 6x_1 - 4x_2 \\
S_3 &= 480 - 2x_1 - 3x_2
\end{align*}
\]

Now, we can write equations \((0a), (1a), (2a)\) and \((3a)\) in the form of a simplex tableau as follows:
Basic Variable | Z | x₁ | x₂ | S₁ | S₂ | S₃ | RHS  
---|---|---|---|---|---|---|---
(0a) Z | 1 | -10 | -8 | 0 | 0 | 0 | 0  
(1a) S₁ | 0 | 5 | 3 | 1 | 0 | 0 | 750  
(2a) S₂ | 0 | 6 | 4 | 0 | 1 | 0 | 800  
(3a) S₃ | 0 | 2 | 3 | 0 | 0 | 1 | 480  

Initial simplex tableau (a)

Note that when non-basic variables are equal to 0, the basic variables are equal to RHS respectively.

**Step 2**

The next step is to find the *entering variable*, which is defined as the variable which corresponds to the most negative coefficient in row (0) of the simplex tableau. In our case, \( x₁ \) is the entering variable because -10 is most negative in row (0).

Now, we can add one more column to our simplex tableau called "RHS/\( a_{ij} \)" column. This column shows the quotients as a result of the division of the RHS figures by the coefficients in the column corresponding to the entering variable for each row except row (0).

Next, we have to determine the *leaving variable*, which is defined as the basic variable which corresponds to the smallest non-negative RHS/\( a_{ij} \) value. In our case, \( S₂ \) is the leaving variable because 133.3 is the smallest non-negative value in the last column.

**Step 3**

Then, the initial simplex tableau has to be transformed to a second tableau. Because \( x₁ \) is entering and \( S₂ \) is leaving, therefore the basic variables of the second tableau will...
be \( Z, S_1, x_1 \) and \( S_3 \). As such, \( x_2 \) and \( S_2 \) will be non-basic. This means that we have to express \( Z, S_1, x_1 \) and \( S_3 \) in terms of \( x_2 \) and \( S_2 \). In order to do this, we have to transform (0a), (1a), (2a) and (3a) into four linear equations (0b), (1b), (2b) and (3b) such that:

(0b) is an equation in \( Z \) expressed in terms of \( x_2 \) and \( S_2 \)
(1b) is an equation in \( S_1 \) expressed in terms of \( x_2 \) and \( S_2 \)
(2b) is an equation in \( x_1 \) expressed in terms of \( x_2 \) and \( S_2 \)
(3b) is an equation in \( S_3 \) expressed in terms of \( x_2 \) and \( S_2 \)

The above algebraic transformation can easily be done by transforming (2a) into (2b) and then substituting (2b) into (0a), (1a) and (3a) as follows:

\[
x_1 = \frac{800}{6} - \frac{4}{6} x_2 - \frac{1}{6} S_2 \quad \text{from (2a)}
\]

or
\[
x_1 = 133.33 - \frac{2}{3} x_2 - \frac{1}{6} S_2 \quad \text{(2b)}
\]

Substituting \( x_1 \) of (2b) into (0a), (1a) and (3a), we obtain:

\[
Z = 1333.3 + \frac{4}{3} x_2 - \frac{5}{3} S_2 \quad \text{(0b)}
\]
\[
S_1 = 83.33 + \frac{1}{3} x_2 + \frac{5}{6} S_2 \quad \text{(1b)}
\]
\[
S_3 = 213.33 - \frac{5}{3} x_2 + \frac{1}{3} S_2 \quad \text{(3b)}
\]

So, we can see that \( Z, x_1, S_1 \) and \( S_3 \) are expressed in terms of \( x_2 \) and \( S_2 \). Now, we write these four equations in a form suitable for entry into the simplex tableau:

\[
Z - \frac{4}{3} x_2 + \frac{5}{3} S_2 = 1333.3 \quad \text{(0b)}
\]
\[
-\frac{1}{3} x_2 + S_1 - \frac{5}{6} S_2 = 83.33 \quad \text{(1b)}
\]
\[
x_1 + \frac{2}{3} x_2 + \frac{1}{6} S_2 = 133.33 \quad \text{(2b)}
\]
\[
\frac{5}{3} x_2 - \frac{1}{3} S_2 + S_3 = 213.33 \quad \text{(3b)}
\]
Then, we enter equation (0b), (1b), (2b) and (3b) into the simplex tableau. This is the second tableau after the first iteration:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Z</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0b)</td>
<td>Z</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( \frac{1}{3} )</td>
<td>0</td>
<td>1333.3</td>
</tr>
<tr>
<td>(1b)</td>
<td>( S_1 )</td>
<td>0</td>
<td>0</td>
<td>( \frac{1}{3} )</td>
<td>1</td>
<td>( \frac{4}{6} )</td>
<td>0</td>
</tr>
<tr>
<td>(2b)</td>
<td>( x_1 )</td>
<td>0</td>
<td>1</td>
<td>( \frac{1}{3} )</td>
<td>0</td>
<td>( \frac{1}{6} )</td>
<td>0</td>
</tr>
<tr>
<td>(3b)</td>
<td>( S_3 )</td>
<td>0</td>
<td>0</td>
<td>( \frac{1}{3} )</td>
<td>0</td>
<td>1</td>
<td>213.33</td>
</tr>
</tbody>
</table>

Second simplex tableau

Step 4
Similarly, we know this time the entering variable is \( x_2 \) and the leaving variable is \( S_3 \). Hence, we continue with the second iteration, that is, to express the basic variables \( Z \), \( S_1 \), \( x_1 \) and \( x_2 \) in terms of the non-basic variables \( S_2 \) and \( S_3 \). This can be done by substituting (3b) into (0b), (1b) and (2b). (See step 3.) The result is:

\[
\begin{align*}
Z + 1.4S_2 + 0.8S_3 &= 1504 \\
S_1 - 0.9S_2 + 0.2S_3 &= 126 \\
x_1 + 0.3S_2 - 0.4S_3 &= 48 \\
x_2 - 0.2S_2 + 0.6S_3 &= 128
\end{align*}
\]

Then, we enter (0c), (1c), (2c) and (3c) into the third simplex tableau:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>Z</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0c)</td>
<td>Z</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.4</td>
<td>0.8</td>
</tr>
<tr>
<td>(1c)</td>
<td>( S_1 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>(2c)</td>
<td>( x_1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>-0.4</td>
</tr>
<tr>
<td>(3c)</td>
<td>( x_2 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Third simplex tableau (final tableau)

Step 5
When there is no negative coefficient in row (0), the optimal solution is achieved and no further iteration is necessary. When non-basic variables are equal to 0, the basic
variables are equal to the values of the RHS column. Therefore, the optimal solution is:

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>Non-basic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z = 1504$</td>
<td>$S_2 = 0$</td>
</tr>
<tr>
<td>$S_1 = 126$</td>
<td>$S_3 = 0$</td>
</tr>
<tr>
<td>$x_1 = 48$</td>
<td></td>
</tr>
<tr>
<td>$x_2 = 128$</td>
<td></td>
</tr>
</tbody>
</table>

**Example 2: A Minimization Problem**

Min $P = 750y_1 + 800y_2 + 480y_3$ .................................................. (0)

subject to

$5y_1 + 6y_2 + 2y_3 \geq 10$ .................................................. (1)

$3y_1 + 4y_2 + 3y_3 \geq 8$ .................................................. (2)

$y_1, y_2, y_3 \geq 0$

**Step 1**

Introduce slack variables $S_1$, $S_2$ and **artificial variables** $A_1$, $A_2$ ($S_1$, $S_2$, $A_1$, $A_2 \geq 0$) such that:

$5y_1 + 6y_2 + 2y_3 - S_1 + A_1 = 10$ .................................................. (1a)

$3y_1 + 4y_2 + 3y_3 - S_2 + A_2 = 8$ .................................................. (2a)

Introduce also a very large value $M$ as the coefficients of $A_1$ and $A_2$ which are added to the objective function such that:

$P = 750y_1 + 800y_2 + 480y_3 + MA_1 + MA_2$ .................................................. (0a)

**Step 2**

Before putting (0a), (1a) and (2a) into a simplex tableau, unlike the maximization problem, we have no convenient way but to eliminate $A_1$ and $A_2$ in equation (0a) by substituting (1a) and (2a) into (0a) as follows:
From (1a),  \[ A_1 = 10 - 5y_1 - 6y_2 - 2y_3 + S_1 \]  \[ \text{.......................... (1b)} \]
From (2a),  \[ A_2 = 8 - 3y_1 - 4y_2 - 3y_3 + S_2 \]  \[ \text{.......................... (2b)} \]
Substituting \( A_1 \) and \( A_2 \) into (0a), we have:

\[
P = 750y_1 + 800y_2 + 480y_3 + M(10 - 5y_1 - 6y_2 - 2y_3 + S_1) \\
+ M(8 - 3y_1 - 4y_2 - 3y_3 + S_2) = 0
\]

Simplifying, we obtain:

\[
P = 18M - (8M - 750)y_1 - (10M - 800)y_2 - (5M - 480)y_3 + MS_1 + MS_2
\]

\[ \text{.......................... (0b)} \]
It can be observed that in (0b), (1b) and (2b), \( P, A_1 \) and \( A_2 \) are expressed in terms of \( y_1, y_2, y_3, S_1 \) and \( S_2 \). Therefore, we can treat \( P, A_1 \) and \( A_2 \) as the basic variables. We now rewrite (0b), (1b) and (2b) in a form suitable for entry into a simplex tableau:

\[
P + (8M - 750)y_1 + (10M - 800)y_2 + (5M - 480)y_3 - MS_1 \\
- MS_2 = 18M
\]

\[ \text{.......................... (0b)} \]

\[
5y_1 + 6y_2 + 2y_3 - S_1 + A_1 = 10
\]
\[ \text{.......................... (1b)} \]

\[
3y_1 + 4y_2 + 3y_3 - S_2 + A_2 = 8
\]
\[ \text{.......................... (2b)} \]
We now put these equations into the initial simplex tableau:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( P )</th>
<th>( y_1 )</th>
<th>( y_2 )</th>
<th>( y_3 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>RHS ( a_1 )</th>
<th>RHS ( a_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0b)</td>
<td>( P )</td>
<td>1</td>
<td>8M-750</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(1b)</td>
<td>( A_1 )</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(2b)</td>
<td>( A_2 )</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Initial simplex tableau

**Step 3**
The *entering variable* this time (for minimization problem) is the variable which corresponds to the most positive coefficient in row (0). In this case, \( y_2 \) is the entering variable because \( (10M - 800) \) is most positive since \( M \) is a very large number. The
leaving variable is $A_1$ because it is the basic variable which corresponds to the smallest positive RHS/$a_{ij}$ value.

The next simplex tableau will therefore have $P$, $y_2$ and $A_2$ as basic variables which can be expressed in terms of non-basic variables $y_1$, $y_3$, $S_1$, $S_2$ and $A_1$. This can be achieved by algebraic transformation involving the substitution of (1b) into (0b) and (2b). (See step 3 of Example 1). The result is:

$$P = \frac{4M + 4000}{3}$$ \hfill (0c)

$$\frac{5}{6} y_1 + y_2 + \frac{1}{3} y_3 - \frac{1}{6} S_1 + \frac{1}{6} A_1 = \frac{5}{3} \quad \text{-----------------} \quad (1c)$$

$$-\frac{1}{3} y_1 + \frac{5}{3} y_3 + \frac{2}{3} S_1 - S_2 - \frac{2}{3} A_1 + A_2 = \frac{4}{3} \quad \text{-----------------} \quad (2c)$$

Then, we enter these three equations into the second simplex tableau:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$P$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>RHS</th>
<th>$a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0c) $P$</td>
<td>1</td>
<td>-26</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>(1c) $y_2$</td>
<td>0</td>
<td>$\frac{5}{3}$</td>
<td>1</td>
<td>$\frac{2}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>$\frac{5}{3}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(2c) $A_2$</td>
<td>0</td>
<td>$\frac{1}{3}$</td>
<td>0</td>
<td>$\frac{5}{3}$</td>
<td>$\frac{2}{3}$</td>
<td>$\frac{2}{3}$</td>
<td>1</td>
<td>$\frac{4}{3}$</td>
<td>0.8</td>
<td></td>
</tr>
</tbody>
</table>

Second simplex tableau after one iteration

**Step 4**

In the next iteration, $y_3$ will be the entering variable because it has the most positive coefficient in row(0) and $A_2$ the leaving variable. So, in the next tableau, $P$, $y_2$ and $y_3$ will be the basic variables. By substituting (2c) into (0c) and (1c), we obtain:

$$P = 126y_1 - 48S_1 - 128S_2 - (M - 48) A_1 - (M - 128) A_2 = 1504 \quad \text{-----------------} \quad (0d)$$

$$0.9y_1 + y_2 - 0.3S_1 + 0.2S_2 + 0.3A_1 - 0.2A_2 = 1.4 \quad \text{-----------------} \quad (1d)$$

$$-0.2y_1 + y_3 + 0.4S_1 - 0.6S_2 - 0.4A_1 + 0.6A_2 = 0.8 \quad \text{-----------------} \quad (2d)$$
We then enter (0d), (1d) and (2d) into the third simplex tableau:

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>P</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0d) P</td>
<td>1</td>
<td>-126</td>
<td>0</td>
<td>0</td>
<td>-48</td>
<td>-128</td>
<td>-(M-48)</td>
<td>-(M-128)</td>
<td>1504</td>
</tr>
<tr>
<td>(1d) $y_2$</td>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>-0.3</td>
<td>0.2</td>
<td>0.3</td>
<td>-0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(2d) $y_3$</td>
<td>0</td>
<td>-0.2</td>
<td>0</td>
<td>1</td>
<td>0.4</td>
<td>-0.6</td>
<td>-0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Third (final) simplex tableau

Step 5

When there is no positive coefficient in row (0), the optimal solution is reached and no further iteration is necessary. The optimal solution is:

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>Non-basic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = 1504</td>
<td>$y_1 = 0$</td>
</tr>
<tr>
<td>$y_2 = 1.4$</td>
<td>$S_1 = 0$</td>
</tr>
<tr>
<td>$y_3 = 0.8$</td>
<td>$S_2 = 0$</td>
</tr>
<tr>
<td></td>
<td>$A_1 = 0$</td>
</tr>
<tr>
<td></td>
<td>$A_2 = 0$</td>
</tr>
</tbody>
</table>
Appendix B
EXAMPLES ON REVISED SIMPLEX METHOD

Before studying this appendix, the reader must have already understood Appendix A, i.e. the conventional simplex method. Those procedures already discussed in the conventional method but also needed in the revised simplex method will not be repeated here. Therefore, Appendix B must be read in conjunction with Appendix A.

Example 1: A Maximization Problem
Max \( Z = 10x_1 + 8x_2 \) \quad \quad \quad \quad (0)
subject to
\[ 5x_1 + 3x_2 \leq 750 \] \quad \quad \quad \quad (1)
\[ 6x_1 + 4x_2 \leq 800 \] \quad \quad \quad \quad (2)
\[ 2x_1 + 3x_2 \leq 480 \] \quad \quad \quad \quad (3)
\[ x_1, x_2 \geq 0 \]

Step 1:
Introduce slack variables \( S_1, S_2 \) and \( S_3 \) \((S_1, S_2, S_3 \geq 0)\) such that:
\[ 5x_1 + 3x_2 + S_1 = 750 \] \quad \quad \quad \quad (1a)
\[ 6x_1 + 4x_2 + S_2 = 800 \] \quad \quad \quad \quad (2a)
\[ 2x_1 + 3x_2 + S_3 = 480 \] \quad \quad \quad \quad (3a)

The revised simplex tableau is a little more complicated than the conventional tableau. The following is the initial revised simplex tableau with equations (1a), (2a) and (3a) entered in it:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>RHS</th>
<th>( a_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) ( S_1 )</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>750</td>
<td></td>
</tr>
<tr>
<td>(2a) ( S_2 )</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>(3a) ( S_3 )</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>480</td>
<td></td>
</tr>
</tbody>
</table>

Initial revised simplex tableau (a)
Step 2

Then, we fill in the $C_j$ row. The $C_j$ values are the coefficients in the objective function. Next, we fill in the $C_j$ column. These are the $C_j$ associated with the basic variables only ($S_1$, $S_2$ and $S_3$ in this case). Now the tableau looks like the following:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
<th>RHS $rac{a_{ij}}{}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) $S_1$</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>750</td>
<td></td>
</tr>
<tr>
<td>(2a) $S_2$</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>800</td>
<td></td>
</tr>
<tr>
<td>(3a) $S_3$</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>480</td>
<td></td>
</tr>
<tr>
<td>$Z_j$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial revised simplex tableau (b)

Next, we have to fill the $Z_j$ row. To do it we multiply each of the $a_{ij}$ values by the $C_j$ of the $i_{th}$ row and add the products for each column. For example:

Value of $Z_j$ under column $x_1$

$= (C_j$ of row 1) $(a_{11}) + (C_j$ of row 2) $(a_{21}) + (C_j$ of row 3) $(a_{31})$

$= 0 \times 5 + 0 \times 6 + 0 \times 2$

$= 0$

Value of $Z_j$ under column $x_2$

$= (C_j$ of row 1) $(a_{12}) + (C_j$ of row 2) $(a_{22}) + (C_j$ of row 3) $(a_{32})$

$= 0 \times 3 + 0 \times 4 + 0 \times 3$

$= 0$

Value of $Z_j$ under the RHS column

$= 0 \times 750 + 0 \times 800 + 0 \times 480$

$= 0$

After $C_j$ row is filled, we then fill the $C_j - Z_j$ row. Its value is found by subtracting $Z_j$ from the corresponding $C_j$ at the top of the tableau for each column. For example:

Value of $C_j - Z_j$ under column $x_1 = 10 - 0 = 10$
Value of $C_j - Z_j$ under column $x_2 = 8 - 0 = 8$

The tableau now looks:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
<th>RHS $a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) $S_1$</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>(2a) $S_2$</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>(3a) $S_3$</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>480</td>
</tr>
<tr>
<td>$Z_j$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Initial revised simplex tableau (c)

Step 3

The entering variable is the variable which corresponds to the most positive value in the $C_j - Z_j$ row. 10 is the most positive in our case and therefore $x_1$ is the entering variable. The leaving variable is the basic variable which corresponds to the smallest non-negative $RHS/a_{ij}$ value. In our case, $S_2$ is the leaving variable because 133.3 is the smallest non-negative $RHS/a_{ij}$ value.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
<th>RHS $a_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) $S_1$</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>750</td>
</tr>
<tr>
<td>(2a) $S_2$</td>
<td>0</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>(3a) $S_3$</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>480</td>
</tr>
<tr>
<td>$Z_j$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Initial revised simplex tableau (d)

Step 4

Then, the initial tableau has to be transformed to a second tableau. In the second tableau, $S_1$, $x_1$ and $S_2$ will be the basic variables. The transformation (or called iteration) is done by substituting (2a) into (1a) and (3a). (For details see step 3 of Example 1 of Appendix A.) Now we have:

$$ -\frac{1}{3}x_2 + S_1 - \frac{5}{6}S_2 = 83.33 \quad \text{(1b)} $$
Then, we establish the second revised simplex tableau and compute all the \( Z_j \), \( C_j - Z_j \) and RHS/\( a_{ij} \) values.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>RHS</th>
<th>RHS/( a_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1b) ( S_1 )</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>83.33</td>
<td>-250</td>
</tr>
<tr>
<td>(2b) ( x_1 )</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>133.33</td>
<td>200</td>
</tr>
<tr>
<td>(3b) ( S_3 )</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>213.33</td>
<td>128</td>
</tr>
</tbody>
</table>

| \( Z_j \)   | 10 | \( \frac{20}{3} \) | 0 | \( \frac{5}{3} \) | 0 | 1333.3 |
| \( C_j - Z_j \) | 0 | \( \frac{4}{3} \) | 0 | \( \frac{5}{3} \) | 0 |

Second revised simplex tableau

**Step 5**

In the second revised simplex tableau, the entering variable is \( x_2 \) because it corresponds to the most positive value in the \( C_j - Z_j \) row. The leaving variable is \( S_3 \) because it corresponds to the smallest non-negative RHS/\( a_{ij} \) value. In the second iteration, we substitute (3b) into (1b) and (2b) and obtain the following equations:

\[
\begin{align*}
S_1 - 0.9S_2 + 0.2S_3 &= 126 \quad \text{(1c)} \\
x_1 + 0.3S_2 - 0.4S_3 &= 48 \quad \text{(2c)} \\
x_2 - 0.2S_2 + 0.6S_3 &= 128 \quad \text{(3c)}
\end{align*}
\]

Then, we enter (1c), (2c) and (3c) into the third revised simplex tableau and compute all the \( Z_j \) and \( C_j - Z_j \) values:
<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>$C_j$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1c) $S_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-0.9</td>
<td>0.2</td>
<td>126</td>
</tr>
<tr>
<td>(2c) $x_1$</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>-0.4</td>
<td>48</td>
</tr>
<tr>
<td>(3c) $x_2$</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-0.2</td>
<td>0.6</td>
<td>128</td>
</tr>
<tr>
<td>$Z_j$</td>
<td>10</td>
<td>8</td>
<td>0</td>
<td>1.4</td>
<td>0.8</td>
<td>1504</td>
<td></td>
</tr>
<tr>
<td>$C_j - Z_j$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1.4</td>
<td>-0.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Third (final) revised simplex tableau

**Step 6**

When there is no positive values in the $C_j - Z_j$ row, the optimal solution is achieved. The basic variables are equal to their corresponding RHS values while all non-basic variables are equal to 0. Therefore, the optimal solution is:

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>Non-basic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z = 1504$</td>
<td>$S_2 = 0$</td>
</tr>
<tr>
<td>$S_1 = 126$</td>
<td>$S_3 = 0$</td>
</tr>
<tr>
<td>$x_1 = 48$</td>
<td></td>
</tr>
<tr>
<td>$x_2 = 128$</td>
<td></td>
</tr>
</tbody>
</table>

**Example 2 : A Minimization Problem**

Min $P = 750y_1 + 800y_2 + 480y_3$  \( \leq 0 \)

subject to

- $5y_1 + 6y_2 + 2y_3 \geq 10$  \( \leq 0 \)
- $3y_1 + 4y_2 + 3y_3 \geq 8$  \( \leq 0 \)
- $y_1, y_2, y_3 \geq 0$

**Step 1**

Introduce slack variables $S_1, S_2$ and artificial variables $A_1, A_2$ ($S_1, S_2, A_1, A_2 \geq 0$) such that:
5y₁ + 6y₂ + 2y₃ - S₁ + A₁ = 10
3y₁ + 4y₂ + 3y₃ - S₂ + A₂ = 8

Introduce also a very large value M as the coefficients of A₁ and A₂ which are added to the objective function such that:

\[ P = 750y₁ + 800y₂ + 480y₃ + MA₁ + MA₂ \]

**Step 2**

Establish the initial revised simplex tableau as follows:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>C_j</th>
<th>y₁</th>
<th>y₂</th>
<th>y₃</th>
<th>S₁</th>
<th>S₂</th>
<th>A₁</th>
<th>A₂</th>
<th>RHS aij</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1a) A₁</td>
<td>M</td>
<td>750</td>
<td>800</td>
<td>480</td>
<td>0</td>
<td>0</td>
<td>M</td>
<td>M</td>
<td>10</td>
</tr>
<tr>
<td>(2a) A₂</td>
<td>M</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Z_j</th>
<th>8M</th>
<th>10M</th>
<th>5M</th>
<th>-M</th>
<th>-M</th>
<th>M</th>
<th>M</th>
<th>M</th>
<th>18M</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₁ - Zₙ</td>
<td>750</td>
<td>800</td>
<td>480</td>
<td>M</td>
<td>M</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-8M</td>
<td>-10M</td>
<td>-5M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial revised simplex tableau

For minimization problems, the entering variable is the variable which corresponds to the most negative value in the C_j - Z_j row. Therefore, y₂ is the entering variable. The leaving variable is the basic variable which corresponds to the smallest non-negative RHS/aᵢj value. Therefore, A₁ is the leaving variable.

**Step 3**

Then, the initial tableau has to be transformed to a second tableau in which y₂ and A₂ will be the basic variables. Such an iteration can be done by substituting (1a) into (2a). By doing so, we obtain:

\[ \frac{5}{6} y₁ + \frac{1}{3} y₃ - \frac{1}{6} S₁ + \frac{1}{6} A₁ = \frac{5}{3} \]

\[ -\frac{1}{3} y₁ + \frac{5}{3} y₃ + \frac{2}{3} S₁ - \frac{2}{3} S₂ - \frac{2}{3} A₁ + A₂ = \frac{4}{3} \]
So, we can now establish the second tableau:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>C_i</th>
<th>y_1</th>
<th>y_2</th>
<th>y_3</th>
<th>S_1</th>
<th>S_2</th>
<th>A_1</th>
<th>A_2</th>
<th>RHS</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1b) y_2</td>
<td>800</td>
<td>2/6</td>
<td>1/3</td>
<td>1/3</td>
<td>-1/6</td>
<td>0</td>
<td>0</td>
<td>M</td>
<td>1/5</td>
<td>5</td>
</tr>
<tr>
<td>(2b) A_2</td>
<td>M</td>
<td>3/3</td>
<td>2/3</td>
<td>2/3</td>
<td>-2/3</td>
<td>0</td>
<td>0</td>
<td>M</td>
<td>4/3</td>
<td>4</td>
</tr>
<tr>
<td>Z_j</td>
<td>2000 M/3</td>
<td>800</td>
<td>800+5M/3</td>
<td>-400+2M/3</td>
<td>-M</td>
<td>400-2M/3</td>
<td>M</td>
<td>4000+4M/3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>C_j - Z_j</td>
<td>750 + M*2000/3</td>
<td>0</td>
<td>480</td>
<td>400-2M/3</td>
<td>M</td>
<td>5M-400/3</td>
<td>0</td>
<td>128</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Second revised simplex tableau

Since M is a very large number, y_3 corresponds to the most negative C_j - Z_j value, and hence it is the entering variable. A_2 is the basic variable which corresponds to the smallest non-negative RHS/a_{ij} value, and so it is the leaving variable.

**Step 4**

The third tableau will have y_2 and y_3 as basic variables. Therefore, in this iteration, express y_2 and y_3 in terms of non-basic variables y_1, S_1, S_2, A_1 and A_2. This is accomplished by substituting (2b) into (1b). After doing so, we obtain:

\[
0.9y_1 + y_2 - 0.3S_1 + 0.2S_2 + 0.3A_1 - 0.2A_2 = 1.4 \quad \text{(1c)}
\]

\[
-0.2y_1 + y_3 + 0.4S_1 - 0.6S_2 - 0.4A_1 + 0.6A_2 = 0.8 \quad \text{(2c)}
\]

So, we can now establish the third tableau:

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>C_i</th>
<th>y_1</th>
<th>y_2</th>
<th>y_3</th>
<th>S_1</th>
<th>S_2</th>
<th>A_1</th>
<th>A_2</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1c) y_2</td>
<td>800</td>
<td>0.9</td>
<td>1</td>
<td>0</td>
<td>-0.3</td>
<td>0.2</td>
<td>0.3</td>
<td>-0.2</td>
<td>1.4</td>
</tr>
<tr>
<td>(2c) y_3</td>
<td>480</td>
<td>-0.2</td>
<td>0</td>
<td>1</td>
<td>0.4</td>
<td>-0.6</td>
<td>-0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>Z_j</td>
<td>624</td>
<td>800</td>
<td>480</td>
<td>-48</td>
<td>-128</td>
<td>48</td>
<td>128</td>
<td></td>
<td>1504</td>
</tr>
<tr>
<td>C_j - Z_j</td>
<td>126</td>
<td>0</td>
<td>0</td>
<td>48</td>
<td>128</td>
<td>M-48</td>
<td>M-128</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Third (final) revised simplex tableau
Step 5
When there is no negative values in the $C_j - Z_j$ row, the optimal solution is reached. The basic variables are equal to their corresponding RHS values while all non-basic variables are equal to 0. Therefore, the optimal solution is:

<table>
<thead>
<tr>
<th>Basic variable</th>
<th>Non-basic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P = 1504$</td>
<td>$y_1 = 0$</td>
</tr>
<tr>
<td>$y_2 = 1.4$</td>
<td>$S_1 = 0$</td>
</tr>
<tr>
<td>$y_3 = 0.8$</td>
<td>$S_2 = 0$</td>
</tr>
<tr>
<td></td>
<td>$A_1 = 0$</td>
</tr>
<tr>
<td></td>
<td>$A_2 = 0$</td>
</tr>
</tbody>
</table>
Appendix C
USE OF SLACK VARIABLES, ARTIFICIAL VARIABLES AND BIG-M

Note 1: If the constraint is of \( \leq \) sign, a slack variable \( S \) will be used.

Note 2: If the constraint is of \( = \) sign, an artificial variable \( A \) will be used.

Note 3: If the constraint is of \( \geq \) sign, a slack variable \( S \) and an artificial variable \( A \) will be used together.

Note 4: When an artificial variable \( A \) is used in a constraint, the objective function must be added a -MA term for a maximization problem or a +MA term for a minimization problem.

Example 1

Maximize \( Z = c_1x_1 + c_2x_2 \) \hspace{1cm} (0)
subject to
\[ a_{11}x_1 + a_{12}x_2 \leq b_1 \] \hspace{1cm} (1)
\[ a_{21}x_1 + a_{22}x_2 = b_2 \] \hspace{1cm} (2)
\[ a_{31}x_1 + a_{32}x_2 \geq b_3 \] \hspace{1cm} (3)
\[ x_1, x_2 \geq 0 \]

The model should be transformed to the following initial set of equations:

\[ Z = c_1x_1 + c_2x_2 - MA_1 - MA_2 \] \hspace{1cm} (0a)
\[ a_{11}x_1 + a_{12}x_2 + S_1 = b_1 \] \hspace{1cm} (1a)
\[ a_{21}x_1 + a_{22}x_2 + A_1 = b_2 \] \hspace{1cm} (2a)
\[ a_{31}x_1 + a_{32}x_2 - S_2 + A_2 = b_3 \] \hspace{1cm} (3a)

The basic variables in the initial simplex tableau will be \( S_1, A_1 \) and \( A_2 \).

Example 2

Minimize \( P = c_1x_1 + c_2x_2 \) \hspace{1cm} (0)
subject to
\[ a_{11}x_1 + a_{12}x_2 \leq b_1 \] \hspace{1cm} (1)
\[ a_{21}x_1 + a_{22}x_2 \geq b_2 \] \hspace{1cm} (2)
\[ a_{31}x_1 + a_{32}x_2 = b_3 \] \hspace{1cm} (3)
\[ x_1, x_2 \geq 0 \]

The model should be transformed to the following initial set of equations:

\[ P = c_1x_1 + c_2x_2 + MA_1 + MA_2 \] \hspace{1cm} (0a)
\[ a_{11}x_1 + a_{12}x_2 + S_1 = b_1 \] \hspace{1cm} (1a)
\[ a_{21}x_1 + a_{22}x_2 - S_2 + A_1 = b_2 \] \hspace{1cm} (2a)
\[ a_{31}x_1 + a_{32}x_2 + A_2 = b_3 \] \hspace{1cm} (3a)

The basic variables in the initial simplex tableau will be \( S_1, A_1 \) and \( A \).
Appendix D
EXAMPLES OF SPECIAL CASES

Example 1  No feasible solution

Maximize \( Z = 5x_1 + 6x_2 \)
subject to
\( x_1 + x_2 \geq 7 \)  \( \) (0)
\( x_1 \leq 4 \)  \( \) (1)
\( x_2 \leq 2 \)  \( \) (2)
\( x_1, x_2 \geq 0 \)  \( \) (3)

If the graphical method is used to solve this problem, one will find that no feasible space can be drawn (see Section 1.2.1 of Chapter 1). There is no feasible space that can satisfy all the three constraints. So, this problem has no feasible solution. If the simplex method is used to solve the problem, the initial and “final” tableaus are shown in Figures D1 and D2 respectively.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>( A_1 )</th>
<th>RHS</th>
<th>( )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 )</td>
<td>-M</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( Z_j )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-M</td>
</tr>
<tr>
<td>( C_j - Z_j )</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. D1  Initial simplex tableau for Example 1

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( S_3 )</th>
<th>( A_1 )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 )</td>
<td>-M</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>( Z_j )</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>-M</td>
<td>32-3M</td>
<td></td>
</tr>
<tr>
<td>( C_j - Z_j )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-5</td>
<td>-6</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Fig. D2  “Final” simplex tableau for Example 1

The “final” tableau in Figure D2 is supposed to be the “final” or “optimal” solution because the \( (C_j - Z_j) \) row has no positive value. However, it can be seen that \( A_1 \) remains to be a basic variable. This means that the constraint \( x_1 + x_2 \geq 7 \) is not satisfied, as a feasible solution should not incur the M penalties. Therefore, when an artificial variable stays as basic and cannot leave to become non-basic, it indicates that there is no feasible solution for the problem.
Example 2  Unbounded situation

Maximize \[ Z = 8x_1 + 5x_2 \]
subject to
\[ x_1 \leq 4 \]  \hspace{1cm} (1)
\[ x_1 + x_2 \geq 3 \]  \hspace{1cm} (2)
\[ x_1, x_2 \geq 0 \]

If the graphical method is used and a graph is drawn, one will see that the feasible space continues without limit in the \( x_2 \) direction. The objective function therefore can move without limit away from the origin. In this situation we say that the optimal solution is unbounded. If the simplex method is used to solve the problem, the initial and "final" tableaus are shown in Figures D3 and D4.

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( A_2 )</th>
<th>RHS</th>
<th>RHS ( a_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>( A_2 )</td>
<td>-M</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>( Z_j )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>( C_j - Z_j )</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-M</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. D3  Initial simplex tableau for Example 2

<table>
<thead>
<tr>
<th>Basic Variable</th>
<th>( C_j )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( S_1 )</th>
<th>( S_2 )</th>
<th>( A_2 )</th>
<th>RHS</th>
<th>RHS ( a_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_2 )</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>( \infty )</td>
</tr>
<tr>
<td>( Z_j )</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>-M</td>
<td>4</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>( C_j - Z_j )</td>
<td>0</td>
<td>5</td>
<td>-8</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

Fig. D4  "Final" simplex tableau for Example 2

The "final" tableau shown in Fig. D4 shows that there is still a positive value in the \( (C_j - Z_j) \) row, but in the RHS/\( a_{ij} \) column there is no finite positive value. It means that there is no leaving variable, and the simplex method cannot continue. This indicates an unbounded situation.

Example 3  Degeneracy

Maximize \[ Z = 10x_1 + 8x_2 \]
subject to
\[ 5x_1 + 3x_2 \leq 624 \]  \hspace{1cm} (1)
\[ 6x_1 + 4x_2 \leq 800 \]  \hspace{1cm} (2)
\[ 2x_1 + 3x_2 \leq 480 \]  \hspace{1cm} (3)
\[ x_1, x_2 \geq 0 \]
This example is a modification of Example 1.1 of Chapter 1. For the above model, all the three resources (storage space, raw material and working hours) are fully utilized at the optimal solution. For such a case, in the final or optimal simplex tableau, there is a basic variable with a zero value (see Figure D5). This corresponds geometrically to the fact that in Figure 1.1 of Chapter 1 three lines are passing through Point A rather than usually two. The phenomenon is known as degeneracy.

Example 4  More than one optimal solution

\[
\begin{align*}
\text{Max } Z & = 2x_1 + 4x_2 \\
\text{subject to} & \\
x_1 + 2x_2 & \leq 4 \\
x_1 & \leq 3 \\
x_2 & \leq 1 \\
x_1, x_2 & \geq 0
\end{align*}
\]

In the above example, when the objective function line moves as far from the origin as possible, it lies along (or it is parallel to) one of the lines bounding the feasible space. The two corner points \((x_1 = 2, x_2 = 1)\) and \((x_1 = 3, x_2 = 0.5)\) are optimal solutions, having the same objective function value of 8. If the simplex method is used to solve the problem, in the final (optimal) simplex tableau, there is a zero value in the \((C_j - Z_j)\) row for a non-basic variable (see Figure D6, \(S_3\) in this case). This is analogous to the zero value of an improvement index calculated from a transportation tableau (see Chapter 4).