CHAPTER 6
Resolving posynomial geometric programming restricted to a system of fuzzy relational equations

6.1 Introduction

The notion of fuzzy relation equations was first investigated by Sanchez [155] in 1976 and then extended by many others [2-4,11,49,102]. The most fundamental type of fuzzy relation equations are based on sup-$T$ composition, where $T$ is typically a continuous triangular norm among which the minimum $T_m$ is the most frequently used one. The solvability criteria of sup-$T$ equations were first established by Sanchez [155] for sup-$T_m$ equations and then extended by others [49,102]. The structure of the complete solution set of sup-$T_m$ equations was first characterized by Sanchez [156] and generalized to sup-$T$ equations by Di Nola et al. [11,21,22,49]. An excellent literature over the origination of the fuzzy relation equation is present in [84]. It has been well established that the complete solution set of a consistent finite system of sup-$T$ equations can be determined by unique maximum solution and a finite number of minimal solutions. The consistency of a system of sup-$T$ equations can be easily verified by checking the potential maximum solution. Markowski [95] showed that when the triangular norm is Archimedean, the minimal solutions correspond one-to-one to the irredudant coverings of a set covering problem but the relation between max-min fuzzy relation equations and the covering problem is more complex. Further, Peeva [135] proposed a method to obtain all the minimal solutions of max-min fuzzy relation equations.

Geometric programming is a class of nonlinear, non-convex optimization problems with objective function and constraints in a special form. Duffin, Zener and Peterson [25] were first to propose geometric programming theory in 1961. In recent years, geometric
programming has been applied in a wide variety of fields as they occur naturally in most of the design and planning problems [12, 201].

Fuzzy relational geometric programming come forwards as a new branch of fuzzy optimization and has received much attention in recent years. This theory has been applied in power system, environmental engineering, and economic management and so on with the range of its application expected to be growing exponentially in future [201].

Yang and Cao [197-201] have made significant contribution to the area of fuzzy relational geometric programming with variants of geometric objectives using a variety of fuzzy operators. Yang and Cao [199] considered a monomial geometric objective with the max-min system of fuzzy relation constraints. A solution method based on the path matrix was used to determine all the minimal solutions of the system. The optimal solution was determined using the maximal solution and minimal solutions of the system. After that Yang and Cao [200] extended the work to posynomial objective, when the exponents and coefficients of objective function are fuzzy numbers. A method using bi-level programming, able to find the membership function of the fuzzy objective optimal value was developed.

Wu [189] also studied a problem with latticized geometric objective function subject to max-min fuzzy relation equations as constraints and gave a reduction procedure for solving the problem. Recently Zhou and Ahat [217] considered a geometric programming problem with a system of max-product fuzzy relational equations as constraints and gave an efficient procedure to find the optimal solution.

This chapter considers a posynomial geometric optimization problem subjected to a system of max-min fuzzy relational equations (FRE) constraints. A two stage procedure has been suggested to solve the problem. Firstly, the system of fuzzy relational equations is solved. The optimization problems are modeled in the same number as many minimal solutions are obtained. An optimal solution for the problem is determined after solving these optimization problems.
6.2 The problem

Consider the following system of fuzzy relational equations

\[ A \circ x = b \] (6.1)

where \( A = [a_{ij}], 0 \leq a_{ij} \leq 1 \) be a \( m \times n \) dimensional fuzzy matrix and right hand matrix \( b = (b_1, b_2, \ldots, b_m), 0 \leq b_i \leq 1 \) be a \( m \) dimensional vector and “\( \circ \)” denotes max-\( \odot \) composition of \( x \) and \( A, \odot \) is the t-norm operator from the Godel algebra \( ([0,1], \lor, \land, \odot, \Theta, 0,1) \). Let \( I = \{1,2,\ldots,m\} \) and \( J = \{1,2,\ldots,n\} \) be the index sets. Given fuzzy relation matrix \( A \), output vector \( b \), and relation (6.1), the resolution problem is to determine all input vectors \( x = [x_1, x_2, \ldots, x_n]^T \) such that:

\[ \max_{j=1}^{n} \min_{i=1}^{m} (a_{ij}, x_j) = b_i \quad \forall i = 1,2,\ldots,m \]

We are interested in solving the following fuzzy relational geometric optimization problem:

\[ \min f(x) \] (6.2)

s.t. \( \max_{j=1}^{n} \min_{i=1}^{m} (a_{ij}, x_j) = b_i \quad \forall i = 1,2,\ldots,m \)

\[ 0 \leq x_j \leq 1, \quad j \in J \]

where \( f(x) \) is the geometric function of \( x \) defined as \[ f(x) = \sum_{k=1}^{K} f_k(x) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} x_j^{r_{jk}}, \]

\( f_k(x) \) represents the \( k^{th} \) monomial in \( x \) and each coefficient \( c_k > 0 \), \( r_{jk} \in \mathbb{R} \) \( (0 < k \leq K, 1 \leq j \leq n) \) is exponent of variable \( x_j \) in the \( k^{th} \) monomial and \( x = [x_1, x_2, \ldots, x_n]^T \) is the solution vector. In this optimization problem the objective function as well as feasible domain is also non-convex, so the optimization problem can be categorized as a non-
convex programming problem. This characteristic of problem offers difficulty in employing the traditional methods for nonlinear programming to be applied directly to solve this optimization problem.

The feasible domain of the problem is given by \( X(A, b) = \{ x \in [0, 1]^n | A \circ x = b \} \) that has been well characterized in chapter 4 in section 4.3. A system of sup-\( \odot \) equations \( A \circ x = b \) defined in (6.1) is said to be in the normal form if its right side elements \( b_i \)'s have a non-increasing order, i.e. \( b_1 \geq b_2 \geq \ldots \geq b_m \). The systems of sup-\( \odot \) equations having same normal form have same solutions set and hence are considered to be equivalent [74]. Any system of fuzzy relational equations can be converted into its normal form in polynomial time. The reduction of system (6.1) to its normal form possibly leads to reduction of computations.

**Definition 6.2.1.** Let \( \odot \) be a continuous \( t \)-norm then there exists a unique operation \( \Theta \), associated with \( \odot \) called as the implication operator defined as:

\[
a \Theta b = \text{Sup} \{ x \in [0, 1] | (a \odot x) \leq b \}
\]

The \( \Theta \), operator satisfies the following two properties:

1. \( a \odot (a \Theta b) \leq b \) \hspace{1cm} (6.3.1)
2. \( (a \odot x) \leq b \) iff \( x \leq a \Theta b \) \hspace{1cm} (6.3.2)

If the system \( A \circ x = b \) is consistent, the maximum solution of (6.1) can be determined explicitly using the analytic expression as follows:

\[
\tilde{x} = A \Theta b = \left[ \bigwedge_{j \in I} (a_{ij} \Theta b_j) \right]_{i \in J}
\]

130
Here \( a_j \Theta b_j = \begin{cases} 1, & \text{if } a_j \leq b_j \\ b_j, & \text{otherwise} \end{cases} \)

When \( X(A,b) \neq \emptyset \), the solution set is given by

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

where \( \bar{X}(A,b) \) is the set of all minimal solutions of (6.1).

**Lemma 6.2.1.** If in the \( i^{\text{th}} \) equation \( a_{ij} < b_j, \forall j \in J \), then the solution set \( X(A,b) = \emptyset \).

**Proof.** If in the \( i^{\text{th}} \) equation \( a_{ij} < b_j \) holds for all \( j \in J \), then for \( x_j \neq a_{ij}, \min(a_{ij}, x_j) \leq \min(a_{ij}, 1) = a_{ij} < b_j \) and for \( x_j = a_{ij}, \min(a_{ij}, x_j) \leq \min(a_{ij}, a_{ij}) = a_{ij} < b_j \). Thus, for both cases, \( \min(a_{ij}, x_j) < b_j, \forall j \in J \). Hence, \( \max_{j \in J} \min(a_{ij}, x_j) < b_i \) which implies that the \( i^{\text{th}} \) equation remains dissatisfied by any variable then the system has no solution i.e., \( X(A,b) = \emptyset \).

**Definition 6.2.2.** A continuous \( t \)-norm \( \circ \) is said to be an Archimedean \( t \)-norm if and only if \( x \circ x < x \) for all \( 0 < x < 1 \) and non-Archimedean otherwise. It is clear that the minimum \( t \)-norm \( T_m \) is continuous and non-Archimedean in nature.

**Lemma 6.2.2.** A vector \( x \in X(A,b) \) is a solution of system (6.1) if and only if for each \( i \in I \), \( \exists j_i \in J \) such that \( \min(a_{ij}, x_{j_i}) = b_i \) and \( \min(a_{ij}, x_{j_i}) \leq b_i, i \in I, j \in J \).

**Proof.** For \( x \in X(A,b) \), \( \max_{j \in J} \min(a_{ij}, x_j) = b_i, \forall i \in I \). This implies \( \min(a_{ij}, x_j) \leq b_i, \forall j \in J \). Therefore, in order to satisfy the equality constraint, there must exist at least one \( j_i \in J \) such that \( \min(a_{ij}, x_{j_i}) = b_i, \forall i \in I \).

A system of fuzzy relational equations defined in (6.1) is said to be homogeneous if \( b = 0 \), and non-homogeneous otherwise. A homogeneous system always has the trivial
solution. In the system $A \circ x = b$ it holds that $\max_{j \in J} \min(a_{ij}, x_j) = b_i, \forall i \in I$. If $\exists b_i = 0$ for some $i \in I$ such that $\exists a_{ij} > 0$, for $j \in J, i \in I$ then $x_j$ has to be zero. Hence, for some $b_i = 0$ if $a_{ij} > 0, j \in J$ we can simply set $x_j = 0$. So the value of all the variables that appear in the $i^{th}$ equation must be 0. Now the system can be reduced and simplified to a new system after deleting all such equations from $A$ and corresponding component from $b$. At the end any solution of the original system can be constructed by simply setting variables $x_j$ to zero wherever $a_{ij} > 0, j \in J$.

### 6.3 Reduction procedure for system (6.1)

Consider the single equation from the system (6.1) given as:

$$(a_{i1} \land x_1) \lor (a_{i2} \land x_2) \lor \ldots \lor (a_{in} \land x_n) = b_i$$

$$0 \leq x_j \leq 1, j \in J$$

By property (6.3.2) of implication operator, $\min(a_{ij}, x_j) \leq b_i$ iff $x_j = (a_{ij} \Theta b_i)$. In this equation, the unit component equation $\min(a_{ij}, x_j) = b_i$ has a solution iff $b_i \leq a_{ij}$. Considering this, the solution set of $\min(a_{ij}, x_j) = b_i$ can be discussed in the following cases:

(i). Case I: If $a_{ij} > b_i$ then we have $x_j = b_i$ as the solution of the unit equation $\min(a_{ij}, x_j) = b_i$.

(ii). Case II: If $a_{ij} = b_i$ then $[b_i, a_{ij} \Theta b_i]$ is the solution set of the unit equation $\min(a_{ij}, x_j) = b_i$. 

132
Case III: If \( a_j < b_i \) then we have \( x_j = \emptyset \), i.e. equation \( \min(a_j, x_j) = b_i \) has no solution in this case.

The above discussions show that the equation \( \min(a, x) = b \) has a solution iff \( b \leq a \) and then the solution set of \( \min(a, x) = b \) is given by \([b, a\Theta b]\). For more details see [135,136].

With the help of computed maximum solution \( \tilde{x} \) and the discussion in above cases the characteristic matrix \( \tilde{P} = (\tilde{p}_{ij})_{\text{mea}} \) of the system \( A \circ x = b \) is defined as:

\[
\tilde{p}_{ij} = \begin{cases} 
[b_i, \tilde{x}_j], & \text{if } \min(a_j, \tilde{x}_j) = b_i \\
\phi, & \text{otherwise}
\end{cases}
\] (6.5)

It is clear from the above argument on the unit equation \( \min(a_j, x_j) = b_i \) that \( b_i \) presents the lower bound for a variable \( x_j \) to satisfy the \( i^{th} \) equation. Each nonempty element \( \tilde{p}_{ij} \) of the characteristic matrix \( \tilde{P} \) gives the whole range of possible values for the variable \( x_j \) to satisfy the \( i^{th} \) equation. The system \( A \circ x = b \) is consistent if and only if \( \tilde{P} \) has no row with all elements as empty elements i.e. if there does not exist some equation not satisfied by any variable. In general, for a continuous non-Archimedean t-norm the non-empty elements in characteristic matrix \( \tilde{P} \) might not be singleton while in the case of an Archimedean norm these entries are always singleton and are equal to the corresponding component of maximal solution.

Now if \( \exists b_i = 0, i \in I \) then without loss of generality corresponding row can be removed from the matrix \( \tilde{P} \). After removing such rows from matrix \( \tilde{P} \) the matrix is simplified to matrix \( P \). In the simplified matrix \( P \) there might be some variables that satisfy only those equations for which \( b_i = 0 \). After removing these rows from \( \tilde{P} \), the columns
corresponding to those variables have only empty elements in the characteristic matrix $P$. Such variables are called pseudo-essential [84].

**Definition 6.3.1.** A variable is said to be multi-essential if its corresponding column in the characteristic matrix $P$ contains a non-singleton element. A multi-essential variable has the characteristic; that it can attain a value other than 0 and corresponding maximal component value in minimal solutions. Such variables can satisfy different no. of equations with different values.

**Definition 6.3.2.** Let $P = (p_{ij})_{m \times n}$ be the simplified characteristic matrix of system (6.1) then a row $i_1$ dominates a row $i_2$ if $p_{i_1j} \neq \phi$ imply $p_{i_1j} \subseteq p_{i_2j}$ for all $j \in J$.

A row of $P$ is redundant if and only if it dominates some other row. Moreover, if a row $i_1$ dominates a row $i_2$, we have $b_{i_1} \leq b_{i_2}$ since $b_i$ represents the lower bound for a variable to satisfy the $i^{th}$ equation. Here the changing the system to its normal form plays important role. For the sake of simplification, redundant rows can be removed from the modified matrix $P$. At this stage some columns can have all empty elements. The variables of such columns cannot have non-zero values in minimal solutions. Such variables are called semi-essential [95]. After removing the redundant rows and the columns corresponding to semi-essential variables from matrix $P$, the matrix transforms to the matrix $P'$. It is noteworthy that if there exists a row in $P'$ having the unique nonempty entry corresponding to a column $j$, then the variable corresponding to that column is called super-essential [95]. The super-essential variable in this case is different from that as discussed in chapter 4. In the system with non-Archimedean based composition the super-essential variable can also attain some value other than its corresponding maximal component value while in case of Archimedean norm based composition they coincide with the corresponding component value in the maximal solution. If $x_j$ is super-essential, it can assume different values in different minimal solutions of system (6.1).
Once the matrix $P'$ is obtained, we adopt the algebraic method for finding all minimal solutions by considering the simplified characteristic matrix $P'$ associated with the formal logical expression. We follow the notations as used in [136]. Each row of matrix $P'$ is associated with logical sum $u_i = \bigvee_j \left( \frac{b_j}{x_j} \right)$ (DNF) for all $j \in J'$ where $J' = \{ j \in J \mid p_j \neq \emptyset \} \forall i \in I$. The whole matrix $P'$ corresponds with the logical product $P' = \bigwedge_i u_i$ (CNF). The truth function obtained in (CNF) can be reduced to DNF using laws of conversion of fuzzy truth function from CNF to DNF as defined in section 6.3.1. The truth function for finding all the minimal solutions can be given as:

$$F_{P'} = \bigwedge_{i \in P'} \bigvee_{j \in J'} \left( \frac{b_j}{x_j} \right)$$

### 6.3.1 Rules to perform conversion from fuzzy truth function in CNF to DNF

1. \[
\left( \frac{b_{i_1}}{x_{i_1}} \right) \left( \frac{b_{i_2}}{x_{i_2}} \right) = \left\{ \begin{array}{ll}
\left( \frac{b_{i_1} \lor b_{i_2}}{x_{i_1}} \right) & \text{if } j_1 = j_2 \\
\left( \frac{b_{i_1}}{x_{i_1}} \right) & \text{otherwise}
\end{array} \right.
\]

2. \[
\left( \frac{b_{i_1}}{x_{i_1}} \right) \left( \frac{b_{i_2}}{x_{i_2}} \right) = \left( \frac{b_{i_1}}{x_{i_1}} \right) \left( \frac{b_{i_2}}{x_{i_2}} \right), \quad \text{(commutative)}
\]

3. \[
\left( \frac{b_{i_1}}{x_{i_1}} \right) \left( \frac{b_{i_2}}{x_{i_2}} \right) \lor \left( \frac{b_{i_2}}{x_{i_2}} \right) = \left( \frac{b_{i_1}}{x_{i_1}} \right) \left( \frac{b_{i_2}}{x_{i_2}} \right) \lor \left( \frac{b_{i_2}}{x_{i_2}} \right) \lor \left( \frac{b_{i_3}}{x_{i_3}} \right), \quad j_1, j_2, j_3 \in J' \quad \text{(distributive)}
\]

4. \[
\left( \frac{b_{i_1}}{x_{i_1}} \right) \ldots \left( \frac{b_{i_t}}{x_{i_t}} \right) \lor \left( \frac{b_{i_1}}{x_{i_1}} \right) \ldots \left( \frac{b_{i_t}}{x_{i_t}} \right)
\]

\[
= \left( \frac{b_{i_t}}{x_{i_t}} \right) \ldots \left( \frac{b_{i_t}}{x_{i_t}} \right) \quad \text{if } b_t \leq b_i, \quad t = 1, 2, \ldots, m
\]

\[
\text{unchanged \quad otherwise}
\]
The whole procedure for finding all the minimal solutions of system (6.1) can be summarized in the Algorithm 1 given below:

**Algorithm 1: Finding all the minimal solutions of (6.1)**

Step 1: Get the matrices $A, b$.
Step 2: Find the maximal solution $\tilde{x}$ by (6.4).
Step 3: Check consistency of the system (6.1). If $A \circ \tilde{x} \neq b$, system is inconsistent, stop the procedure.
Step 4: Find the characteristic matrix $\tilde{P}$ of $A$ using (6.5) and then find simplified characteristic matrix $P$.
Step 5: Find the reduced matrix $P'$.
Step 6: Find all the minimal solutions of (6.1) by applying logical rules of inferences defined in section 6.3.1.

Once the solution set of system (6.1) is determined by Algorithm 1, we construct as many optimization problems as many minimal solutions are obtained considering each of the convex sub-feasible region formed by one minimal solution and unique maximal solution as the feasible region. Then the optimal solution $x^*$ of the original optimization problem in (6.2) is obtained by solving the above optimization problems i.e.

$$f(x^*) = \min(f_r(x^*_r)) \quad , \quad r = 1, 2, ..., |\tilde{X}(A, b)|$$

where $|\tilde{X}(A, b)|$ is the cardinality of set of minimal solutions of system (6.1) and $x^*_r$ is the optimal solution of the $r^{th}$ optimization problem formed. The reduced sub-optimization problems thus formed are shown in figure 6.1 below.
6.4 Solving reduced optimization problems

In the past few years, the application of genetic algorithm for numerical optimization has accelerated in times. In case of nonlinear optimization it is impractical to obtain the global optimal solution. Genetic algorithms offer the ease in finding good converging solutions of nonlinear optimization problems with least mathematical computation effort. The efficient design of the algorithm decides its effectiveness to solve the considered problem. Once the optimization problems have been formed in the previous section, a specific genetic algorithm is applied to solve the different optimization problems thus formed. The design of the genetic algorithm used can be described in the following steps:

6.4.1 Initialization

For initialization of population generate an initial population of fixed size with the super-essential variables assuming fixed values and assigning remaining variables a value in a range obtained for that variable. The generated population is examined for feasibility.
The feasible population is then evaluated and then undergoes the process of selection and recombination.

### 6.4.2 Selection and recombination

The main purpose of selection is to maintain good copies of individuals for the next generation. We have used the *tournament selection* as the selection strategy. Tournament selection starts by selecting a set of individuals at random and tournaments are played among them and the best fit player wins and chosen for mating. Similar process is repeated until a population of desired strength is selected. The number of players in a set denotes the tournament size. Bigger tournament size enhances the selection pressure so small tournament size is always a good choice. To ease the procedure a tournament size 3 has been used for selection. Selected individuals undergo crossover and mutation and new individuals are obtained.

The arithmetic crossover and mutation operator as designed in section 4.5 of the chapter 4 have been applied. They provide feasible solutions at the end that pass on to the next generation. The procedure keeps running until some pre-specified terminating criterion is not met. In general, stopping criterion is the no. of generations fixed in prior.

The overall procedure applied for solving the considered optimization problems can be summarized in the following algorithm:

**Algorithm 2: Genetic Algorithm procedure**

Step 1: Define maximum number of iterations as gen_max and initialize the iteration number as gen =1.

Step 2: Randomly generate initial population of size say $k$ within the specified bounds of the decision variables for that particular optimization problem.

Step 3: Check feasibility of solutions.
Step 4: If feasible individual(s) are found go to step 5 else go to step 2.

Step 5: Select the best individuals using objective function formed for that problem.

Step 6: Generate offsprings for next generations by applying crossover and mutation operations defined in section 4.5 of chapter 4.

Step 7: Select feasible individuals among offsprings and set gen = gen + 1.

Step 8: If better offsprings than parents are found go to step 9 else discard the offsprings and parent population remains same for the next generation also and now go to step 10.

Step 9: Update the solution and value of objective function and now this generation becomes the parent generation.

Step 10: Is gen equal to gen_max? If yes, then stop else go to step 5.

On the whole, the summarized procedure applied for solving the considered optimization problem (6.2) can be described in the following steps:

**Algorithm 3**

Step 1: Find all the minimal solutions by the Algorithm 1.
Step 2: Find all the sub-feasible regions considering all the minimal solutions.
Step 3: Solve the different optimization problems obtained using Algorithm 2.
Step 4: Find single optimal solution after solving these different sub-optimization problems formulated.
6.5 Illustrative example

Example 6.1. Consider the following posynomial fuzzy relational optimization problem:

Min \( f(x) = 5x_1^{-0.2}x_2^{-0.3}x_3x_4^{-1}x_5 + 2x_1^{-0.2}x_2^{-1.5}x_3^2x_4^{-2}x_5^1 \)

s.t \( A \circ x = b \), \( 0 \leq x_j \leq 1 \), \( i = 1,2,...,6 \), \( j = 1,2,...,5 \) where,

\[
A = \begin{bmatrix}
0.9 & 0.8 & 0.6 & 0.3 & 0.9 \\
0.8 & 0.7 & 0.8 & 1 & 0.8 \\
0.6 & 0.9 & 0.8 & 0.9 & 0.5 \\
0.4 & 0.2 & 0.5 & 0.6 & 0.2 \\
0.3 & 0.3 & 0.5 & 0.2 & 0.1 \\
0.4 & 0.1 & 0.2 & 0.3 & 0.5 \\
\end{bmatrix}, \quad b = [0.8 \ 0.8 \ 0.8 \ 0.5 \ 0.5 \ 0.4]^T
\]

Maximal solution of computed by (6.4) come out to be \( \tilde{x} = (0.8 \ 0.8 \ 1 \ 0.5 \ 0.4)^T \).

Since \( A \circ \tilde{x} = b \) the system of FRE is consistent.

The characteristic matrix \( \hat{P} \) of system (6.1) is computed as:

\[
\hat{P} = \begin{bmatrix}
x_1 & x_2 & x_3 & x_4 & x_5 \\
0.8 & 0.8 & \phi & \phi & \phi \\
0.8 & \phi & [0.8,1] & \phi & \phi \\
\phi & 0.8 & [0.8,1] & \phi & \phi \\
\phi & \phi & [0.5,1] & 0.5 & \phi \\
\phi & \phi & [0.5,1] & \phi & \phi \\
0.4 & \phi & \phi & \phi & 0.4 \\
\end{bmatrix}
\]

As row 4, dominates row 5, row 4 is redundant and can be removed from \( \hat{P} \). After removing these rows, column corresponding to \( x_4 \) is emptied out. Hence, \( x_4 \) is semi-
essential. After removing the redundant rows the simplified matrix $P$ is obtained as follows:

$$P = \begin{bmatrix}
0.8 & 0.8 & \phi & \phi & \phi \\
0.8 & \phi & [0.8,1] & \phi & \phi \\
\phi & 0.8 & [0.8,1] & \phi & \phi \\
\phi & \phi & [0.5,1] & \phi & \phi \\
0.4 & \phi & \phi & 0.4
\end{bmatrix}$$

After removing the all empty columns, we obtain the matrix $P'$ as follows:

$$P' = \begin{bmatrix}
0.8 & 0.8 & \phi & \phi \\
0.8 & \phi & [0.8,1] & \phi \\
\phi & 0.8 & [0.8,1] & \phi \\
\phi & \phi & [0.5,1] & \phi \\
0.4 & \phi & \phi & 0.4
\end{bmatrix}$$

Since $x_3$ is the only nonempty entry in row 5, $x_3$ is super-essential. The associated fuzzy truth function for the simplified characteristic matrix $P'$ can be written as:

$$F_P' = \left( \frac{b_{3a}}{x_{ji}} \lor \frac{b_{3a}}{x_{ji}} \right) \land \left( \frac{b_{3a}}{x_{ji}} \lor \frac{b_{3a}}{x_{ji}} \right) \land \left( \frac{b_{3a}}{x_{ji}} \lor \frac{b_{3a}}{x_{ji}} \right) \land \left( \frac{b_{3a}}{x_{ji}} \lor \frac{b_{3a}}{x_{ji}} \right) \land \left( \frac{b_{3a}}{x_{ji}} \lor \frac{b_{3a}}{x_{ji}} \right)$$

After converting this fuzzy truth function above in CNF to DNF using laws of conversion in section 6.3.1 we have:

$$= \left( \frac{0.8}{x_{ji}} \lor \frac{0.8}{x_{ji}} \lor \frac{0.5}{x_{ji}} \right) \lor \left( \frac{0.8}{x_{ji}} \lor \frac{0.8}{x_{ji}} \right) \lor \left( \frac{0.8}{x_{ji}} \lor \frac{0.8}{x_{ji}} \right) \lor \left( \frac{0.4}{x_{ji}} \lor \frac{0.4}{x_{ji}} \right)$$
The corresponding minimal solutions of the system (6.1) are:

\[
\begin{align*}
\bar{x}^1 &= (0.8 \ 0.8 \ 0.5 \ 0 \ 0)^T, \quad \bar{x}^2 = (0.8 \ 0 \ 0.8 \ 0 \ 0)^T, \\
\bar{x}^3 &= (0 \ 0.8 \ 0.8 \ 0.4)^T, \quad \bar{x}^4 = (0.4 \ 0.8 \ 0.8 \ 0 \ 0)^T
\end{align*}
\]

The whole solution set of the fuzzy relation equation in (6.1) is given as:

\[
X(A,b) = \{0.8 \ 0.8 \ [0.5,1] \ [0.0,5] \ [0.0,4]] \cup \{0.8 \ [0.0,8] \ [0.8,1] \ [0.0,5] \ [0.0,4]] \cup [0.8,0.5] \ [0.0,5] \ [0.0,4]} \\
\cup \{0.8 \ [0.8,1] \ [0.0,5] \ 0.4] \cup [0.4,0.8] \ 0.8 \ [0.8,1] \ [0.0,5] \ [0.0,4]}. 
\]

It is clear that if \( x_j = 0 \) then \( l_j / x_j = \infty \). Keeping this fact in consideration the four optimization problems corresponding to four minimal solutions can be formulated as:

\[
P1 \quad \text{Min } f_1(x) = 5.59 x_1^2 x_4^{-1} x_5^4 + 2.92 x_3^2 x_4^{-2} x_5^4 \\
\text{s.t.} \quad 0.5 \leq x_1 \leq 1, \\
\quad 0 < x_4 \leq 0.5, \\
\quad 0 < x_5 \leq 0.4
\]

\[
P2 \quad \text{Min } f_2(x) = 5.22 x_2^{-0.3} x_3^2 x_4^{-1} x_5^4 + 2.09 x_2^{-1.5} x_3^2 x_4^{-2} x_5^4 \\
\text{s.t.} \quad 0 < x_2 \leq 0.8, \\
\quad 0.8 \leq x_3 \leq 1, \\
\quad 0 < x_4 \leq 0.5, \\
\quad 0 < x_5 \leq 0.4
\]

\[
P3 \quad \text{Min } f_3(x) = 2.14 x_1^{-0.2} x_3^2 x_4^{-1} + 1.12 x_1^{-0.2} x_3^2 x_4^{-2} \\
\text{s.t.} \quad 0 < x_1 \leq 0.8, \\
\quad 0.8 \leq x_3 \leq 1, \\
\quad 0 < x_4 \leq 0.5
\]

\[
P4 \quad \text{Min } f_4(x) = 5.35 x_1^{-0.2} x_3^2 x_4^{-1} x_5^4 + 2.795 x_1^{-0.2} x_3^2 x_4^{-2} x_5^4 \\
\text{s.t.} \quad 0.4 \leq x_1 \leq 0.8, \\
\quad 0.8 \leq x_3 \leq 1, \\
\quad 0 < x_4 \leq 0.5, \\
\quad 0 < x_5 \leq 0.4
\]
The optimal solution of problem (6.2) can be obtained as:

\[ f(x^*) = \min(f_1^*, f_2^*, f_3^*, f_4^*) \]

On solving these four optimization problems using Algorithm 2, the four good converging solutions are listed in table 6.1 and the performance of the corresponding GAs are shown graphically by the figures 6.2-6.5.

<table>
<thead>
<tr>
<th>Problem</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( f(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P1 )</td>
<td>0.8000</td>
<td>0.8000</td>
<td>0.6751</td>
<td>0.4511</td>
<td>0.000032</td>
<td>0.00039451521903</td>
</tr>
<tr>
<td>( P2 )</td>
<td>0.8000</td>
<td>0.7172</td>
<td>0.9420</td>
<td>0.3932</td>
<td>0.0001</td>
<td>0.00470617192855</td>
</tr>
<tr>
<td>( P3 )</td>
<td>0.7933</td>
<td>0.8000</td>
<td>0.8002</td>
<td>0.4991</td>
<td>0.4000</td>
<td>5.89151635308652</td>
</tr>
<tr>
<td>( P4 )</td>
<td>0.6826</td>
<td>0.8000</td>
<td>0.8131</td>
<td>0.2567</td>
<td>0.000037</td>
<td>0.00016933802324</td>
</tr>
</tbody>
</table>

Table 6.1: Converging solutions for the four optimization problems

It is clear that \( x^* = (0.6826, 0.8000, 0.8131, 0.2567, 0.000037) \) with \( f(x^*) = 0.00016933802324 \) is the optimal solution of the problem (6.1).
Figure 6.3: Performance of GA for $P2$

Figure 6.4: Performance of GA for $P3$
6.6 Conclusion

The chapter discusses a fuzzy relational geometric programming problem with a posynomial geometric objective function subjected to max-min fuzzy relation equation constraints. The extensive nonlinear nature of the objective function and feasible domain hinders many nonlinear programming techniques to be applied directly. Finding all the minimal solutions of the system (6.1) is considered to be an NP hard problem so a reduction procedure is required. In the proposed method, the solution procedure consists of two steps. Firstly, the fuzzy relational system is solved to determine the entire solution set. Different optimization problems are formulated with the help of unique maximal solution and the individual minimal solutions obtained. In the second level, a well designed genetic algorithm is applied to solve the different optimization problems obtained. The genetic operators crossover, mutation are designed after carefully studying the nature of feasible domain. The single optimal solution of the problem (6.2) is obtained with the help of the least converging solutions of the different optimization problems obtained.

Figure 6.5: Performance of GA for \( P4 \)