Chapter 17: Iterative Improvement and Linear Programming

DR. S. SRIDHAR
ASSOCIATE PROFESSOR
DEPARTMENT OF IST
ANNA UNIVERSITY
objectives

- Basics of linear programming
- Basics of linear programming problem formulation
- Understanding of simplex method and its variants
- Understanding of the principle of duality
- Basics of max-flow and the Ford–Fulkerson algorithm
- Introduction to matching and optimal bipartite matching
- Fundamentals of stable marriage problem
Iterative Improvement

Fig. 17.1 Iterative improvement
What is LPP?

Linear programming is a mathematical technique for solving optimization problems whose objective function and constraints can be represented using linear equations and inequalities.
Advantages and Limitations of Linear Programming

The main advantages of linear programming are as follows:

1. It is a flexible and scientific tool that can solve optimization problems effectively.
2. Linear programming can help us evaluate the alternatives and is highly suitable for problems that are neither too small (i.e., having smaller instances) or very large (having very large instances).

Some of the limitations of linear programming are as follows:

1. The objective function and all the constraints should be expressed as a linear function.
2. A linear programme can have only one objective function, whereas most of the applications have multiple objectives.
Decision variables  Decision variables or variables are unknown quantities that are required to be determined in order to solve the given problem. These variables are called decision variables, as the problem is to find the values that these variables can take to solve the problem. The objective function and constraints of a given optimization problem should be expressed as linear functions of decision variables. For example, an expected number of items of a given problem can be modelled as a variable $X$. A variable can have a fractional value (called a continuous variable) or only binary values. A variable that has only binary values are called binary variables. Mostly in linear programming, only continuous variables are encountered.

Constraints  Constraints are the conditions or restrictions imposed on problems. These are modelled as linear equations. For example, the condition $x > 0$ is a constraint, as the variable $x$ is forced to take values greater than zero only.
Formulation of LPP

**Inequality**  Inequality means usage of operators ‘≤’ or ‘≥’ to model linear equations. For example, a constraint can be $x_1 \geq 1$. Here, the constraint is using the inequality ≥. Inequality provides more flexibility in linear programming.

**Objective function**  An objective function represents the goal of a given problem. Often in an optimization problem, the objective function is to maximize or minimize a factor such as profit, effort, or time. In linear programming, the objective function is also modelled as a linear function and is often determined based on the requirements of the problem.
Procedure for LPP

1. Identify decision variables.
2. Construct an objective function of a given problem using these decision variables. Express it in the form of a linear function such as $Z = C_1X_1 + C_2X_2 + \cdots + C_nX_n$, which can be maximized or minimized. In linear programming, optimization problems are always associated with maximization or minimization of the objective function of the given problem.
3. Identify all the constraints and express them in the form of a linear equation, or express inequalities in the form of decision variables. Constraints are the possible values that a variable of a linear programming problem can take.
4. Decision variables always take non-negative values. Identify the non-negativity constraint of the linear programming. For example, $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$. 
Sample Problem

Maximize $\sum_{j=1}^{n} c_j x_j$ subject to the constraints

$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall i = 1, 2, 3, \ldots, m$

$x_j \geq 0 \quad \forall j = 1, 2, 3, \ldots, m$
Example 17.1  Let there be three items A, B, and C. The raw material required to manufacture these products are $r_1$ and $r_2$. The respective daily allotment of raw materials is 90 and 80 due to the shortage of raw materials. The profits of manufacturing the products are given in Table 17.1. Formulate an LPP such that the profit is maximized, taking into account the given constraints of the allotment of raw materials.
Table 17.1 Formulation of LPP for product profit

<table>
<thead>
<tr>
<th>Product</th>
<th>Raw material $r_1$</th>
<th>Raw material $r_2$</th>
<th>Profit per unit (in ₹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>6</td>
<td>150</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>5</td>
<td>200</td>
</tr>
<tr>
<td>Daily allotment</td>
<td>90</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
Example

The mathematical formulation of this problem is given as follows:

Maximize $z = 100x_1 + 150x_2 + 200x_3$

$3x_1 + 4x_2 + 5x_3 \leq 90$

$2x_1 + 6x_2 + 5x_3 \leq 80$

$x_1, x_2, x_3 \geq 0$
What is Graph Method?
What is a graph method?

The graphical method is thus used to find the feasible region for the given constraints, and all points of this feasible region are feasible solutions. The following terminologies are important for understanding the graphical method further:

**Convex set**  A region (or set $S$) is convex, if any two points of the region are connected by a line such that the line also falls within the set.

**Redundant constraint**  A constraint that does not affect the feasible region is called a redundant constraint.
Procedure for Graphical Method

**Step 1:** Consider the inequality constraints as equality constraints. A line can be drawn for every constraint.

**Step 2:** Determine the feasibility region. Recollect that a feasible region is a collection or set of all feasible solutions. Corners of the feasible region can be solved either from the graph or by solving the constraint equations.

**Step 3:** Determine the values of the objective function. The maximum value of the objective function and its corresponding vertex provide the solution of the given problem, if it is a maximization problem. On the other hand, the minimization problem is associated with the minimum value of the objective function, and its corresponding vertex area is the solution of the problem.
Example

**Example 17.4** Consider the following LPP:

Maximize \( z = 7x_1 + 3x_2 \) subject to the following constraints:

\[
\begin{align*}
2x_1 + 6x_2 &\leq 24 \\
6x_1 + 2x_2 &\leq 24 \\
x_1, x_2 &\geq 0
\end{align*}
\]

Solve the problem using the corner point method.
Example

1. Consider the first constraint
   \[2x_1 + 6x_2 \leq 24\]
   (a) The corner method first assumes inequality as an equality. Thus, the equation becomes as follows:
   \[2x_1 + 6x_2 = 24\]
   (b) On substituting \(x_1 = 0\) in this equation, one gets the following equation:
   \[\therefore 6x_2 = 24 \Rightarrow x_2 = 4\]
   Hence, the coordinate point A is (0,4).
Example

(c) Then by substituting $x_2 = 0$ in this equation, one gets the following relation:

$$2x_1 = 24 \Rightarrow x_1 = 12$$

Therefore, the coordinate point B is (12,0).

(d) Plot these points as a line in a graph as shown in Fig. 17.3.

![Graph for first constraint](image-url)
Example...

2. Similarly, plot the line for the second constraint equation:

\[6x_1 + 2x_2 \leq 24\]

(a) Convert the inequality to equality, as follows:

\[\therefore 6x_1 + 2x_2 = 24\]

(b) Let us assume that \(x_1 = 0\); therefore,

\[2x_2 = 24\]
\[x_2 = 12\]

Therefore, the coordinate point D is \((0, 12)\).

(c) Then let us assume that \(x_2 = 0\); hence,

\[6x_1 = 24\]
\[x_1 = 4\]

Therefore, the coordinate point E is \((4, 0)\).

(d) Plot these points as a line to obtain to a graphical solution, as shown in Fig. 17.4.
3. To find the feasible region, plot the lines together as shown in Fig. 17.5.
Point C

C can be obtained by solving the following equations:

\[ 6x_1 + 2x_2 = 24 \]
\[ 2x_1 + 6x_2 = 24 \]

\[ R_1 - 3R_2: \text{ one gets} \]
\[ -16x_2 = -48 \]
\[ x_2 = 3 \]

Substituting this value back into the equation,

\[ 6x_1 + 6 = 24 \]
\[ 6x_1 = 21 - 6 = 18 \]
\[ x_1 = 3 \]

This shows that

\[ x_1 = x_2 = 3 \]

\[ \therefore \text{ The points of the vertex are } (3, 3) \text{ at corner C.} \]
Final Solution

Thus, the objective function is

\[ z = 7x_1 + 3x_2 \]

\[ O(0, 0) = 0 \text{ at the point } (0, 0) \]

\[ E(4, 0) = 7(4) + 3(0) = 28 \text{ at the point } (4, 0) \]

\[ C(3, 3) = 7(3) + 3(3) = 30 \text{ at the point } (3, 3) \]

\[ A(0, 4) = 7(0) + 3(4) = 12 \text{ at the point } (0, 4) \]

The points D(0, 12) and B(12, 0) yield values 36 and 84, respectively. However, as they are not part of the feasible region, they are ignored. Therefore, one can observe that the maximum profit is 30 at \( x_1 = 3 \) and \( x_2 = 3 \), that is, vertex C. Thus, the optimal solution of this problem is obtained.
Exceptional Cases

**Multiple optimal solutions**  A case of non-unique solution arises when the line is parallel to one of the lines that bound the feasible region, as shown in Fig. 17.8. In this case, multiple optimal solutions are possible.
Exceptional cases

**Infeasible solutions**  There may be a situation where no points satisfy all the constraints of the problem. This leads to infeasible solutions.

**Unbounded solutions**  When one or more variables keep increasing without violating the constraints of the given problem, the objective function value would also be very large. This leads to unbounded solutions.
Simplex Method

The simplex method is a non-graphical procedure for solving canonical LPPs in an iterative manner. The procedure starts with an initial basic feasible solution. Then it finds the next best solution in the next iteration. Iterations are repeated till the optimal solution is obtained.
Informal Algorithm for Simplex Method

Step 1: Formulate the problem as an LPP.
Step 2: Convert any inequality constraint into an equality constraint of linear programming by adding slack variables.
Step 3: Calculate the initial basic feasible solution and construct an initial simplex table.
Step 4: Check for the optimality condition (optimality test).
   4a: Compute $z_j - c_j$, where $z$ is the objective function and $c$ is the coefficient of decision variables that appear in the objective function $z$.
   4b: If $z_j - c_j \geq 0$, then the solution obtained in Step 4 is optimal. Otherwise, the optimal solution is yet to be obtained.
Step 5: Determine the incoming or entering variable to the basis. Basis is a set of basic variables associated with the given problem. Basis or Base column is the column of all basic variables.
Step 6: Choose a column in the objective function row that represents a greater negative value. In the case of a tie, the basic variables are chosen. This column is called pivot column. Similarly, identify the pivot row, and this determines the value that goes out. Divide the independent column and the elements of the pivot column. If the division results in zero, then the problem is completed. Otherwise, choose the lowest positive quotient, and this indicates the row that goes out of the base. This row is called the 'pivot row'. If there is a tie, then a choice is made in favour of the basic variable. The pivot element is the intersection of the pivot row and the pivot column.

Step 7: Determine the new solution as follows:
7a: Divide the entire pivot row by the pivot element to get a new pivot row.
7b: The new row = old row – k × new pivot row. Here, k is an element that is present in the pivot column of the row that is being manipulated.

Step 8: Update the table and revise the solution till $z_j - c_j$ is either zero or positive.
Example

**Example 17.7** Solve the following problem using the simplex method:

Maximize \( z = 4x_1 + 7x_2 \)

Subject to the following constraints:

\[
\begin{align*}
4x_1 + 3x_2 & \leq 12 \\
2x_1 + 4x_2 & \leq 12 \\
x_1 & \geq 0 \\
x_2 & \geq 0
\end{align*}
\]

**Solution** Convert this set of equations into a canonical form by introducing slack variables (i.e., \( x_3 \) and \( x_4 \)). The canonical form of an LPP is given as follows:

\[
\begin{align*}
4x_1 + 7x_2 + 0 \cdot x_3 + 0 \cdot x_4 & = 12 \\
2x_1 + 4x_2 & = 12 \\
x_1, x_2, x_3, x_4 & \geq 0
\end{align*}
\]
Construct the initial simplex table as shown in Table 17.4.

<table>
<thead>
<tr>
<th>Table 17.4</th>
<th>Simplex table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base</strong></td>
<td>C₀</td>
</tr>
<tr>
<td><strong>x₃</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>x₄</strong></td>
<td>0</td>
</tr>
</tbody>
</table>

Look for the maximum number in Z row without giving attention to its sign. It is -7. Therefore, choose this column as the pivot column. It can be observed that the pivot element can be calculated by dividing the independent column and the elements of the pivot column ($θ_{z₀} = Z₀/x₄$) as follows:

θ₃₄ = 12/3 = 4  θ₄₄ = 12/4 = 3

Here θ is a variable that indicates the ratio of independent column to the elements of the pivot column, and determines the departing variables. The minimum of the aforementioned calculation is 3. Therefore, choose the x₃ row as the pivot row. Hence, the variable x₃ enters the base (or basis), and the variable that goes out of the base or basis is x₄. Thus, the pivot row
The simplex table is updated using the new formula: [new row = (old row) − (coefficient or pivot element of the old row of the incoming variable) × new pivot row]. The computation for the new row $x_3$ is shown in Table 17.5.

<table>
<thead>
<tr>
<th></th>
<th>12</th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>0</th>
<th>Old row contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>Coefficient</td>
</tr>
<tr>
<td>3</td>
<td>½</td>
<td>1</td>
<td>0</td>
<td>¼</td>
<td></td>
<td>Pivot row</td>
</tr>
<tr>
<td>3</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td>−0.75</td>
<td>New row = old row − (coefficient) × new pivot row</td>
<td></td>
</tr>
</tbody>
</table>

Similarly, the objective row is computed from $(z - c_i)$. This is done by multiplying $C_5$ of $x_1$ and $x_2$, that is, 0 and 7, and its associated vector of $[x_1, x_2, x_3, x_4]$. This gives $0 \times [2.5 \ 0 \ 1 \ -0.75] - [4 \ 7 \ 0 \ 0] + 7 \times [0.5 \ 1 \ 0.25] - [4 \ 7 \ 0 \ 0]$. Thus, it can be seen that

\begin{align*}
7 \times 0.5 - 4 &= -0.5 \\
7 \times 1 - 7 &= 0 \\
7 \times 0 - 0 &= 0 \\
7 \times 0.25 - 0 &= 1.75
\end{align*}
Substituting $x_1$ and $x_2 = 7$ in the original equation, one gets $Z = 21$. The new updated table is shown as Table 17.6.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>$C_0$</th>
<th>$Z_0$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>0</td>
<td>3</td>
<td>2.5</td>
<td>0</td>
<td>1</td>
<td>-0.75</td>
</tr>
<tr>
<td>$x_3$</td>
<td>7</td>
<td>3</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>$Z$</td>
<td>21</td>
<td>-0.5</td>
<td>0</td>
<td>0</td>
<td>1.75</td>
<td></td>
</tr>
</tbody>
</table>

It can be observed that the pivot column is chosen based on the only negative number present in the preceding table, which is $-0.5$. The pivot row can be calculated based on the $\theta$ ratio, as follows:

\[
\theta_{x_3} = \frac{3}{2.5} = 1.2
\]

\[
\theta_{x_2} = \frac{3}{0.5} = 6
\]
This is updated in Table 17.8. Similarly, the objective row can be calculated by multiplying the contribution factor of $x_1$ and $x_2$ (i.e., $Z$) with the corresponding original contribution, (this is $4 \times [1 \ 0 \ 0.4 \ -0.3] + 7 \times [0 \ 1 \ -0.2 \ 0.4] - [4 \ 7 \ 0 \ 0])$ and the results are as follows:

\[
\begin{align*}
(4 \times 1) + (7 \times 0) - 4 &= 0 \\
(4 \times 0) + (7 \times 1) - 7 &= 0 \\
(4 \times 0.4) + (7 \times (-0.2)) - 0 &= 0.2 \\
(4 \times -0.3) + (7 \times 0.4) - 0 &= 1.6.
\end{align*}
\]

Substituting the values of $x_1$ and $x_2$ in the original objective function gives the value of $z = 21.6$. The new table is shown as Table 17.8.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>4</th>
<th>7</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>$C_0$</td>
<td>$Z$</td>
<td>$x_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>$x_1$</td>
<td>4</td>
<td>1.2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>7</td>
<td>2.4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$Z$</td>
<td>21.6</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Final Solution

It can be observed that none of the entries of the Z row of the current table is negative. Hence, the simplex method terminates. Therefore, it can be observed that the optimal solution is $Z = 21.6$, with $x_1 = 1.2$ and $x_2 = 2.4$. 
Non-Canonical Problems - Minimization

1. Objective function is of minimization type.
2. The constraints may have equality constraints and inequality involving ‘≥’ sign.
How to transform constraints

Handling Minimization of Objective Function
The simplest way of tackling a minimization problem is to multiply the objective function by \(-1\). In other words, minimize \(z = f(x)\) is equivalent to maximize \(z = f(-x)\). For example, if the original objective function is minimization of \(-x_1 + 4x_2\), then the corresponding maximization function obtained by multiplying the objective function by \(-1\) is given as follows:

Maximize \(x_1 - 4x_2\)

Conversion of Inequalities
Many times the non-canonical form of an LPP may have constraints having ‘=’ or ‘\(\geq\)’ rather than \(\leq\) sign. Therefore, if any such situations arise, then all these constraints are required to be converted to canonical forms. Therefore, we start with a non-feasible solution and work towards an optimal solution.
Contd..

**Surplus variables**  A surplus variable is a positive variable that is subtracted to convert an inequality of type ‘\( \geq \)’ to an equality of ‘\( = \)’ form. For example, consider the following inequality equation:

\[
2x_1 + 6x_2 \geq 10
\]

This can be converted to an equality equation as follows:  
\[
2x_1 + 6x_2 - x_3 = 10
\]

Here, the variable \( x_3 \) is called a surplus variable. It can be observed that it is a negative value and is subtracted. A surplus variable has to be introduced as a zero coefficient in the objective function.

**Artificial variables**  An artificial variable is added to satisfy the non-negativity constraint. This variable is used for constraints having an ‘\( \leq \)’ sign. Sometimes, introduction of surplus values may not lead to a feasible solution. Initially, a simplex starts with a zero solution by assigning zero to the decision variables. If zero is assigned to the basic variables \( x_1 \) and \( x_2 \), then the slack variable can sometimes become negative. To avoid this, an artificial variable is introduced to ensure that the constraint is non-negative. For example, \[
2x_1 + 6x_3 - x_3 + A_1 = 10
\]
creates an artificial variable. Here, \( A_1 \) is called an artificial variable. In the objective function, it is taken as \(+A\) if it is a maximization problem and \(-A\) if it is a minimization problem.
Algorithm

Equality constraints Sometimes, the constraints of the given problem have an equality constraint. For example, $x_1 + x_2 = 3$ is an equality constraint. Suppose a constraint is $x_1 + x_2 = 3$, then it can be replaced by two equivalent inequality constraints, as follows:

$$x_1 + x_2 \leq 3$$

Unrestricted variables Let us consider the following LPP:

Maximize $x_1 + x_2$
$x_1 + 2x_2 \leq 3$
$x_1 - x_2 \leq 2$

The problem does not have non-negativity constraints for $x_1$ and $x_2$. In other words, the variables $x_1$ and $x_2$ are unrestricted with respect to signs. Therefore, these variables have to be changed to the following forms:

$x_1 = x_1' - x_1''$
$x_2 = x_2' - x_2''$
Two-Phase Method

Phase I  Artificial variables are introduced to provide an initial basic feasible solution. If the minimum value of the objective function is zero, then Phase II is initiated.

Phase II  In Phase II, the initial basic solution of Phase I is used, the original objective function is reused, and the normal simplex procedure is used to compute the final solutions.
Principle of Duality

1. Identify the variables of the given primal LPP. The number of variables will be equal to the number of constraints of the given problem.
2. The objective function of the dual problem will be written based on the RHS of the primal problem.
3. The column coefficients of the primal problem will now become the row coefficients of the primal problems.
4. Minimization of the primal problem will become maximization in the dual problem, and vice versa.
5. Operator ≤ in the primal problem would become ≥ in the dual one, and vice versa.
6. Dual variables associated with the ‘=’ sign in the constraint of the original problem will be unrestricted with respect to sign in the dual problem.
Example 17.9  Consider the following problem:

Maximize $z = 4x_1 + 6x_2$

Subject to the following constraints:

$2x_1 + 3x_2 \leq 10$
$x_1 + 2x_2 \leq 30$
$2x_1 + 4x_2 \leq 35$
$x_1, x_2 \geq 0$
**Solution**  The given problem is of maximization type. Therefore, the dual of this problem is of minimization type. The variables of the given problem can be given as follows:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Coefficient</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>$x_2$</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Now the dual problem can be given as follows:

Minimize $z' = 10y_1 + 30y_2 + 35y_3$

Subject to the following constraints:

- $2y_1 + y_2 + 2y_3 \geq 4$
- $3y_1 + 2y_2 + 4y_3 \geq 6$
- $y_1, y_2, y_3 \geq 0$

This is the dual of the given problem.
Max-flow algorithm

One encounters ‘flow’ in our daily life such as traffic flow, water flow, or money flow. A flow is a movement of a commodity from one place to another. A max-flow problem is one of the most important network problems encountered in linear programming, whose objective is to maximize the flow from a designated source to a designated sink. A network flow graph is a directed graph where there are two special nodes or vertices, called the source and destination. A network flow graph is also known as a transportation graph.
1. All the edges of this transportation graph can be visualized as pipes. The flow of diesel can be denoted as $x_{ij}$. Flow is always a positive quantity. That is,

$$x_{ij} \geq 0 \text{ for all valid destinations of } i \text{ and } j$$

2. The pipe size can be defined in terms of capacity. Let the capacity of the edge of the transportation graph be $c_{ij}$. Here $i$ and $j$ are valid nodes of the transportation network. A flow can never be greater than the capacity. This is mathematically represented as follows:

$$x_{ij} \leq c_{ij} \text{ for all valid destinations of } i \text{ and } j$$

The capacity $c_{ij}$ is always greater than zero if an edge exists between any direct nodes $i$ and $j$. If $(u, v) \in E$, then $c_{ij} = 0$. The flow $x_{ij}$ is always less than the capacity $c_{ij}$. This is called a capacity constraint.

3. There are designated source, the origin of piping of diesel, and a specially designated node, called a sink, that represents the final destination of the piping.
4. As per Kirchhoff’s law, the amount flowing in is equal to that flowing out. In other words, a site cannot store the quantity that is flowing. If $\sum_{k=1}^{n} x_{i,k}$ is the quantity that is flowing out
Assumptions

and $\sum_{k=1}^{n} x_{k,i}$ is the quantity that is flowing in, then it can be observed that

$$\sum_{k=1}^{n} x_{i,k} = \sum_{k=1}^{n} x_{k,i}$$
Informal algorithm

Step 1: Start with no flow at all in the network. This implies that

\[ x_{1j} = 0, \text{ for all nodes } 2 \text{ to } N - 1 \]

Step 2: Find the augmenting path connecting \( s \) and \( t \). A path \( P \) is augmenting if it exists from the source to destination. Find the minimum flow along the path.

Step 3: Increase and adjust the flow of the network based on the direction of edges, as per Step 2. Store the minimum flow of the path.

Step 4: Repeat Steps 1–3 till no augmenting path exists.
**Example 17.12** Consider the transportation network shown in Fig. 17.10. Apply the Ford–Fulkerson algorithm and find the max-flow.

![Transportation Graph](image)

**Solution** The capacity of the transportation graph (given in Fig. 17.10) is expressed as a capacity matrix shown in Table 17.13.
Table 17.13  Capacity matrix

<table>
<thead>
<tr>
<th>Nodes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 17.11  Initial flow graph
Construct augment path 1-3-5

**Fig. 17.12** Flow graph for edge 1–3–5

**Table 17.14** Updated flow

<table>
<thead>
<tr>
<th>Edges</th>
<th>Total capacity</th>
<th>Current load</th>
<th>Excess capacity of edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–3</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3–4</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4–5</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
</tbody>
</table>
Fig. 17.13  Flow graph of edge 1–3–4–5
Path 1-2-3-5

<table>
<thead>
<tr>
<th>Edges</th>
<th>Total capacity</th>
<th>Current load</th>
<th>Excess capacity of the edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–2</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2–4</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4–5</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
Final flow graph

Fig. 17.14  Final flow graph
Incidentally, it can be observed that the final flow graph is similar to the original capacity shown in Fig. 17.9, and also all the capacities of the edges are fully utilized. Thus, it can be concluded that the max-flow is $0 + 4 + 4 + 2 = 10$. There is no further augmenting path. Hence, the algorithm terminates with a maximum flow of 10. The maximum route that can move more commodities is 1–3–4–5, with 18 units. The maximum flow is, however, 10 as per the constraints of the algorithm.
Min-cut Max-flow

Fig. 17.15 Some min-cuts
Formal algorithm

Algorithm Ford-Fulkerson(G)

%% Input: Graph G
%% Output: Max-flow
Begin
For every edge (i,j) ∈ E do
    flow = 0
End for
While (there exists an augmenting path P in residual graph) do
    Find min-flow in an augmenting path P
    For all (i,j) of P do
        Update min-flow using P and modify the residual network
    End for
End while
Return min-flow
End
Bipartite Graph
**Alternate path**  An alternate path alternates between matching and non-matching edges.

**Flipping edges**  An augmenting path is obtained by a flipping operation.
Flipping of Edges

Fig. 17.17 Flipping of edges
Informal algorithm

**Step 1:** Start with matching $M$ as null.

**Step 2:** Repeat the following steps:
- For all free vertices call a BFS algorithm to find an augmenting path $P$.
- If the augmenting path $P$ exists, then flip the edges.
- Else stop and report the current matching as maximum matching.

**Step 3:** Return matching $M$. 
Algorithm bipartite_matching(G)

%% Input: Bipartite graph G
%% Output: Maximum matching M

Begin
    M = Null
    Find augmenting path P
    while (P exists) then
        M = M ⊕ P
        Find next augmenting path P
    End while
    Return M
End
Complexity analysis

*Complexity Analysis*
Let the number of vertices of a given bipartite graph be $n$ and edges be $m$. Then the time taken by the algorithm is equal to the number of iterations and time per iteration. The number of iterations is at most $(n/2)$, and time taken per iteration is addition of an edge. This is at most $m$. Therefore, the time complexity of the algorithm is $O(n/2 \times m) = O(mn)$.
Stable Marriage Problem

The stable marriage problem is a classic example of iterative improvement design. The objective of the stable matching problem is to match $n$ men and $n$ women in a stable manner. Matching, in this context, is a marriage relation in which all men and women are uniquely assigned in a unique manner. If all men and women are assigned, then it is called perfect matching.
Informal algorithm

Thus, Gale and Shapely algorithm can informally be given as follows:

**Step 1:** Initially, all men and women are ‘free’.
**Step 2:** Let $m$ be the first unmatched (i.e., free) man.
**Step 3:** Find a women $w$ such that
   - 3a: $w$ is the most desirable women in $m$ list
   - 3b: $m$ is more desirable to $w$ than her current spouse $Y$ or $w$ is free
**Step 4:** Match $(m, w)$.
**Step 5:** Repeat Steps 2–4 till all men and women are matched.
**Step 6:** End and return stable pairs.
Formal algorithm

**Algorithm galeshapely(m,w)**

```plaintext
% % Input: Array of m[1 .. n] and w[1 .. n] that has preference order and status
% % Output: Stable pair of m and w
Begin
Assign status of each men m[1 .. n] and women w[1 .. n] as free
While (free man exists) do
    m chooses his favourite women in the same order m[1 .. n]
    case (status of w) do
        % % Status of women in the array [1 .. n]
        status = free: Assign (m,w)
        exit
        status ≠ free: if w prefers m then
            assign (m,w)
        else
            Reject proposal of m
        End if
    End case
End while
Return stable pairs (m,w)
End
```
Complexity Analysis

There are $n$ men and $n$ women. Every man proposes to at most $n$ women and no man proposes to the same woman twice; and in every step, an unmarried person gets a match or a the spouse is changed. Hence, at most $n^2$ proposals are involved in this algorithm. Therefore, at most $O(n^2)$ time is required for a complete match of men and women.
Alternate path  A path that alternates between matching and non-matching edges
Artificial variable  A variable that is introduced to avoid the scenario of zero slack variables during the basic feasible solution
Augmenting path  Any path that exists between specially designated vertices—source and destination
Capacity matrix  A matrix that represents the capacity of the edges of a transport network
Constraints  Conditions or restrictions in the given problem
Convex set  A region (or set S) in which any two points are connected by a line such that the line also falls within the set
Corner extreme points  Vertices or corner points of the feasible region that represent solutions of the given problem.
Cut  A set of edges, which when removed results in a condition of having no path between the source and destination
Cut capacity  The sum of all edges of a cut
Decision variable  An unknown that needs to be computed
Linear function  A function that consists of variables having a power of 1
Linear programming  A mathematical technique for finding the optimal solution
Matching  A condition of pairing between two vertices
Maximum matching  Matching of maximum cardinality
Model  A set of equations that characterize a given problem
Non-negativity constraint  A condition that forces a variable to be positive
Objective function  The goal of the problem that is expressed as a linear function of variables that need to be maximized or minimized as per the problem requirement
Optimal solution  The best solution that satisfies the given objective function and constraint
Perfect matching  The resulting condition when one could pair all vertices
Primal and dual  The primary LPP and its equivalent LPP, respectively
Glossary

**Degeneracy** A phenomenon of a left out variable in a tie situation resulting in zero in the next iteration of the simplex table.

**Duality** The condition that every LPP has another corresponding LPP.

**Feasible region** A collection of all feasible solutions.

**Feasible solutions** All possible values of decision variables that satisfy the given objective function and constraints.

**Flow** The movement of a commodity in a network.

**Free vertex** An unassigned vertex in a bipartite graph.

**Graphical method** A method for solving an LPP using a graphical procedure to obtain the feasible region and vertex corner.

**Inequality** The presence of ≤ or ≥ operator as part of linear equations, instead of the = sign.

**Infeasible solutions** Situations where no points satisfy all the constraints of the problem.

**Iterative improvement** The technique of proposing an initial solution and improving it in successive iterations.

**Redundant constraint** A constraint that does not affect the feasible region.

**Simplex method** A method for solving an LPP using the concept of extreme points.

** Slack variable** A variable that can convert an inequality to an equality.

**Stable pairs in stable marriage** A condition that the preferences of the partners match.

**Surplus variable** A variable having a negative value that is introduced to convert the inequality of type ≥ to an equality.

**Two-phase simplex method** A variant of simplex method that is used to solve minimization problems directly.

**Unbounded solution** A solution such that optimal solutions continue to grow, when feasible values are large.

**Unbounded solutions** Situations such that when one or more variables keeps increasing without violating the criteria, the objective function value also increases.

**Unrestricted variable** A variable that has no restriction with respect to sign.