CHAPTER 5

Covering problem for fuzzy linear optimization problem
CHAPTER 5
Covering problem for fuzzy linear optimization problem

5.1 Background and motivation

In Chapter 4, we considered a linear optimization model with fuzzy relation equations as constraints. Review of previous literatures shows that finding a simple algorithm for the resolution of fuzzy relation equations is still a challenge today. Different works have been done considering linear optimization models with different compositions of fuzzy relation equations based on classical and algebraic methods. Markovskii [84] considered max-product fuzzy relation equations and showed that solving such equations is closely related to the covering problem, which is an \( NP \)-hard problem. Moreover, its minimal solutions correspond to irredundant coverings and the relation between max-min fuzzy relation equations and the covering problem is more complex—it is no longer possible to establish one-to-one mapping of minimal solutions and the solutions of some covering problems.

Thapar and Pandey [150] considered an optimization model with a linear objective function subject to max-\( t \) fuzzy relation equations as constraints, where \( t \) is an Archimedean t-norm. They applied the concept of covering problem to establish 0-1 integer programming problem equivalent to linear programming problem and proposed a binary coded genetic algorithm to obtain the optimal solution.

Khorram and Ghodousian [60] considered the linear optimization problem with max-average fuzzy relation equations as constraints, which does not belong to the group of max-\( t \) composition, \( t \) being a t-norm. Further work in this regard can be found in Wu [160]. Khorram et al. [61] studied the linear optimization problem with max-star fuzzy relation equations as constraints.
This chapter discusses two linear optimization problems. One is a linear optimization problem with max-$\otimes$ fuzzy relation equations as constraints, $\otimes$ being an Archimedean t-norm. The optimal solution of the problem is obtained by two methods. Firstly, the concept of covering is established to find the solutions of fuzzy relation equations and it is shown that the minimal solutions of equations correspond to irredundant coverings. Concepts of essential, non-essential variables are suggested for fuzzy relation equations. Ways of simplification of coverings and finding irredundant coverings are illustrated by an example. Secondly, using the concept of covering, the linear programming problem is converted into an equivalent 0-1 integer programming problem. A binary coded genetic algorithm is designed to obtain the optimal solution of the problem.

Other problem discussed is a linear optimization problem with max-$\ast$ fuzzy relation equations as constraints, where $\ast$ denotes being an algebraic operator which appears in weaker class of t-norms. Some properties for the existence of solution are established and the optimal solution of the problem is obtained by using a binary coded genetic algorithm.

5.2 The problem I

Let $A=[a_{ij}], 0 \leq a_{ij} \leq 1$, be a $m \times n$ dimensional fuzzy matrix and $b=[b_1, b_2, \ldots, b_n]$, $0 \leq b_j \leq 1$, be a $n$-dimensional vector, then the following system of fuzzy relation equations is defined by $A$ and $b$:

$$x \circ' A = b$$ (5.1)

where $\circ'$ denotes max-$\otimes$ composition of $x$ and $A$, $\otimes$ denotes an Archimedean t-norm from algebra over residuated lattice $L=\langle[0,1], \land, \lor, \otimes, \rightarrow, 0,1\rangle$. It is intended to find a solution vector $x=[x_1, x_2, \ldots, x_m]$, with $0 \leq x_i \leq 1$, such that
\[
\max_{i=1}^{m} (x_i \otimes a_{ij}) = b_j, \ \forall \ j = 1, 2, \ldots, n \tag{5.2}
\]

Let \( I = \{1, 2, \ldots, m\} \) and \( J = \{1, 2, \ldots, n\} \) be the index sets. We are interested in solving the following optimization problem subject to max-\( \otimes \) fuzzy relation equations as constraints:

\[
\text{Min } Z = \sum_{i \in I} c_i x_i \tag{5.3}
\]

s.t. \( \max_{i \in I} (x_i \otimes a_{ij}) = b_j, \ \forall \ j \in J \tag{5.4} \)

\( 0 \leq x_i \leq 1, \ \forall \ i \in I \)

where \( c = [c_1, c_2, \ldots, c_m] \in \mathbb{R}^m \) is a \( m \)-dimensional vector, \( c_i \) represents the weight (or cost) associated with variable \( x_i, \ i \in I \).

Let \( X(A,b) = \{ x \in [0,1]^m \mid x \odot A = b \} \) represents the solution set of fuzzy relation equations (5.4). For any \( x^1, x^2 \in X \), we say \( x^1 \leq x^2 \) if and only if \( x^1_i \leq x^2_i, \ \forall \ i \in I \). Therefore \( \leq \) forms a partial ordering relation on \( X \) and \( (X, \leq) \) becomes a lattice. Equations in (5.4) form a system of latticized polynomial equations. \( \bar{x} \in X(A,b) \) is the maximum solution, if \( x \leq \bar{x}, \ \forall \ x \in X(A,b) \). Similarly, \( \bar{x} \in X(A,b) \) is a minimal solution, if \( x \leq \bar{x} \) implies \( x = \bar{x}, \ \forall \ x \in X(A,b) \). According to [66], if \( X(A,b) \neq \emptyset \), then it is, in general, a non-convex set which can be completely determined by an unique maximum solution \( \bar{x} \) and several minimal solutions \( \bar{x} \).

The maximum solution can be computed explicitly by the residual implicator (pseudo complement). The maximum solution of fuzzy relation equations (5.4) can be obtained by assigning
\[ \tilde{x} = A \rightarrow b = \left[ \min_{j \in J} (a_{ij} \rightarrow b_j) \right] \]  

where

\[ a_{ij} \rightarrow b_j = \sup \{ x_i \in [0,1] | (x_i \otimes a_{ij}) \leq b_j \} \]  

If \( \tilde{X}(A,b) \) denotes the set of all minimal solutions, then the complete solution set of fuzzy relation equations (5.4) can be formed as (5.7).

\[ X(A,b) = \bigcup_{x \in \tilde{X}(A,b)} \{ x \in [0,1]^n \mid \tilde{x} \leq x \leq \tilde{x} \} \]  

5.3 The covering problem

**Definition 5.3.1.** Let \( e_j \) denotes the \( j \)th equation of fuzzy relation equations (5.4) and let \( r = [r_1, r_2, \ldots, r_i, \ldots, r_m] \) be a solution of fuzzy relation equations (5.4). Then for each equation \( e_j \), there exists value \( r_i \) of some variable \( x_i \) such that \( r_i \otimes a_{ij} = b_j \). This value \( r_i \) is said to be a realizing value for equation \( e_j \) and we say that \( e_j \) is realized by \( r_i \) in \( r \).

For a realizing value \( r_i \), the equality \( r_i = a_{ij} \rightarrow b_j \) holds. For \( a_{ij} \geq b_j \), \( (a_{ij} \rightarrow b_j) \otimes a_{ij} = b_j \).

**Definition 5.3.2.** A variable \( x_i \) is said to be essential if \( a_{ij} \geq b_j \) for some \( j \in J \). Define \( E_j = \{ i \in I \mid a_{ij} \geq b_j \}, \forall j \in J \). Essential variable \( x_i \) corresponds to \( i \in E_j \). An essential variable \( x_i \) may have different values for different equations \( e_j \). Clearly, \( r_i \) is the value of essential variable \( x_i \), \( i \in E_j \). A variable \( x_i \) is non-essential if \( a_{ij} < b_j, \forall j \in J \). In other words, a variable \( x_i \) is non-essential if \( i \notin E_j \).
So, the equations of the system (5.4) can be satisfied only by essential variables. Presence of essential variables is a necessary condition for the compatibility of fuzzy relation equations (5.4). It may happen that for \( i \in E_j \), \( x_i \) is an essential variable, but value of \( x_i \) is not equal to \( r_i \). Thus, a system having essential variables, can be both, compatible and non-compatible. And if system has no essential variables, then it is non-compatible.

**Definition 5.3.3.** Let \( \tilde{x}_i = \min_{j \in J} (a_{ij} \rightarrow b_j) \). Then define \( \tilde{x}_i \) as the base value of \( x_i \). We say that \( \tilde{x}_i \) belongs to an equation \( e_j \) if \( \tilde{x}_i = a_{ij} \rightarrow b_j \) is achieved on \( e_j \). The base value \( \tilde{x}_i \) can belong to several equations and an equation can possess base values of several variables.

**Lemma 5.3.4.** The base value \( \tilde{x}_i \) is the maximum value of essential variable \( x_i \) in the solutions of a system (5.4).

**Proof.** Let \([r_1, r_2, \ldots, r_t, \ldots, r_m]\) be a solution to system (5.4). Suppose that \( \tilde{x}_i \) is not the maximum value, i.e. \( \exists r_i \) s.t. \( r_i > \tilde{x}_i \). Let \( \tilde{x}_i \) belong to an equation \( e_j \). Since \( \otimes \) is monotonic, \((r_i \otimes a_{ij}) > (\tilde{x}_i \otimes a_{ij}) = (a_{ij} \rightarrow b_j) \otimes a_{ij} = b_j \) for \( a_{ij} \geq b_j \), i.e. \((r_i \otimes a_{ij}) > b_j \). So \( r_i \) violates equation \( e_j \), a contradiction.

**Corollary 5.3.5.** The maximum value of an essential variable is equal to its base value and for non-essential variable this value is 1.

**Lemma 5.3.6.** If an essential variable \( x_i \) has a realizing value \( r_i \) in some equation \( e_j \), then \( r_i = \tilde{x}_i \) and \( \tilde{x}_i \) belongs to \( e_j \).

**Proof.** If \( r_i \) realizes some equation \( e_j \), then \( r_i = a_{ij} \rightarrow b_j \geq \tilde{x}_i \). But \( r_i \geq \tilde{x}_i \) is impossible by Lemma 5.3.4, therefore \( r_i = a_{ij} \rightarrow b_j = \tilde{x}_i \).
Lemma 5.3.7. If a system of fuzzy relation equations (5.4) is compatible, then for any essential variable $x_i$ there is a solution $[r_1, r_2, \ldots, r_i, \ldots, r_m]$, where $r_i = \tilde{x}_i$.

**Proof.** If a value $r_i$ of an essential variable $x_i$ is not a realizing value in some solution $[r_1, r_2, \ldots, r_i, \ldots, r_m]$, then $r_i < \tilde{x}_i$ according to Lemmas 5.3.4 and 5.3.6. If value of $r_i$ is increased upto $\tilde{x}_i$, then we obtain a solution again. Also, for any equation $e_j$, to which $\tilde{x}_i$ belongs, $\tilde{x}_i \otimes a_{ij} = (a_{ij} \rightarrow b_j) \otimes a_{ij} = b_j$, i.e. $\tilde{x}_i$ realizes $e_j$. For any other equation $e_{j'}$, by the definition of $\tilde{x}_i$, we have $\tilde{x}_i \otimes a_{ij'} = (a_{ij'} \rightarrow b_{j'}) \otimes a_{ij'} = b_{j'}$, i.e. the value $\tilde{x}_i$ does not violate $e_{j'}$.

Theorem 5.3.8. If a system of fuzzy relation equations (5.4) is compatible, then it has the unique maximal solution $\bar{x}$, consisting of the maximal value of all variables.

**Proof.** It follows from the fact that each component of any solution can be independently increased up to its maximal value, defined in Corollary 5.3.5.

The concept of covering can be understood by the help of Table 5.1 given below:

<table>
<thead>
<tr>
<th></th>
<th>$s^1$</th>
<th>\ldots</th>
<th>$s^j$</th>
<th>\ldots</th>
<th>$s^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_i$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_j$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_n$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Covering table $T$
Table 5.1 shows the covering table $T$. A row $s_j$ of Table 5.1 corresponds to an equation $e_j$ and column $s'_i$ corresponds to a variable $x_i$. $s'_j$ is an element located on the intersection of row $s_j$ and column $s'_i$. We say that value of $s'_j$ equals one iff $x_i$ is an essential variable and the base value $\tilde{x}_i$ belongs to equation $e_j$. A column $s'_i$ covers a row $s_j$ iff $s'_j = 1$. In other words, we say that variable $x_i$ and $\tilde{x}_i$ covers equation $e_j$. A set of non-zero columns $C$ forms a covering of a set of rows, if every row of the set is covered by at least one column from set $C$.

**Theorem 5.3.9.** A system of fuzzy relation equations (5.4) is compatible iff there exists a covering $C$ for all rows of the table $T$.

**Proof.** Let us consider that system of fuzzy relation equations (5.4) is compatible. Then we will show that there exists a covering $C$ for all rows of the table $T$. For any row $s_j$ of the covering table $T$, the equation $e_j$ has some realizing value $r_i$ of some variable $x_i$, hence, by Lemma 5.3.6, $r_i = \tilde{x}_i$, and $\tilde{x}_i$ belongs to equation $e_j$. Therefore, by the definition of covering, $s'_j = 1$ and row $s_j$ corresponding to equation $e_j$ is covered by the column $s'_i$. Conversely, if there exists a covering $C$ for all rows of the table $T$, then all the variables which belong to $C$ are equal to their base values, and rest of the variables are equal to zero. Thus, in the solution of the system of fuzzy relation equations (5.4), every equation $e_j$ is realized by any base value $\tilde{x}_i$, covering $e_j$.

**Corollary 5.3.10.** A system of fuzzy relation equations (5.4) is compatible iff the table $T$ has no zero rows.

**Proof.** If a table $T$ has a zero row, then this implies that there does not exist a covering for all the rows of $T$. By Theorem 5.3.9, there cannot be a solution. And vice versa, if $T$ has no zero rows, then at least one covering definitely exists.
**Definition 5.3.11.** A column $s^i$ of the table $T$ corresponding to a variable $x_i$ is redundant in a covering $C$ if after deleting $s^i$ from covering $C$, remainder of $C$ is still a covering. A covering $C$ is said to be irredundant if it has no redundant columns. We denote an irredundant covering by $C$ and the set of all irredundant coverings of the table $T$ by $C(X)$. Therefore for each column $s^i$ of irredundant covering $C$ there exists a row $s_j$ covered only by $s^i$.

A correspondence between the coverings $C$ of the table $T$ and the solutions $X$ of the system of fuzzy relation equations (5.4) is defined with the help of two functions $C(X)$ and $X(C)$.

**Theorem 5.3.12.** $X(C)$ is a minimal solution of system of fuzzy relation equations (5.4).

**Proof.** Only non-zero components in $X(C)$ can be decreased. But these are base values of variables, which belong to $C$. If any of these values is decreased, then an equation $e_j$, covered in $C$ only by $x_i$ (such exists since $C$ is irredundant), cannot be satisfied.

**Theorem 5.3.13.** $C(X)$ is an irredundant covering.

**Proof.** The covering $C(X)$ consists of variables, which have realizing and, hence, positive values in the solution $X$. If $C$ is not irredundant, then some column $s^i$ can be excluded from $C$, so that the remainder $C'$ still is a covering. But then there is a solution $X(C')$, in which all the variables except $x_j$, have the same values, but $x_j = 0$, which contradicts the fact that $X$ is a minimal solution.

**Definition 5.3.14.** A row $s_j$ is redundant if it can be excluded from the covering table $T$ without changing the set of possible coverings. A redundant row corresponds to a
redundant equation, which can be excluded from the system of fuzzy relation equations (5.4) without changing the solution set.

In other words, a row $s_j$ is redundant if every covering of some set of rows, different from $s_j$, covers $s_j$ as well. We say that a row $s_j$ majorizes a row $s'_j$, if $s'_j = 1$ imply $s_j' = 1$. Obviously, $s_j$ is redundant since any column covering $s'_j$ covers $s_j$ also.

**Theorem 5.3.15.** If a row $s_j$ is redundant, then $s_j$ majorizes some other row $s'_j$.

**Proof.** Consider any row different from $s_j$. If $s_j$ does not majorize this row, then this row contains 1 in some column, in which $s_j$ contains 0. Mark this column and find the next row different from $s_j$ and not covered by the marked column. If $s_j$ does not majorize this row, then this row contains 1 in some column, in which $s_j$ contains 0. Mark this column also and then consider other rows different from $s_j$ and not covered by the marked columns. Continuing this process, we shall come to some row $s'_j$, majorized by $s_j$, or we shall exhaust the set of rows different from $s_j$ and not covered by marked columns. But in the second case the marked columns form a covering of all rows different from $s_j$, which does not cover $s'_j$. This contradicts the redundancy of $s_j$.

**5.4 Two sub-problems of problem I**

Fang and Li [27] showed that an optimal solution for the model (5.3)-(5.4) with m or max-product compositions can be obtained from two sub-problems for separating the negative and non-negative coefficients in the objective function. Consider problem (5.3)-(5.4). When $X(A,b) \neq \emptyset$, two special cases exist:
For any given cost vector \( c = [c_1, c_2, \ldots, c_m] \in \mathbb{R}^m \), define \( c' = [c'_1, c'_2, \ldots, c'_m] \in \mathbb{R}^m \) and \( c'' = [c''_1, c''_2, \ldots, c''_m] \in \mathbb{R}^m \) such that

\[
\begin{align*}
  c'_i &= \begin{cases} 
    c_i & \text{if } c_i \geq 0 \\
    0 & \text{if } c_i < 0
  \end{cases} \\
  c''_i &= \begin{cases} 
    0 & \text{if } c_i \geq 0 \\
    c_i & \text{if } c_i < 0
  \end{cases} \\
  \forall i \in I.
\end{align*}
\] 

(5.8)

Obviously, \( c = c' + c'' \). Now, we consider the following two sub-problems:

**Min** \( Z'(x) = \sum_{i \in I} c'_i x_i \) \hspace{1cm} (5.9)

s.t. \( \max_{i \in I} (x_i \otimes a_{ij}) = b_j, \forall j \in J \) \hspace{1cm} (5.10)

\( 0 \leq x_i \leq 1, \forall i \in I \)

**Min** \( Z''(x) = \sum_{i \in I} c''_i x_i \) \hspace{1cm} (5.11)

s.t. \( \max_{i \in I} (x_i \otimes a_{ij}) = b_j, \forall j \in J \) \hspace{1cm} (5.12)

\( 0 \leq x_i \leq 1, \forall i \in I \)

**Lemma 5.4.1.** Let \( \tilde{x} \in X(A, b) \) be the maximum solution of fuzzy relation equations (5.4). If \( c_i \leq 0, \forall i \in I \), then \( \tilde{x} \) is the optimal solution of the problem (5.3)-(5.4).

**Proof.** For any solution \( x \in X(A, b) \), we have \( 0 \leq x \leq \tilde{x} \). Since \( c_i \leq 0, \forall i \in I \), we have

\[
\sum_{i \in I} c_i x_i \geq \sum_{i \in I} c_i \tilde{x}_i.
\]

Therefore, \( \tilde{x} \) is the optimal solution.

**Lemma 5.4.2.** If \( c_i \geq 0, \forall i \in I \), then one of the minimal solutions of fuzzy relation equations (5.4) is an optimal solution of the problem (5.3)-(5.4).
**Proof.** Since \( X(A,b) = \bigcup_{\bar{x} \in X(A,b)} \{ x \in [0,1]^n \mid \bar{x} \leq x \leq \bar{x} \} \), thus for any \( \{ x \mid \bar{x} \leq x \leq \bar{x} \} \), we have \( \sum_{i \in I} c_i x_i \leq \sum_{i \in I} c_i \tilde{x}_i \). Therefore, a minimal solution gives the smallest value of the objective function. Since \( \tilde{X}(A,b) \) consists of finite number of minimal solutions, hence one of the minimal solution of problem (5.3)-(5.4) is an optimal solution.

From, Lemma 5.4.1, it is clear that the maximum solution \( \tilde{x} \) of fuzzy relation equations (5.4), when it exists, solves problem (5.11)-(5.12). Also, by Lemma 5.4.2, one of the minimal solutions of fuzzy relation equations (5.4), say \( \tilde{x}^* \), solves problem (5.9)-(5.10). By combining \( \tilde{x} \) and \( \tilde{x}^* \), we can construct a new solution

\[
x_i^* = 
\begin{cases} 
\tilde{x}_i^* & \text{if } c_i \geq 0 \\
\tilde{x}_i & \text{if } c_i < 0
\end{cases} \quad \forall i \in I \quad (5.13)
\]

If \( X(A,b) \neq \emptyset \) and \( x^* \) is defined according to (5.13), then \( x^* \) is an optimal solution of problem (5.3)-(5.4) with an optimal value \( Z(x^*) = \sum_{i \in I} (c_i^* \tilde{x}_i + c_i^* \tilde{x}_i^*) \).

**5.5 Minimal solutions of the problem I**

Let us consider that after the simplification through redundant coverings, the covering table is simplified as much as possible. Consider some methods of finding the irredundant coverings with the help of simplified covering table.

Simplified covering table after removing redundant coverings is shown in Table 5.2.
Table 5.2: Simplified covering table after removing redundant coverings

<table>
<thead>
<tr>
<th></th>
<th>$s^1$</th>
<th>$\ldots$</th>
<th>$s^j$</th>
<th>$\ldots$</th>
<th>$s^m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_j$</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_n$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.2 consists of two irredundant coverings $\{s^1\}$ and $\{s^2, s^3, \ldots, s^m\}$. We consider algebraic method of searching irredundant coverings by considering the simplified covering table associated with the formal logical expression. Each row of table $T$ is associated with logical sum $p_j = \bigvee x_i$, for all $j$, such that $s_j^i = 1$. The whole table $T$ is associated with the logical product $P = \bigwedge \limits{j} p_j$. Expression $P$ can be reduced according to following laws of bounded distributive lattice (where $\land$ refers to meet (product) and $\lor$ refers to join (sum) operator):

(i) $a \land a = a$, $a \lor a = a$ (Idempotent)
(ii) $a \land b = b \land a$, $a \lor b = b \lor a$ (Commutative)
(iii) $a \land (b \land c) = (a \land b) \land c$, $a \lor (b \lor c) = (a \lor b) \lor c$ (Associative)
(iv) $a \lor (b \land c) = (a \lor b) \land (a \lor c)$, $a \land (b \lor c) = (a \land b) \lor (a \land c)$ (Distributive)
(v) $a \land 1 = a$, $a \lor 1 = a$

After simplification of the logical product $P$, the minterms obtained corresponds to the irredundant coverings which gives the set of minimal solutions of the problem (5.3)-(5.4).
Now our aim is to find the minimal solution with least value of the objective function. Let $k$ be the total number of minterms obtained. Then we say that problem has $k$ irredundant coverings, hence $k$ minimal solutions. Table 5.3 shows the minterms table. Each row of the table corresponds to a minterm. If the $k$th minterm contains the $i$th variable, then the value corresponding to the cell is 1.

Table 5.4 shows the objective function table, where the value of $Z'$ is calculated for each minterm corresponding to a minimal solution. The minimal solution giving minimum value of the objective function is selected.

<table>
<thead>
<tr>
<th>Minterm</th>
<th>$x_1$</th>
<th>...</th>
<th>$x_i$</th>
<th>...</th>
<th>$x_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k'$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Minterms table

<table>
<thead>
<tr>
<th>Minterm</th>
<th>$c_i'x_1$</th>
<th>...</th>
<th>$c_i'x_i$</th>
<th>...</th>
<th>$c_m'x_m$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k'$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Objective function table
Algorithm 1: For obtaining optimal solution of problem (5.3)-(5.4)

Step 1: Find essential and non-essential variables. If system has no essential variables, then it is incompatible, stop.

Step 2: Find base values of all essential variables.

Step 3: Compute maximum solution (the maximum value of all essential variables are equal to their base values and non-essential variables are equal to 1).

Step 4: Form two sub-problems using (5.9)-(5.10) and (5.11)-(5.12).

Step 5: Obtain optimal value \( Z' = \sum_{i \in I} c_i \bar{x}_i \) for problem (5.11)-(5.12).

Step 6-1: Obtain the covering table by excluding all columns corresponding to the non-essential variables and the variables \( \{x_i, i \in I \mid c_i < 0\} \) and all rows corresponding to the equations realized by \( \{x_i, i \in I \mid c_i < 0\} \).

Step 6-2: Obtain simplified covering table.

Step 6-3: Obtain the logical product \( P \). Simplify it by using laws listed in Section 5.5.

Step 6-4: Obtain all irredundant coverings.

Step 6-5: Find optimal solution \( \bar{x}^* \) and optimal value \( Z' = \sum_{i \in I} c_i \bar{x}_i \) for problem (5.9)-(5.10).

Step 7: Compute optimal value of the objective function \( Z = Z' + Z'' \) and find optimal solution \( x^* \) using (5.13) for the original problem (5.3)-(5.4).
Example 5.5.1. Consider the following optimization problem subject to max-Lukasiewicz t-norm fuzzy relation equations with $x \otimes a = \max (0, x + a - 1)$, which is one type of the max-Archimedean t-norm compositions.

Min $Z(x) = x_1 + 2x_2 + 3x_3 - 0.5x_4 + 4.4x_5 + 1.5x_6 + 2x_7 + 2.5x_8 + 1.25x_9$

s.t. $Ax' = b$, $0 \leq x_i \leq 1, i = 1, 2, \ldots, 9$

$A = \begin{bmatrix}
0.18 & 0.23 & 0.75 & 0.43 & 0.70 & 0.65 & 0.42 & 0.82 & 0.35 & 0.45 \\
0.15 & 0.56 & 0.90 & 0.56 & 0.72 & 0.92 & 0.43 & 0.61 & 0.68 & 0.46 \\
0.12 & 0.71 & 0.76 & 0.72 & 0.45 & 0.72 & 0.58 & 0.67 & 0.43 & 0.48 \\
0.25 & 0.62 & 0.32 & 0.57 & 0.54 & 0.61 & 0.70 & 0.65 & 0.76 & 0.36 \\
0.22 & 0.80 & 0.95 & 0.81 & 0.70 & 0.53 & 0.67 & 0.80 & 0.64 & 0.70 \\
0.35 & 0.93 & 0.61 & 0.19 & 0.90 & 0.78 & 0.80 & 0.63 & 0.55 & 0.45 \\
0.21 & 0.45 & 0.49 & 0.80 & 0.34 & 0.82 & 0.33 & 0.54 & 0.45 & 0.52 \\
0.12 & 0.43 & 0.64 & 0.38 & 0.46 & 0.62 & 0.45 & 0.76 & 0.25 & 0.32 \\
0.31 & 0.38 & 0.68 & 0.47 & 0.63 & 0.72 & 0.26 & 0.42 & 0.80 & 0.77
\end{bmatrix}$

$b = [0.00 \ 0.55 \ 0.70 \ 0.56 \ 0.52 \ 0.72 \ 0.42 \ 0.64 \ 0.48 \ 0.45]$

Step 1: In above example, all variables are essential and there is no non-essential variable.

Step 2: Essential variables have the following base values (the numbers in brackets are the numbers of equations to which the base values belong):

$x_1 = 0.82 \ (1,5,8), \ x_2 = 0.80 \ (1,3,5,6,9), \ x_3 = 0.84 \ (1,2,4,7), \ x_4 = 0.72 \ (1,7,9),$

$x_5 = 0.75 \ (1,2,3,4,7,10), \ x_6 = 0.62 \ (1,2,5,7), \ x_7 = 0.76 \ (1,4), \ x_8 = 0.88 \ (1,8),$ 

$x_9 = 0.68 \ (1,9,10).$

Step 3: Maximum solution is given by

$\hat{x} = [0.82 \ 0.80 \ 0.84 \ 0.72 \ 0.75 \ 0.62 \ 0.76 \ 0.88 \ 0.68].$
Step 4: Sub-problem 1: Min $Z'(x) = -0.5x^4$

Sub-problem 2: Min $Z'(x) = x_1 + 2x_2 + 3x_3 + 4.4x_5 + 1.5x_6 + 2x_7 + 2.5x_8 + 1.25x_9$

Step 5: Optimal solution for sub-problem 1: $Z'(\bar{x}) = -0.5 \times 0.72 = -0.36$

Step 6-1: Obtain covering table (Table 5.5). Since $c_4 < 0$, hence variable $x_4$ is excluded from the columns of the covering table and equations 1, 7, 9 are excluded from the rows of the covering table.

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_2$</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_3$</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_4$</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_5$</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_8$</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_{10}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.5: Covering table-Example 5.5.1

Step 6-2: Now from Table 5.5 it is clear that rows $e_3$ and $e_5$ majorizes $e_6$, hence $e_3$ and $e_5$ can be excluded from the covering table to obtain simplified covering table (Table 5.6).
Table 5.6: Simplified covering table—Example 5.5.1

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$x_8$</th>
<th>$x_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_2$</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_4$</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_6$</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e_8$</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>$e_{10}$</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

Step 6-3: Obtain the logical product

$$P = (x_3 \lor x_5 \lor x_6) \land (x_2 \lor x_3 \lor x_4) \land (x_1 \lor x_9) \land (x_3 \lor x_9).$$

After simplification $P$ is obtained as follows:

$$P = (x_1 \land x_2 \land x_5) \lor (x_2 \land x_3 \land x_6) \lor (x_1 \land x_2 \land x_3 \land x_6) \lor (x_2 \land x_3 \land x_8 \land x_9)$$

$$\lor (x_1 \land x_2 \land x_6 \land x_7 \land x_9) \lor (x_2 \land x_6 \land x_7 \land x_8 \land x_9).$$

Step 6-4: From Step 6-3, irredundant coverings obtained are

$$(x_1, x_2, x_3), (x_2, x_3, x_6), (x_1, x_2, x_3, x_6), (x_2, x_3, x_9, x_9), (x_1, x_2, x_6, x_7, x_9), (x_2, x_6, x_7, x_8, x_9)$$

Step 6-5: Obtain minterms table (Table 5.7) and objective function table (Table 5.8).

Minimum value corresponds to minterm 1 and minterm 5 (Table 5.8). Hence there two optimal minimal solutions given by

$$\bar{x}^{x_1} = [0.82 \ 0.80 \ 0 \ 0 \ 0.75 \ 0 \ 0 \ 0 \ 0]$$

$$\bar{x}^{x_2} = [0.82 \ 0.80 \ 0 \ 0 \ 0 \ 0.62 \ 0.76 \ 0 \ 0.68]$$

From the objective function table (Table 5.8), $Z'(\bar{x}^{x_1}) = Z'(\bar{x}^{x_2}) = 5.72.$
Step 7: Combining the optimal solutions obtained from sub-problem 1 and sub-problem 2, we obtain two optimal solutions from (5.13):

\[ x^{*1} = [0.82 \ 0.80 \ 0 \ 0.72 \ 0.75 \ 0 \ 0 \ 0 \ 0] \]

\[ x^{*2} = [0.82 \ 0.80 \ 0 \ 0.72 \ 0.62 \ 0.76 \ 0.68] \]
Optimal value of the objective function

\[ Z = Z(x^*) = Z(x^{**}) = Z' + Z'' = 5.72 - 0.36 = 5.36. \]

### 5.6 0-1 integer programming and genetic algorithm to solve problem I

Define \( J_i = \{ j \in J \mid \tilde{x}_i \otimes a_{ij} = b_j \}, \forall i \in I \) and \( I_j = \{ i \in I \mid \tilde{x}_j \otimes a_{ij} = b_j \}, \forall j \in J \). The system of fuzzy relation equations (5.4) is simplified from the covering table after deleting all the redundant rows. The 0-1 integer programming problem equivalent to problem (5.3)-(5.4) with max-min composition was proposed by Fang and Li [27]. According to Theorem 5.3.8, all the variables belonging to covering \( C \) are equal to their base values, while rest of the variables are zero. Furthermore, in the solution of the system of fuzzy relation equations (5.4) every equation \( e_j \) is realized by any base value \( \tilde{x}_i \) covering \( e_j \). Therefore, each component of the optimal solution is either 0 or \( \tilde{x}_i \).

Let us define \( \tilde{x}_i y_i = \begin{cases} 0 & \text{if } y_i = 0 \\ x_i & \text{if } y_i = 1 \end{cases} \), where \( y_i \in \{0,1\}, \forall i \in I \).

Then the equivalent 0-1 integer programming problem to the linear programming problem (5.3)-(5.4) is given as (5.14)-(5.15)

\[
\begin{align*}
\text{Min } \bar{Z} & = \sum_{i \in I} c_i \tilde{x}_i y_i \\
\text{s.t. } & \sum_{i \in l_j} y_i \geq 1, \forall j \in J \\
& y_i \in \{0,1\}, \forall i \in I
\end{align*}
\]  

(5.14)  

(5.15)

Note that the constraints of the problem (5.14)-(5.15) require that, \( \forall j \in J \), there exists at least one \( i \in I_j \), such that \( y_i = 1 \). The theorem below shows that solving problem (5.3)-(5.4) is equivalent to solving 0-1 integer programming problem (5.14)-(5.15).
Theorem 5.6.1. Let $\bar{X}(A,b) = \{ y = (y_i)_{i \in I} \mid y_i \in \{0,1\}, \forall i \in I \}$ be the set of feasible solutions of problem (5.14)-(5.15) and $X(A,b)$ be the set of feasible solutions of problem (5.3)-(5.4). For each $x \in X(A,b)$, there exists $y \in \bar{X}(A,b)$, and vice-versa, such that $Z(x) = \bar{Z}(y)$.

Proof. For $x \in X(A,b)$, $Z(x) = \sum_{i \in I} c_i x_i = \sum_{i \in I} c_i \bar{x}_i \cdot \max \{x_{i,j}\}$, such that $\sum_{i \in I} x_{i,j} \geq 1, \forall j \in J$, where $x_{i,j} = 0$ or $1, \forall i \in I$ and $\forall j \in J$. Let $y_i = \max \{x_{i,j}\}$. Then $Z(x) = \sum_{i \in I} c_i x_i = \sum_{i \in I} c_i \bar{x}_i y_i = \bar{Z}(y)$. Also note that since $x_{i,j} = 0$ or $1, \forall i \in I, j \in J$ therefore $y_i = 0$ or $1, \forall i \in I$. Thus $y \in \bar{X}(A,b)$. Conversely, for any $y \in \bar{X}(A,b)$, for each $y_i = 1$, set $x_{i,j} = 1, \forall j \in J_i$ and for each $y_i = 0$, set $x_{i,j} = 0, \forall j \in J_i$. Therefore, $x_{i,j} = 0$ or $1, \forall i \in I, j \in J$. Also for each $j \in J$, we have $\sum_{i \in I} x_{i,j} = \sum_{i \in I} y_i \geq 1$. Therefore, $\bar{Z}(y) = \sum_{i \in I} c_i \bar{x}_i y_i = \sum_{i \in I} c_i \bar{x}_i \max \{x_{i,j}\} = Z(x)$.

Algorithm 2: For simplification of the problem using concept of covering

Step 1: Get the matrix $A, b$. Find essential and non-essential variables. If system has no essential variables, then it is incompatible, stop.

Step 2: Find base values of all essential variables.

Step 3: Compute maximum solution (the maximum value of all essential variables are equal to their base values and for non-essential variables this value is 1).

Step 4: Find $J_i = \{ j \in J \mid \bar{x}_i \odot a_{ij} = b_j \}, \forall i \in I$. Obtain covering table and find irredundant coverings to obtain simplified covering table.

119
Step 5: Convert the reduced problem to equivalent 0-1 integer programming problem.

There are different methods for solving integer programming problems. In this chapter, binary coded genetic algorithm is designed to solve the problem (5.14)-(5.15) equivalent to the linear programming problem (5.3)-(5.4). Genetic algorithms were introduced as a computational analogy of adaptive systems. They are modelled loosely on the principles of the evolution via natural selection, employing a population of individuals that undergo selection in the presence of variation-inducing operators such as mutation and crossover. A fitness function is used to evaluate individuals, and reproductive success varies with fitness. The genetic algorithm evaluates in each generation the solutions and finally the optimal solution is found which yields the optimal value of the objective function. There is no need to divide the optimization problem into two sub-problems and neither there is need to find the set of minimal solutions. Algorithm below states how the genetic algorithm is initialized.

Algorithm 3: For initialization of the binary coded genetic algorithm

Step 1: Initialize number of generations (say \( g = 50 \)) and number of populations (say \( k = 15 \)).

Step 2: Set generation \( g = 1 \).

Step 3: Randomly generate matrix \( R \) of dimension \( m \times k \) where each element of \( R \) is a binary sting 0 or 1 (note that each column vector of the matrix \( R \) corresponds to a solution of the equivalent integer programming problem).

Step 4: Check for each column (solution vector) of \( R \), whether it satisfies the constraints of equivalent 0-1 integer programming problem or not. If the solution vector satisfies the constraints of the problem, then the solution vector is considered, else left out and the matrix \( R \) is updated.
Step 5: Define matrix $T$ of same dimension as of matrix $R$ (where after updating $R$ is of
dimension $m \times l$, $l \leq k$). For $i = 1, 2, \ldots, m$ and $s = 1, 2, \ldots, l$,

$$T(i, s) = \begin{cases} \hat{x}_i & \text{if } R(i, s) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 6: Obtain matrix $M_{\text{objective}} = c^T \cdot T$.

Step 7: Assign objective function value $Z$, the minimum value from $M_{\text{objective}}$.

Step 8: Apply selection operator, crossover and mutation operators as discussed in Section 5.7.

Step 9: While $g \leq 50$

$$g = g + 1,$$

Goto Step 8.

Step 10: Output the objective function value $Z$ and the optimal solution.

5.7 Selection, crossover and mutation operations

After the initialization process the selection operator is applied. Selection is the stage of a
genetic algorithm in which individual chromosomes (solutions) are chosen from a
population for recombination or crossover. In binary coded genetic algorithm, the
tournament selection operator is applied which simply identifies good solutions in the
population and eliminate bad solutions from the population so that multiple copies of
good solutions can be placed in the population. In our case the current population is the
matrix $R$ and the matrix $M_{\text{objective}}$ represents the values of the objective function
evaluated at different solution vectors of the population. The current population $R$ and
the matrix $M_{\text{objective}}$ is updated after the selection operator.
After the selection operator, one-point crossover is applied on the solution vectors of the population in hand. One point crossover is applied by randomly choosing a crossing site. In the population $R$, two solutions undergo crossover to produce two new solutions. The population $R$ and the matrix $M_{objective}$ is updated after the crossover operator.

The third is the mutation operator which alters the strings (solution vectors) locally to hopefully create a better string. It keeps diversity in the population. A bit-wise mutation operator is applied on the matrix $R$. The population $R$ and the matrix $M_{objective}$ is updated after the mutation operation. After the mutation operator has been applied, again check for each column of $R$, whether it satisfies the constraints of the equivalent 0-1 integer programming problem or not and the matrix $R$ and $M_{objective}$ is updated by removing all infeasible solutions. Now before moving to the next generation, the minimum value from $M_{objective}$ is selected and if this value is less than the previous value of $Z$, then objective function $Z$ is assigned this value and the solution vector corresponding to the minimum value is the solution to the optimization problem. For the next generation, $R$ is taken as the current population and again the three operators are applied and same procedure is repeated.

**Example 5.7.1.** Consider the optimization problem subject to max-Lukasiewicz t-norm fuzzy relation equations discussed in Section 5.5.

Firstly, Algorithm 2 is applied to obtain the simplified problem.

Step 1: In above example, all variables are essential and there is no non-essential variable. System is compatible.

Step 2: Essential variable have the following base values

$\tilde{x}_1 = 0.82, \tilde{x}_2 = 0.80, \tilde{x}_3 = 0.84, \tilde{x}_4 = 0.72, \tilde{x}_5 = 0.75, \tilde{x}_6 = 0.62, \tilde{x}_7 = 0.76, \tilde{x}_8 = 0.88, \tilde{x}_9 = 0.68$.

Step 3: Maximum solution is given by
\[ \bar{x} = [0.82 \ 0.80 \ 0.84 \ 0.72 \ 0.75 \ 0.62 \ 0.76 \ 0.88 \ 0.68] \].

Step 4: \( J_1 = \{1,5,8\}, J_2 = \{1,3,5,6,9\}, J_3 = \{1,2,4,7\}, J_4 = \{1,7,9\}, J_5 = \{1,2,3,4,7,10\}, J_6 = \{1,2,5,7\}, J_7 = \{1,4\}, J_8 = \{1,8\}, J_9 = \{1,9,10\} \).

<table>
<thead>
<tr>
<th></th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( x_6 )</th>
<th>( x_7 )</th>
<th>( x_8 )</th>
<th>( x_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_1 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( e_2 )</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_3 )</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_4 )</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_5 )</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_6 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_7 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_8 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_9 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( e_{10} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.9: Covering table—Example 5.7.1

From Table 5.9, it is clear that row \( e_1 \) majorizes all the rows of the covering table. Also, rows \( e_3, e_5 \) and \( e_9 \) majorizes \( e_6 \) and row \( e_7 \) majorizes \( e_2 \), hence \( e_1, e_3, e_5, e_7, e_9 \) can be excluded from the covering table to obtain simplified covering table (Table 5.10).
Step 5: From the simplified covering table, \( I_2 = \{3, 5, 6\}, I_4 = \{3, 5, 7\}, I_6 = \{2\}, I_8 = \{1, 8\}, I_{10} = \{5, 9\} \). The optimization problem can be converted into an equivalent 0-1 integer programming problem as:

\[
\text{Min } Z = y_1 \tilde{x}_1 + 2y_2 \tilde{x}_2 + 3y_3 \tilde{x}_3 - 0.5y_4 \tilde{x}_4 + 4.4y_5 \tilde{x}_5 + 1.5y_6 \tilde{x}_6 + 2y_7 \tilde{x}_7 + 2.5y_8 \tilde{x}_8 + 1.25y_9 \tilde{x}_9
\]

s.t. \( y_3 + y_5 + y_6 \geq 1 \)

\( y_3 + y_5 + y_7 \geq 1 \)

\( y_2 = 1 \)

\( y_1 + y_8 \geq 1 \)

\( y_5 + y_9 \geq 1 \)

\( y_i \in \{0, 1\}, \forall i = 1, 2, \ldots, 9 \)

Now Algorithm 3 is applied to the equivalent 0-1 integer programming problem of the linear programming problem.

Step 1: Initialize number of generations, say, \( g = 50 \) and number of populations, say \( k = 15 \).
Step 2: Set generation $g = 1$.

Step 3: For the simplified problem, the matrix $R_{m \times k}$ is given below:

$$R = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
\end{bmatrix}$$

Step 4: Check for each column of $R$, whether it satisfies the constraints of the equivalent 0-1 integer programming problem or not

s.t. $y_3 + y_5 + y_6 \geq 1$

$y_3 + y_5 + y_7 \geq 1$

$y_2 = 1$

$y_4 + y_6 \geq 1$

$y_5 + y_9 \geq 1$

$y_i \in \{0, 1\}, \forall i = 1, 2, \ldots, 9.$

Note that only $1^{st}$, $3^{rd}$, $4^{th}$, $7^{th}$, $10^{th}$, $11^{th}$, $12^{th}$ and $13^{th}$ column vectors (solution vectors) of the matrix $R$ satisfy all the constraints. Updated matrix $R$ is obtained as follows:
Step 5: Each of the columns of $R$ represents a solution vector to the equivalent 0-1 integer programming problem. Once a matrix of solution vectors is obtained, it is necessary to evaluate the solution, particularly in the context of the underlying objective function.

\[
R = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1
\end{bmatrix}
\]

Step 6: $M_{\text{objective}} = [7.56, 7.56, 10.37, 10.08, 8.41, 6.72, 7.67, 8.52]$. 

Step 7: Assign objective function value $Z = 6.72$, the minimum value from $M_{\text{objective}}$.

Step 8-Step 10: For this example, a binary-coded genetic algorithm was run for 50 iterations and selection, crossover and mutation operators were applied. The value of the objective function was found to be 5.36 and the corresponding solution vectors for the
0-1 integer programming problem from the population $R$ was obtained as $[1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0]^T$, $[1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1]^T$. Therefore the optimal solutions to the linear optimization problem are

$$x^{*1} = [0.82 \ 0.80 \ 0 \ 0.72 \ 0.75 \ 0 \ 0 \ 0 \ 0],$$
$$x^{*2} = [0.82 \ 0.80 \ 0 \ 0.72 \ 0 \ 0.62 \ 0.76 \ 0 \ 0.68].$$

### 5.8 The problem II

Let $A=[a_{ij}]$, $0 \leq a_{ij} \leq 1$, be a $m \times n$ dimensional fuzzy matrix and $b=[b_1, b_2, ..., b_n]$, $0 \leq b_j \leq 1$, be a $n$-dimensional vector, then the following system of fuzzy relation equations is defined by $A$ and $b$:

$$x \odot A = b \quad (5.16)$$

where $\odot$ denotes max-$*$ composition of $x$ and $A$, $*$ being an algebraic operator. It is intended to find a solution vector $x = [x_1, x_2, ..., x_m]$, with $0 \leq x_i \leq 1$, such that

$$\max_{i=1}^{m} (x_i \cdot a_{ij}) = b_j, \forall j = 1, 2, ..., n \quad (5.17)$$

where

$$x_i \cdot a_{ij} = \frac{x_i + a_{ij}}{1 - x_ia_{ij}} \quad (5.18)$$

Let $I = \{1, 2, ..., m\}$ and $J = \{1, 2, ..., n\}$ be the index sets. We are interested in solving the following optimization problem subject to max-$*$ fuzzy relation equations as constraints:
Min \[ Z = \sum_{i \in I} c_i x_i \]  \hspace{1cm} (5.19)

s.t. \[ \max_{i \in I} (x_i * a_{ij}) = b_j, \forall j \in J \]  \hspace{1cm} (5.20)

\[ 0 \leq x_i \leq 1, \forall i \in I \]

where, \( c = [c_1, c_2, \ldots, c_m]^T \in \mathbb{R}^m \) is an \( m \)-dimensional vector, \( c_i \) represents the weight (or cost) associated with variable \( x_i, i \in I \). Equation (5.20) shows that \( \forall i \in I \), neither \( x_i \) nor \( a_{ij} \) can exceed \( b_j \), for any \( j \), since \( b_j \) is the upper bound. Thus for any \( i \in I \) and \( j \in J \), \( x_i \in [0, \min_{j \in J} (b_j)] \) and \( a_{ij} \in [0, b_j] \). Also for some \( i \in I \) and \( j \in J \), \( a_{ij} = b_j \Rightarrow x_i = 0 \) and conversely, for some \( i \in I \), \( x_i = 0 \Rightarrow \exists j \in J \text{ s.t. } a_{ij} = b_j \). And if for some \( i \in I \), \( a_{ij} = 0, \forall j \in J \Rightarrow x_i = \min_{j \in J} (b_j) \). Thus for unknown \( x_i \), system (5.20) is consistent for \( a_{ij} \leq b_j \). It is observed that \( * \) is continuous, monotonically non-decreasing, commutative and associative operator. Also, max operator has restricted distributive property over \( * \), i.e. for \( a, b_1, b_2 \in [0,1] \), \( a \leq b_1, a \leq b_2 \), \( \max (a, b_1 * b_2) = \max (a, b_1) * \max (a, b_2) \).

This algebraic operator appears in weaker class of t-norms and may give maximum solution as zero. Maximum solution cannot be one. For weak association of input and output, the model (5.19)-(5.20) is useful. The maximum solution is obtained and the problem is converted into an equivalent 0-1 integer programming problem. A binary coded genetic algorithm is applied to find the optimal solution of the problem (5.19)-(5.20) without decomposing the problem into two sub-problems. The algorithm directly searches for an optimal solution of the problem.

**Lemma 5.8.1.** If in the \( j \)th equation \( a_{ij} > b_j \) holds for any \( i \in I \), then the solution set \( X(A,b) = \emptyset \).
Proof. If in the $j$th equation $a_{ij} > b_j$ holds for any $i \in I$, then since $0 \leq x_i < 1$, $x_i + a_{ij} > x_i + b_j$ and $x_i a_{ij} \geq x_i b_j$. Thus, $1 - x_i a_{ij} \leq 1 - x_i b_j$ and

$$\max_{i \in I} \left( \frac{x_i + a_{ij}}{1 - x_i a_{ij}} \right) \geq \frac{x_i + a_{ij}}{1 - x_i a_{ij}} \geq \frac{x_i + b_j}{1 - x_i b_j} > b_j. $$

This leads to $\max_{i \in I} \left( \frac{x_i + a_{ij}}{1 - x_i a_{ij}} \right) > b_j$. Hence there exists no solution for the $j$th equation. Thus $X(A,b) = \emptyset$.

Definition 5.8.2. Let $a_{ij} \leq b_j$, $\forall i \in I$. Then the $i$th component of the maximum vector $\tilde{x}$ is defined by

$$\tilde{x}_i = \min_{j \in J} \left( \frac{b_j - a_{ij}}{1 + a_{ij} b_j} \right), \forall i \in I.$$

Theorem 5.8.3. $\tilde{x}$ is the maximum solution of the problem (5.19)-(5.20).

Proof. From definition of $\tilde{x}_i$, it is clear that $\tilde{x}_i \leq \frac{b_j - a_{ij}}{1 + a_{ij} b_j}$, $\forall i \in I$ and there exists at least one $i \in I$, s.t. $\tilde{x}_i = \frac{b_j - a_{ij}}{1 + a_{ij} b_j}$. It is to be shown that $\tilde{x}$ is the maximum solution of problem (5.19)-(5.20). Suppose that $\tilde{x}$ is not the maximum solution. Let there exists $\bar{x} \in X(A,b)$ such that $\tilde{x}_i > \frac{b_j - a_{ij}}{1 + a_{ij} b_j}$. This implies $\tilde{x}_i (1 + a_{ij} b_j) > (b_j - a_{ij})$ or $\frac{x_i + a_{ij}}{1 - x_i a_{ij}} > b_j$ which is a contradiction. Therefore, $\tilde{x} \leq \bar{x}, \forall \bar{x} \in X(A,b)$ and $\tilde{x}$ is the maximum solution.

Definition 5.8.4. $x \in X(A,b)$ is called a zero solution if $x_i = \tilde{x}_i = \bar{x}_i = 0$, $\forall i \in I$.

Remark 5.8.5. For $i \in I$, if for some $j \in J$, $a_{ij} = b_j$, then $\tilde{x}_i = x_i = \bar{x}_i = 0$. Obviously, if $b_j = 0$ for some $j \in J$, then $a_{ij} = 0$, $\forall i \in I$. Therefore solution is the zero solution, i.e. $\tilde{x}_i = x_i = \bar{x}_i = 0, \forall i \in I$.  

129
Remark 5.8.6. System of fuzzy relation equations (5.20) is consistent for \( j \in J \) if and only if \( a_{ij} \leq b_j, \forall i \in I \), otherwise inconsistent.

Definition 5.8.7. For any solution \( x \in X(A, b) \), \( x_i \) is called a binding variable if
\[
\frac{x_i + a_{ij}}{1 - x_i a_{ij}} = b_j
\]
holds for some \( j \in J \) and a constraint \( j \in J \) is said to be a binding constraint if
\[
\frac{x_i + a_{ij}}{1 - x_i a_{ij}} = b_j
\]
holds for some \( i \in I \).

Let \( X(A, b) \neq \emptyset \). Define \( I_j = \left\{ i \in I \mid \frac{x_i + a_{ij}}{1 - x_i a_{ij}} = b_j \right\} \), \( \forall j \in J \) and \( J_i = \left\{ j \in J \mid \frac{x_i + a_{ij}}{1 - x_i a_{ij}} = b_j \right\} \), \( \forall i \in I \).

Lemma 5.8.8. \( X(A, b) \neq \emptyset \Leftrightarrow I_j \neq \emptyset, \forall j \in J \).

Proof. If \( X(A, b) \neq \emptyset \), then for every \( j \in J \), \( \exists i \in I \), s.t. \( \frac{x_i + a_{ij}}{1 - x_i a_{ij}} = b_j \). Thus \( I_j \neq \emptyset \).

Conversely, if \( I_j \neq \emptyset, \forall j \in J \), there exists a solution for every equation \( j \). Therefore, \( X(A, b) \neq \emptyset \).

Corollary 5.8.9. Let \( x \in X(A, b) \), then \( x \in [0, \min_{j \in J} (b_j)] \).

Lemma 5.8.10. Let \( x \in X(A, b) \), then for each \( j \in J \), there exists \( i_{0} \in I \) such that
\[
\frac{x_i + a_{i,j}}{1 - x_i a_{i,j}} = b_j \text{ and } \frac{x_i + a_{ij}}{1 - x_i a_{ij}} \leq b_j, \forall i \in I.
\]
Proof. For \( x \in X(A, b) \), \( \max_{i \in I} \left( \frac{x_i + a_{ij}}{1-x_i a_{ij}} \right) = b_j, \forall j \in J \). Thus \( \frac{x_i + a_{ij}}{1-x_i a_{ij}} \leq b_j, \forall j \in J \). Hence there exists at least one \( i_c \in I \) such that \( \frac{x_{i_c} + a_{i_c j}}{1-x_{i_c} a_{i_c j}} = b_j \).

Lemma 5.8.11. Let \( \tilde{x} \in X(A, b) \) be the maximum solution and \( x \in X(A, b) \). If \( x_i \) is binding in the \( j \)-th equation then \( \tilde{x}_i \) is also binding in the \( j \)-th equation.

Proof. For \( x \in X(A, b) \), \( \max_{i \in I} \left( \frac{x_i + a_{ij}}{1-x_i a_{ij}} \right) = b_j, \forall j \in J \). Also \( \frac{x_i + a_{ij}}{1-x_i a_{ij}} \) is increasing function in \( x_i \) with maximum value \( b_j \). This implies that \( \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} \leq b_j, \forall j \in J \). Now if \( x_i \) is binding variable in the \( j \)-th equation, then \( \frac{x_i + a_{ij}}{1-x_i a_{ij}} = b_j \) holds for \( j \in J \). Since \( x_i \leq \tilde{x}_i \), therefore \( b_j = \frac{x_i + a_{ij}}{1-x_i a_{ij}} \leq \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} \leq b_j \). This implies that \( b_j = \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} \leq b_j \). Therefore \( \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} = b_j \) and \( \tilde{x}_i \) is also binding in the \( j \)-th equation.

Theorem 5.8.12. Let \( \tilde{x} \in X(A, b) \) be the maximum solution. For any solution, \( x \in X(A, b) \), if \( x_i \) is a binding variable then \( x_i = \tilde{x}_i \).

Proof. For \( x \in X(A, b) \), \( \max_{i \in I} \left( \frac{x_i + a_{ij}}{1-x_i a_{ij}} \right) = b_j, \forall j \in J \). Since \( x_i \) is a binding variable, thus \( \frac{x_i + a_{ij}}{1-x_i a_{ij}} = b_j \) for some \( j \in J \). From Lemma 5.8.11, since \( x_i \) is binding, thus \( \tilde{x}_i \) is also binding, therefore \( \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} = b_j \). Now, let \( x_i < \tilde{x}_i \). Thus, \( b_j = \frac{x_i + a_{ij}}{1-x_i a_{ij}} < \frac{\tilde{x}_i + a_{ij}}{1-\tilde{x}_i a_{ij}} = b_j \), a contradiction. Therefore, if \( x_i \) is a binding variable then \( x_i = \tilde{x}_i \).
5.9 0-1 integer programming and genetic algorithm to solve problem II

The equivalent 0-1 integer programming problem of the linear programming problem (5.19)-(5.20) is defined as

\[
\text{Min } \bar{Z} = \sum_{i \in I} c_i y_i \bar{x}_i
\]

\text{s.t. } \sum_{i \in I_j} y_i \geq 1, \forall j \in J

(5.21)

(5.22)

\[ y_i \in \{0,1\}, \forall i \in I \]

Note that the constraints of the problem (5.21)-(5.22) require that, \( \forall j \in J \), there exists at least one \( i \in I_j \), such that \( y_i = 1 \). The theorem below shows that solving problem (5.19)-(5.20) is equivalent to solving 0-1 integer programming problem (5.21)-(5.22).

**Theorem 5.9.1.** Let \( \bar{X}(A,b) = \{ y = (y_i)_{i \in I} | y_i \in \{0,1\}, \forall i \in I \} \) be the set of feasible solutions of problem (5.21)-(5.22) and \( X(A,b) \) be the set of feasible solutions of problem (5.19)-(5.20). For each \( x \in X(A,b) \), there exists \( y \in \bar{X}(A,b) \), and vice-versa, such that \( Z(x) = \bar{Z}(y) \).

**Algorithm 4: For converting the problem into equivalent 0-1 integer programming problem**

Step 1: Get the matrix \( A, b, c = [c_1, c_2, \ldots, c_m]^T \). Compute the maximum solution \( \bar{x} \). If \( \bar{x} \cdot A = b \), problem is feasible. If problem is infeasible, stop.

Step 2: Find \( I_j(\bar{x}) = \{ i \in I | \frac{\bar{x}_i + a_{ij}}{1 - \bar{x}_i a_{ij}} = b_j \}, \forall j \in J \).

Step 3: Convert the problem to equivalent 0-1 integer programming problem.
Algorithm 5: For initialization of the binary coded genetic algorithm

The algorithm follows on same lines as Algorithm 3 in Section 5.6.

Example 5.9.2. Consider the following optimization problem subject to max-* fuzzy relation equations with $x^*a = \frac{x+a}{1-xa}$, 

$$\text{Min } Z(x) = x_1 - 2x_2 + 3x_3 - 0.5x_4 + 4.4x_5 + 1.5x_6 + 2x_7$$

subject to $x^*A = b$, $0 \leq x_i \leq 1$, $i = 1, 2, ..., 7$

First we apply Algorithm 4 to convert the problem into equivalent 0-1 integer programming problem.

Step 1: $\tilde{x} = [0.2339 \ 0.1611 \ 0.1079 \ 0.0661 \ 0.1613 \ 0.0745 \ 0.0769]$. Feasibility of the solution is checked. Since $\tilde{x}^*A = b$, problem is feasible.

Step 2: $I_1 = \{1, 2, 5\}$, $I_2 = \{1, 3, 5\}$, $I_3 = \{2, 5, 6, 7\}$, $I_4 = \{2, 3, 7\}$, $I_5 = \{3, 4, 6\}$.
Step 3: The optimization problem can be converted into an equivalent 0-1 integer programming problem as:

\[
\begin{align*}
\text{Min } Z &= y_1\bar{x}_1 - 2y_2\bar{x}_2 + 3y_3\bar{x}_3 - 0.5y_4\bar{x}_4 + 4.4y_5\bar{x}_5 + 1.5y_6\bar{x}_6 + 2y_7\bar{x}_7 \\
\text{s.t.} & \quad y_1 + y_2 + y_3 \geq 1 \\
& \quad y_1 + y_3 + y_5 \geq 1 \\
& \quad y_2 + y_3 + y_6 + y_7 \geq 1 \\
& \quad y_2 + y_3 + y_7 \geq 1 \\
& \quad y_3 + y_4 + y_6 \geq 1 \\
& \quad y_i \in \{0, 1\}, \forall i = 1, 2, \ldots, 7
\end{align*}
\]

Now Algorithm 5 is applied to the equivalent 0-1 integer programming problem of the linear problem.

Step 1: Initialize number of generations, say, \( g = 50 \) and number of populations, say \( k = 15 \).

Step 2: Set generation \( g = 1 \).

Step 3: Matrix \( R_{m \times k} \) is obtained as:

\[
R = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]
Step 4: Check for each column of $R$, whether it satisfies the constraints of the equivalent 0-1 integer programming problem (in Step 3) or not. Note that only 1st, 5th, 6th, 7th, 9th, 10th, 11th, 12th, 13th and 15th column vectors (solution vectors) of the matrix $R$ satisfy all the constraints. Updated matrix $R$ is obtained as follows:

$$R = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}$$

Step 5: Each column of $R$ represents a solution vector to the equivalent 0-1 integer programming problem. Once a matrix of solution vectors is obtained, it is necessary to evaluate the solution, particularly in the context of the underlying objective function.

$$T = \begin{bmatrix}
0.2339 & 0 & 0 & 0.2339 & 0 & 0.2339 & 0.2339 & 0 & 0 & 0 \\
0 & 0 & 0.1611 & 0 & 0.1611 & 0.1611 & 0 & 0.1611 & 0.1611 & 0.1611 \\
0.1079 & 0 & 0.1079 & 0.1079 & 0.1079 & 0.1079 & 0.1079 & 0.1079 & 0 & 0.1079 \\
0 & 0.0661 & 0 & 0.0661 & 0.0661 & 0 & 0.0661 & 0.0661 & 0 & 0 \\
0.1613 & 0.1613 & 0.1613 & 0.1613 & 0 & 0 & 0 & 0 & 0.1613 & 0.1613 & 0.1613 \\
0.0745 & 0.0745 & 0.0745 & 0 & 0.0745 & 0 & 0.0745 & 0.0745 & 0.0745 & 0 \\
0 & 0.0769 & 0 & 0 & 0.0769 & 0.0769 & 0 & 0.0769 & 0 & 0
\end{bmatrix}$$

Step 6:

$M_{objective} = [1.3792 \ 0.9422 \ 0.8231 \ 1.2343 \ 0.2342 \ 0.3894 \ 0.6364 \ 0.9439 \ 0.4993 \ 0.7113]$

Step 7: Assign objective function value $Z = 0.2342$, the minimum value from $M_{objective}$.

Step 8-Step 10: For this example, a binary-coded genetic algorithm was run for 50 iterations and selection, crossover and mutation operators were applied. The optimal
value of the objective function was found to be -0.12135 and the corresponding solution vector for the 0-1 integer programming problem from the population $R$ was obtained as $[1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T$, therefore the optimal solution to the linear optimization problem is $x^* = [0.2339 \ 0.1611 \ 0 \ 0.0661 \ 0 \ 0 \ 0]$. 

**Example 5.9.3.** Consider the following optimization problem subject to max-* fuzzy relation equations with $x^* a = \frac{x + a}{1 - xa}$,

$$\text{Min } Z(x) = -3x_1 + x_2 + 2x_3 + 4x_4 - x_5 + x_6 - 2x_7 + 2.5x_8 + 1.8x_9$$

subject to $x^\top A = b$, $0 \leq x_i \leq 1$, $i = 1, 2, \ldots, 9$

\[A = \begin{bmatrix} 0.6500 & 0.6800 & 0.6500 & 0.7500 & 0.3200 & 0.7000 \\ 0.6000 & 0.3000 & 0.6800 & 0.1100 & 0.1200 & 0.6500 \\ 0.6000 & 0.7760 & 0.3600 & 0.7100 & 0.5800 & 0.6000 \\ 0.6500 & 0.7500 & 0.2200 & 0.7500 & 0.3500 & 0.5500 \\ 0.6000 & 0.2500 & 0.3400 & 0.7000 & 0.5000 & 0.5500 \\ 0.2800 & 0.3200 & 0.5500 & 0.2400 & 0.4617 & 0.5500 \\ 0.4000 & 0.1000 & 0.4800 & 0.3000 & 0.4800 & 0.4500 \\ 0.6064 & 0.7500 & 0.3300 & 0.6500 & 0.5500 & 0.3000 \\ 0.5200 & 0.6907 & 0.6000 & 0.6000 & 0.2000 & 0.6000 \end{bmatrix}\]

\[b = [0.6500 \ 0.8000 \ 0.7000 \ 0.7500 \ 0.6000 \ 0.7000] \]

Convert the problem into equivalent 0-1 integer programming problem by applying the Algorithm 4.

**Step1:** $\tilde{x} = [0 \ 0.0136 \ 0.0148 \ 0 \ 0.0328 \ 0.1083 \ 0.0932 \ 0.0313 \ 0.0704]$. 

Feasibility of the solution is checked. Since, $\tilde{x}^\top A = b$, problem is feasible.
Step 2: \( I_1 = \{1, 4, 8\}, \ I_2 = \{3, 8, 9\}, \ I_3 = \{2, 6, 9\}, \ I_4 = \{1, 4, 5\}, \ I_5 = \{3, 6, 7\}, \ I_6 = \{1, 6, 9\}. \)

Step 3: The optimization problem can be converted into an equivalent 0-1 integer programming problem as:

Min \( Z = -3y_1\tilde{x}_1 + y_2\tilde{x}_2 + 2y_3\tilde{x}_3 + 4y_4\tilde{x}_4 - y_5\tilde{x}_5 + y_6\tilde{x}_6 - 2y_7\tilde{x}_7 + 2.5y_8\tilde{x}_8 + 1.8y_9\tilde{x}_9 \)

s.t. \( y_1 + y_4 + y_8 \geq 1 \)
\( y_3 + y_6 + y_9 \geq 1 \)
\( y_2 + y_6 + y_9 \geq 1 \)
\( y_1 + y_4 + y_3 \geq 1 \)
\( y_3 + y_6 + y_7 \geq 1 \)
\( y_1 + y_6 + y_9 \geq 1 \)
\( y_i \in \{0, 1\}, \ \forall \ i = 1, 2, \ldots, 9 \)

Now Algorithm 5 is applied to the equivalent 0-1 integer programming problem of the linear problem.

Step 1: Initialize number of generations, say, \( g = 50 \) and number of populations, say \( k = 15 \).

Step 2: Set generation \( g = 1 \).

Step 3: Matrix \( R_{mxk} \) is obtained as:
Step 4: Check for each column of $R$, whether it satisfies the constraints of the equivalent 0-1 integer programming (in Step 3) problem or not. Note that only 2nd, 3rd, 4th, 7th, 8th, 9th, 10th, 11th, 13th, 14th and 15th column vectors (solution vectors) of the matrix $R$ satisfy all the constraints. Updated matrix $R$ is obtained as follows:

$$ R = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
\end{bmatrix} $$

Step 5: Each column of $R$ is a solution vector to the equivalent 0-1 integer programming problem. Once a matrix of solution vectors is obtained, it is necessary to evaluate the solution, particularly in the context of the underlying objective function.
Step 6:
\[ M_{\text{objective}} = \begin{bmatrix} 0.2646 & 0.0622 & 0.2782 & 0.0322 & -0.0813 & 0.0297 & -0.0677 & 0.2482 & 0.0622 & 0.1565 & -0.0813 \end{bmatrix} \]

Step 7: Assign objective function value \( Z = -0.0813 \), the minimum value from \( M_{\text{objective}} \).

Step 8-Step 10: For this example, a binary-coded GA was run for 50 iterations and selection, crossover and mutation operators were applied. The optimal value of the objective function was found to be -0.1760 and the corresponding solution vector for the 0-1 integer programming problem from the population \( R \) was obtained as \( [1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T \), therefore the optimal solution to the linear optimization problem is \( x^* = [0 \ 0.0136 \ 0.0148 \ 0 \ 0.0328 \ 0 \ 0.0932 \ 0 \ 0] \).

5.10 Conclusion

This chapter discusses linear optimization problem with max-Archimedean and max-* fuzzy relation equations as constraints. For the linear optimization problem with max-Archimedean fuzzy relation equations as constraints, optimal solution is obtained by using two methods. Firstly, using the concept of covering problem, concept of essential, non-essential variables is seen and it is shown that the irredundant covering corresponds to the minimal solutions of the problem. The solution of any system of equations with max-Archimedean composition is a covering problem in pure form. The optimal solution of the problem is achieved using algebraic method. Secondly, since such problems, in general, are combinatorial in nature, thus they belong to the class of \( NP \)-hard problems.
A reduction procedure is required. The problem is reduced using the concept of covering. The reduced problem is represented as an equivalent 0-1 integer programming problem and binary coded genetic algorithm is developed to solve it. The population is initialized first and then genetic operators’ selection, crossover, and mutation are applied. Optimal solution appears after a predefined number of generations. There is no need to divide the optimization problem into two sub-problems according to negative and positive coefficients in the objective function. If there are alternate optimal solutions, then the algorithm gives alternate optimal solutions and the underlying value of the objective function. It was observed that for mostly all the problems the optimal solution was found before 50 generations. Number of iterations increases with the size of the problem.

For the linear optimization problem with max-* fuzzy relation equations as constraints, solvability conditions for existence of solution are established and the problem is represented as an equivalent 0-1 integer programming problem solvable by binary coded genetic algorithm.