

Research Article

A Hybrid Technique Based on a Genetic Algorithm for Fuzzy Multiobjective Problems in 5G, Internet of Things, and Mobile Edge Computing

Allahkaram Shafiei,¹ Mohammad (Behdad) Jamshidi ,² Farzad Khani,³ Jakub Talla,² Zdeněk Peroutka,² Rahma Gantassi,⁴ Mohammed Baz ,⁵ Omar Cheikhrouhou ,⁶ and Habib Hamam ^{7,8}

¹Department of Computer Science, Czech Technical University, Prague, Czech Republic

²Research and Innovation Centre for Electrical Engineering (RICE), University of West Bohemia, Pilsen, Czech Republic

³The International Association of Engineers, Hong Kong, China

⁴Communication System Laboratory SysCom (ENIT), University of Tunis El Manar (UTM), Tunis, Tunisia

⁵Department of Computer Engineering, College of Computer and Information Technology, Taif University, P.O. Box. 11099, Taif 21994, Saudi Arabia

⁶CES Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia

⁷Faculty of Engineering, Université de Moncton, Moncton E1A3E9, NB, Canada

⁸School of Electrical Engineering, University of Johannesburg, Johannesburg 2006, South Africa

Correspondence should be addressed to Habib Hamam; habib.hamam@umoncton.ca

Received 12 April 2021; Revised 26 July 2021; Accepted 11 August 2021; Published 27 October 2021

Academic Editor: Dilbag Singh

Copyright © 2021 Allahkaram Shafiei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Emerging communication technologies, such as mobile edge computing (MEC), Internet of Things (IoT), and fifth-generation (5G) broadband cellular networks, have recently drawn a great deal of interest. Therefore, numerous multiobjective optimization problems (MOOP) associated with the aforementioned technologies have arisen, for example, energy consumption, cost-effective edge user allocation (EUA), and efficient scheduling. Accordingly, the formularization of these problems through fuzzy relation equations (FRE) should be taken into consideration as a capable approach to achieving an optimized solution. In this paper, a modified technique based on a genetic algorithm (GA) to solve MOOPs, which are formulated by fuzzy relation constraints with s -norm, is proposed. In this method, firstly, some techniques are utilized to reduce the size of the problem, so that the reduced problem can be solved easily. The proposed GA-based technique is then applied to solve the reduced problem locally. The most important advantage of this method is to solve a wide variety of MOOPs in the field of IoT, EC, and 5G. Furthermore, some numerical experiments are conducted to show the capability of the proposed technique. Not only does this method overcome the weaknesses of conventional methods owing to its potentials in the nonconvex feasible domain, but it also is useful to model complex systems.

1. Introduction

Fuzzy relation equations (FRE) theory has a large number of features making it capable of formulating uncertain information and nonlinear functions for complex systems [1–3]. For example, in [1], a description of the distinctive nature of

the bipolar max-product FRE solvability with a concentration on the standard negation was rendered. Moreover, several features associated with minimal or maximal solutions, as long as the equations can be solved, were demonstrated. Furthermore, there are many problems in different fields of studies based on communication technologies which can be

solved by FRE, such as the instructional information resources allocation, optimizing the size and performance of the microstrip components, the fifth-generation (5G) of broadband cellular networks, Internet of Things (IoT), could computing, and mobile edge computing (MEC), and a large number of engineering systems [1, 2, 4–9]. As technologies and new computational techniques based on fuzzy systems advance, we have faced a large number of issues that can be described by multiobjective optimization problems (MOOP) [10–14]. Besides, there are vast areas of research, such as deep learning in medicine or image processing, which can be extended using the proposed familiarization [15–19]. For instance, maximal-minimal FREs were presented to characterize the peer-to-peer (P2P) data transmission mechanism in the sharing system of the instructional information resources [2]. To solve these types of problems, evolutionary algorithms (EA), such as the genetic algorithm (GA), can be a suitable approach [20]. GA is generally exploited to create high-performance solutions to search problems and optimization [20]. This algorithm works based on a biologically inspired process, including operators, such as selection, crossover, and mutation. Thus, it is a suitable choice for minimizing multiobjective problems associated with wireless networks, IoT, and many other engineering MOOPs [21, 22]. In fact, the feasible domain in MOOP, is nonconvex and the objective functions are not necessarily linear, so the traditional methods are not applicable for these issues. Therefore, we are going to solve this problem using a modified GA-based method.

In this paper, the max-min operator, which is an intersection operator, is used to solve MOOPs of 5G, MEC, and IoT. In this way, we have used an s -norm operator which is a union operator. The norm we use in this paper is probabilistic sum-max, which is similar to the union of events in probability. This norm is handy, so it is more applicable with respect to other s -norms. In general, the GA has many aspects in common to solve fuzzy problems. However, one of the aspects of difference regarding this algorithm is the generation of the primary population which depends on the operator. The main and new part of this paper is about this generation. There would be also a new way to modify infeasible solutions. To do this, we should understand the meaning of the feasible domain perfectly as explained in further sections. Furthermore, we propose some methods to reduce the size of the problem without affecting the efficient solution.

This paper is organized as follows. Section 2 introduces preliminaries, definitions, and the main MOOPs of 5G, MEC, and IoT that can be solved by the proposed technique. Section 3 demonstrates the proposed GA-based method to solve the intended MOOP with detailed explanations stepwise. In Section 4, we illustrate the introduced algorithm by conducting several numerical experiments followed by conclusions in Section 5.

2. Multiobjective Optimization in MEC, IoT, and 5G

In this section, the most recent issues in the field of emerging communication technology and mobile edge computing that can be solved using the proposed technique are illustrated.

Although three different areas have been considered here, the point is that, in general, they can be illustrated as a fuzzy MOOP. Therefore, a common MOOP model is introduced to cover the problems.

5G-related problems are the first area that can be illustrated by fuzzy MOOP and the proposed method can solve them. In the communication area, the fifth-generation technology standard called 5G is the novel standard for enhancing the abilities of cellular communication systems. Since 2019, many telecommunications companies have started expanding this technology and it has been expected that it could be replaced with fourth-generation technology or 4G in the near future [23]. Based on the GSM Association report, this technology is expected to have at least 1.7 billion users and subscribers around the world until 2025. 5G wireless networks are a noticeable part of 5G technology that is predicted to offer many different kinds of facilities to use in a wide range of applications. Massive Multiple-Input-Multiple-output (mMIMO) is one of the most important 5G-based technologies working with a vast amount of data in gigabit scale and equipment, including service multiple users and numerous antennas that can cover all subscribers at the same frequency time. Accordingly, deploying 5G networks, which are integrated mMIMO technology for wireless application, needs the synchronization of a couple of different objectives and variables to reach an optimized operation and performance [24]. It is an obvious example of MOOP that could be easily solved using the proposed method. Moreover, one of the most significant issues in environmental-friendly transmission and wireless communication is to allocate the resources with considering efficient energy [25]. Therefore, this issue can be demonstrated as a MOOP, in which optimizing energy with resource allocation can be described. In addition, fuzzification of the problem is an appropriate way to cover those variables that are more influential for achieving an optimal solution, including energy efficiency (EE), power consumption, signal-interference noise ratio (SINR), and data rate. Besides, user scheduling is another problem in multiuser multiple-input-multiple-output (MU-MIMO). In other words, at an equivalent time and frequency condition, the network simultaneously involves several individuals from spatial-division multiplexing, which are used for the improvement of channels [26].

IoT issues that can be defined by fuzzy MOOP are placed as the second type of problems, which can be solved by this method. IoT demonstrates a network of different kinds of physical devices and objects called “things,” which are made and equipped by a software, hardware, sensors, other Internet-based equipment in order to exchange and connect the data with other systems and devices. One of the important features of 5G networks is their capabilities for adopting the IoT with clouds. Densification is a paradigm of 5G that can be described based on the fuzzy MOOP. It relates to facilitating the conditions, in which the provision of services is improved to predict a large number of user equipment in the future for IoT applications. In this regard, small cells are integrated with other parts of the network for supporting a huge number of user equipment, decreasing

energy consumption, and increasing the quality of services [27].

The third area in which this method could be used is solving MOOPs in MEC. With the fast advancement of telecommunication systems, the utilization of smartphones has become an inseparable part of today's life. In addition, considerable development of mobile application markets, which is originated from the high data rate and utilization of mobile technologies, leads to increasing the users in this area. MEC is a new technology in cellular telecommunication systems to create agile computation and enhance services and applications for mobile devices (MDs) anywhere and anytime through a range of endowing processes in wireless networks. According to the theoretical analysis and because the reduction of some parameters, such as price cost, execution delay, and energy consumption, is our objective, the formularization of such issues in a MOOP-oriented model has been considered [28]. To summarize, the aforementioned issues can be described as a fuzzy-MOOP issue that involves multiple objective functions as follows:

$$\begin{aligned} & \min\{f_1(x), f_2(x), \dots, f_p(x)\} \\ & \text{s.t. } x \circ A = b, \\ & 0 \leq x_i \leq 1, \end{aligned} \quad (1)$$

where the operations "o" is defined by

$$o: \max(x_i + a_{ij} - x_i a_{ij}) = b_j, \quad \text{for } 1 \leq i \leq n \text{ and } 1 \leq j \leq m. \quad (2)$$

The vector $x = (x_1, x_2, \dots, x_n)$ is the decision variable. The matrix $A = [a_{ij}]$ is an $m \times n$ matrix, where $0 \leq a_{ij} \leq 1$ and $b = (b_1, b_2, \dots, b_m)$, where $0 \leq b_j \leq 1$. The solution set of problem (1) is as follows:

$$X(A, b) = \{(x_1, x_2, \dots, x_n) | x \in [0, 1], 1 \in I, x \circ A = b\}. \quad (3)$$

For $x^1, x^2 \in X$, we say $x^1 \leq x^2$ if and only if $x_i^1 \leq x_i^2$ for all $i \in I$. Therefore, the relation " \leq " defines a partially ordered relation on X .

Definition 1. We call $\hat{x} \in x(A, b)$ the maximum solution if, for each $x \in X(A, b)$, $x \leq \hat{x}$. We also call $\check{x} \in x(A, b)$ the minimal solution if, for each $x \leq \check{x}$, one concludes $x = \check{x}$.

Theorem 1.

$$X(A, b) = \bigcup_{\check{x} \in \check{X}} \{x | \check{x} \leq x \leq \hat{x}\}. \quad (4)$$

Proof. See [29].

The objective f_k , $k = 1, 2, \dots, p$, might be nonlinear and the information about minimal solutions might be incomplete for this problem.

Let X be the feasible domain of (1); that is,

$$X = \{x \in \mathbf{R}^n | x \circ A = b, 0 \leq x_i \leq 1, i \in I\}. \quad (5)$$

$x \in X$ is called the solution vector and $Z = (f_1(x), f_2(x), \dots, f_p(x))$ the objective vector. \square

Definition 2. The point $\bar{x} \in X$ is called an efficient solution or the Pareto optimal solution if and only if there is no $x \in X$ such that, for all $k \in K$, we have $f_k(x) \leq f_k(\bar{x})$ and $f_k(x) < f_k(\bar{x})$ for some k . Otherwise, \bar{x} is called a non-efficient point. The set of all Pareto optimal solutions is called Pareto optimal set [30].

Definition 3. Suppose $z^1, z^2 \in Z$ are two objective functions. We say z^1 dominates z^2 if and only if $z^1 \leq z^2$ and $z^1 \neq z^2$. In fact, $z_k^1 \leq z_k^2$ for all $k \in K$ and $z_k^1 < z_k^2$ for at least one $k \in K$ [31].

Definition 4. We call $\bar{z} \in Z$ nondominated if and only if there is no $z \in Z$ such that z dominates \bar{z} [30].

The concept of nondomination is used for an objective vector, whereas the concept of efficiency is used for a solution vector. The vector $\bar{x} \in X$ is efficient if and only if its objective vector is nondominated. We cannot reduce the value of any of the objective vectors in an efficient point without increasing the value of one of the other objective vectors. *The efficient set or the optimal Pareto set* is the set of all efficient points. Similarly, the *nondominated set* is the set of all nondominated objective vectors. In this case, no utility function of the Pareto set is used to determine the optimal solution.

3. Solution for Multiobjective Optimization

A MOOP is commonly solved by scalarization or non-scalarization techniques. In scalarization, first, the MOOP is converted into a single-objective optimization problem by combining the multiple objectives through parameters. Then, the resulting single-objective or scalarized problem is solved by standard optimization methods and software. The nonscalarization techniques use some other way of solving. However, in using these methods, certain conditions on convexity and some other assumptions are needed to ensure the existence of the optimal and efficient solution. But in the intended problem, the feasible region is generally a non-convex set. In such a scenario, the genetic algorithm is a proven technique to solve the MOOP. Figure 1 presents the proposed Genetic Algorithm for MOOP.

3.1. Genetic Algorithm for MOOP. Since GA is a general method, it is used to solve many optimization problems where using traditional methods is very difficult to find exact solutions of the problems.

The Pareto optimization concept is the only approach for solution determination of a MOOP. Several methods use the genetic algorithm to solve a MOOP. The most famous of these methods are the population-based non-Pareto, plain aggregation approach, the Pareto-based approach, and the Niche induction approach [32–35].

3.2. Problem Reduction. Reducing the problem helps to solve the problem much easier by using the genetic algorithm. If the feasible domain of (1) is not empty, then it has a maximum solution \hat{x} , which is defined as follows [36, 37]:

$$\hat{x}_i = \bigwedge_{j=1}^m [a_{ij} \otimes b_j], \quad (6)$$

where

$$a_{ij} \otimes b_j = \begin{cases} 1, & (1 - a_{ij}) < b_j - a_{ij}, \\ \frac{b_j - a_{ij}}{1 - a_{ij}}, & (1 - a_{ij}) \geq b_j - a_{ij}. \end{cases} \quad (7)$$

Now, suppose $\hat{x}_{i'} = 0$ for some $i' \in I$. Since $x \leq \hat{x}$ for each $x \in X$, then $x_{i'} \leq \hat{x}_{i'}$. Therefore, for each $x \in X$, $x_{i'} = 0$.

Definition 5. For $x \in X(A, b)$, we call $x_{i'}$ a binding variable if $x_i(1 - a_{ij_0}) + a_{ij_0} = b_{j_0}$ for $j_0 \in J$.

Set

$$I_j = \{i \in I | x_i(1 - a_{ij}) + a_{ij} = b_j\} \quad (8)$$

if $X(A, b) \neq \emptyset$, and

$$\Lambda = I_1 \times I_2 \times \cdots \times I_m, \quad (9)$$

where I_j is the index set corresponding to \hat{x}_i which is a binding variable for j th constraints.

The following lemma is similar to Lemma 3 from [38].

Lemma 1. If $\|I_j\| = 1$, then, for $i \in I_j$, we have $\hat{x}_i = \tilde{x}_i = ((b_j - a_{ij}) / (1 - a_{ij}))$.

Let $J_f^i = \{j \in J | f_j = i\}$ be an index set. For each $f \in \Lambda$, the set J_f^i corresponds to the index set of constraints, where \hat{x}_i is the binding variable for them.

Lemma 2. For any $j, j' \in J_f^i$, we have

$$\frac{b_{j'} - a_{ij'}}{1 - a_{ij'}} = \frac{b_j - a_{ij}}{1 - a_{ij}}. \quad (10)$$

Proof. See [38] for details of the proof.

Let us define the function $F: \Lambda \rightarrow [0, 1]^n$ which is of course a member of $X(A, b)$ [39] as follows [29]:

$$F_i(f) = \begin{cases} \frac{b_j - a_{ij}}{1 - a_{ij}}, & j \in J_f^i \neq \emptyset, \\ 0, & J_f^i = \emptyset. \end{cases} \quad (11)$$

Now suppose $\|I_j\| = 1$ and $k \in I_j$. According to Lemma 1, for any feasible solution $x \in X$, the value of x_k is constant and equals the value of \hat{x}_k . Now, set

$$\begin{aligned} I^0 &= \{i \in I | \hat{x}_i = 0\}, \\ J^0 &= \{j \in J | i \in I_j, i \in I^0\}, \\ I' &= I \setminus I^0, \\ J' &= J \setminus J^0. \end{aligned} \quad (12)$$

We eliminate the i th row of the matrix A for $i \in I^0$ and the j th column of the matrix A for $j \in J^0$ and call this generated matrix A' . Similarly, we eliminate the j th component of the vector b for $j \in J^0$ to generate a reduced vector b' . Now, let us set

$$\begin{aligned} \bar{J} &= \{j \in J' | \|I_j\| = 1\}, \\ \bar{I} &= \{i \in I' | i \in I_j, j \in \bar{J}\}. \end{aligned} \quad (13)$$

We eliminate the i th row of A' for $i \in \bar{I}$ and j th column of A' for $j \in \bar{I}$ to get the matrix A'' and eliminate the j th component of b' for $j \in \bar{J}$ to get the vector b'' . Therefore, we set

$$\begin{aligned} I'' &= I' \setminus \bar{I}, \\ J'' &= J' \setminus \bar{J}, \\ \Lambda'' &= \prod_{j \in J''} I_j. \end{aligned} \quad (14)$$

Now onwards, we will work with this reduced problem. \square

3.3. The Proposed Modified GA-Based Approach. As shown in Figure 1, before using the proposed algorithm, the problem is reduced by using the methods introduced above. The most important parts of the proposed genetic algorithm are the crossover operator, mutation, and selection-reproduction procedure. In this proposed algorithm, any solution vector is an n dimensional vector $x = (x_1, \dots, x_n)$ where $x_i \in [0, 1]$ for $i = 1, 2, \dots, n$. Our genetic algorithm proposed for MOOP includes an iterative loop. Before the loop, a primary set is generated by using Algorithm 1 which we will expose later. Effective solutions are put in a set E . This set is updated inside the loop composed of steps 4, 5, and 6. The entire process will be detailed in Algorithm 2.

3.3.1. Creation of the Primary Population. The creation of the primary population is directly related to the success of the proposed genetic algorithm. The primary population must be generated in a way to suitably cover the search region. Using this information, we introduce an applicable method. This method is divided into two separate parts.

- (i) Vector b'' is empty: in this case, all restrictions are changed to equality using the constant components of the solution vector. Then, we assign any value between zero and the upper bound of the variable to each variable.
- (ii) Vector b'' is not empty: in this case, we divide the primary population into two parts. The first part lies

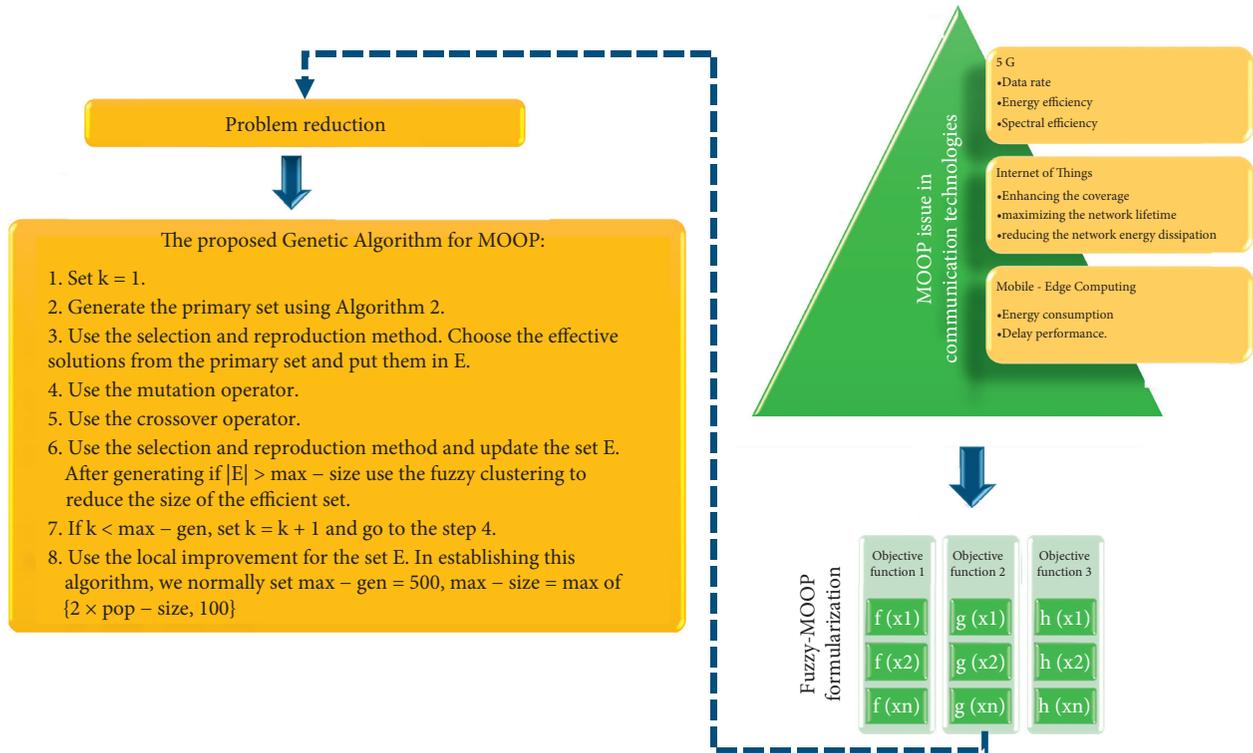


FIGURE 1: The proposed genetic algorithm for MOOP.

Suppose d is the dimension size of the number of the population which is supposed to be generated.

- (1) If b^i is empty, then generate $2 \times |I''| \times d$ members as follows: for each x the value of the i th component for $i \in I''$ is chosen randomly from $[0, \hat{x}_i]$.
- (2) If b is not empty, then generate the first part of the population for each $i' \in I''$ as follows: set $x_i = \hat{x}_i$ for $i \in I$ and $i \neq i'$, and assign a random value from $[0, \hat{x}_i]$ to $x_{i'}$. We generate the second part of the population by choosing $x_{i'}$ from $[0, \hat{x}_{i'}]$ for each $i' \in I''$. This generates $|I''| \times d$ individuals.
- (3) Check the feasibility of the new solution. If the solution is not feasible, use Algorithm 3.

ALGORITHM 1: Creation of the primary population

Suppose k is the number of individuals generated, max-size is the size of the efficient set, max-gen is the maximum of the generation.

- (1) Set $k = 1$.
- (2) Generate the primary set using Algorithm 1.
- (3) Use the selection and reproduction method. Choose the effective solutions from the primary set and put them in E .
- (4) Use the mutation operator.
- (5) Use the crossover operator.
- (6) Use the selection and reproduction method and update the set E . After generating, if $|E| > \text{max-size}$, use the fuzzy clustering to reduce the size of the efficient set.
- (7) If $k < \text{max-gen}$, set $k = k + 1$ and go to step 4.
- (8) Use the local improvement for the set E . In establishing this algorithm, we normally set $\text{max-gen} = 500$, $\text{max-size} = \max\{2 \times \text{pop-size}, 100\}$.

ALGORITHM 2: The proposed algorithm for fuzzy multiobjective problems.

on the boundary of the feasible domain and the second part lies in the interior of the feasible domain.

Lemma 3. *The lower bound of the variable x_i is zero for $i \in I''$.*

Proof. For $i \in I''$, if $i \in I'' \cap I_j$, then we have $\|I_j\| > 1$. Because of this, we choose $p \in \Lambda$ such that none of the components of p is equal to i . This means that $F_i(p)$ is equal to zero and because $F(p) \in X$, a solution is found where its i th component x_i is zero. Therefore, the lower bound of x_i is zero.

Now, we are going to generate some solutions as the population using Lemma 3. We suppose every component of the maximum solution is constant except for the i th component for $i \in I''$, which is supposed to be zero. The solution x' which is obtained using this procedure is a solution vector. The reason is that $x' \leq \hat{x}$ and we know that there is a minimal solution \hat{x} where its i th component is zero, that is, $x'_i = \hat{x}_i$. Since the other components of x' are the same as the components of \hat{x} , $\hat{x} \leq x'$, and by Theorem 1, x' is the feasible solution of problem (1). We use the lower bound and \hat{x} to generate the primary population. The number of the generated population is called the dimension size and it is the same for each component. The second part of the primary population is generated by choosing a number between zero and the maximum of the components satisfying the feasibility. If the solution is not feasible, we improve it such that all the constraints change to equality. This is done by using Algorithm 3. The total number of the population generated in this way equals $2 \times |I''| \times d$, where d is the dimension size. \square

Example 1. Consider the following reduced problem:

$$\begin{aligned} A &= \begin{pmatrix} 0.3 \\ 0.4 \end{pmatrix}, \\ b &= 0. \end{aligned} \quad (15)$$

The maximum solution of this problem is $\hat{x} = (0.71, 0.66)$. Now, generate the primary population. We assign the maximum value 0.66 to x_2 and choose d random values in $[0, 0.71]$ for x_1 . For the next step, we set $x_1 = 0.71$ and choose d random values in $[0, 0.66]$ for x_2 . We also generate the second part of the population by choosing x_1 's in $[0, 0.71]$ and x_2 's in $[0, 0.66]$. As soon as a solution is generated, we check if the unique restriction of the problem changes to equality; then, the solution is a member of the population. Otherwise, we use Algorithm 3 to make it feasible.

3.3.2. Selection and Reproduction Method. There is a fitness vector in a multiobjective optimization problem. The fitness vector is determined by the objective function vector. Since the genetic algorithm needs a scalar to evaluate each

individual, objective functions must be combined in such a way to create such a scalar.

Several methods have been introduced for this purpose. One of these methods is called Roulette Wheel. The ranking (r_p) which is used in this method is based on the number of dominants:

$$\begin{aligned} r_p &= \text{pop} - \text{size} - \{\text{the number of dominants}\}, \\ p &= 1, \dots, \text{pop} - \text{size}. \end{aligned} \quad (16)$$

The probability of choosing an individual to have the rank r_p is

$$p(\text{choosing an individual}) = \frac{r_p}{\sum_{i=1}^{\text{pop-size}} r_i}. \quad (17)$$

By the selection and reproduction method, the population tends towards the efficient set. However, random errors related to crossover and mutation operators can cause the divergence of the population. In fact, in this method, we consider an interval for each individual. The total length of the interval is $\sum_{i=1}^{\text{pop-size}} r_i$. We consider the intervals as $[0, \sum_{i=1}^{\text{pop-size}} r_i]$. We partition this interval into $\text{pop} - \text{size}$ sections, where the length of the i th section is r_i . For choosing an individual (solution), it is enough to choose a random number in $[0, \sum_{i=1}^{\text{pop-size}} r_i]$. If this number is in the i th section, the solution to this section will be chosen to generate. This procedure may be used for both convex and nonconvex solution spaces. However, the combination of two good solutions cannot always generate a good solution, especially if the objective function is nonlinear. Therefore, during this procedure, we must keep the non-dominant or efficient solutions.

Suppose E is the set of all efficient solutions. It is also supposed that \tilde{E} is the efficient set related to the latter set. The set \tilde{E} is based on the current population, so none of the individuals in the current population has domination on the members of \tilde{E} . In each step of the iteration of the choosing method, after that the fitness of each individual is finalized, we check whether any individual of \tilde{E} has domination over any individual of E . If this is the case, the individual will replace the one that is dominated. If this is not the case, we find out whether it is dominated by a member of E . This member will be added to E if this is not the case. Otherwise, no change will be made in E .

Because the efficient solution set is usually connected, set E is getting larger by the evolution program. This will cause some difficulties in the calculation. For preventing this problem, after 50 generations, the efficient set is partitioned into several manageable subsets. In establishing the process, the maximum size is determined by two parameters: 100 and the population size. On one hand, this set has to be large enough to determine the efficient set, and on the other hand, it should not be very large, since the efficiency of the algorithm should not be ignored. For this purpose, we use fuzzy clustering methods to keep the size of the efficient set manageable [40].

3.3.3. Mutation. The idea is to partially modify an individual (solution) while keeping some of the current information. In

- (1) Choose one of the constraints which cannot be changed to equality, e.g., the i th constraint
- (2) Choose one of the indices in I_j randomly, e.g., i , and set the value of the i th component equal to \hat{x}_i
- (3) The feasibility of the novel solution is evaluated. If it is not feasible go to the first step; otherwise, stop

ALGORITHM 3: Improvement of the solution for optimization

the maximum solution, each variable is active. Using this feature, we introduce a mutation operator to give variation to the population. Any individual has a chance to have a mutation. In this case, any component is chosen for mutation. We assign a value between zero and the maximum value and investigate the mutated individual. If it is not feasible, we modify it. Algorithm 4 provides more details about the mutation operator.

Example 2. Suppose we have a reduced system of fuzzy relation equations as follows:

$$\begin{aligned}
 b &= (0.8, 0.9), \\
 A &= \begin{pmatrix} 0.5 & 0.4 \\ 0.7 & 0.6 \\ 0.2 & 0.8 \end{pmatrix}, \\
 \hat{x} &= (0.6, 0.33, 0.5), \\
 I_1 &= \{1, 2\}, \\
 I_2 &= \{3\}.
 \end{aligned} \tag{18}$$

Consider the solution vector $x^1 = (0.6, 0.03, 0.5)$. Suppose $\epsilon = 0.1$. We generate a 3-dimensional vector r , where the components belong to $[0, 1]$. For example, $r = (0.0672, 0.391, 0.748)$. As the only component of r , which is less than ϵ is defined as r_1 , only x_1 is mutated and x_2, x_3 will remain the same. Therefore, $x^1 = (0.482, 0.03, 0.5)$ is a solution. If we look at the mutated solution to analyze feasibility, we find out that the first variable does not change to equality and so the second and third variables may change it to equality. Suppose the second index is chosen; then, by Algorithm 3, set $x_2^1 = \hat{x}_2$. The solution is in the form $(0.482, 0.33, 0.5)$, which is the feasible solution to the problem.

3.3.4. Crossover. In this paper, we use the two-point crossover operator [41]. The reason is that it keeps the good characters of the individual (solution) for the next generation. Despite the mutation operator, this operator makes the population by combining two different individuals. The good values of the variables will probably remain safe. The crossover operator is particularly designed for the solution space of the fuzzy relation equations [42].

Definition 6. Suppose $x^1, x^2 \in X$, $0 \leq \lambda \leq 1$, and $\gamma \geq 1$.

- (1) A linear contraction of x^1 supervised by x^2 defines a new point:

$$x' = \lambda x^1 + (1 - \lambda)x^2. \tag{19}$$

- (2) A linear extraction of x^1 supervised by x^2 defines a new point:

$$x' = \gamma x^1 - (\gamma - 1)x^2. \tag{20}$$

Crossover operator can be defined as both linear contraction and linear extraction supervised by another solution. The probability of occurring contraction and extraction are equal. Therefore, some pairs of solutions may be contracted and some of them may be extracted. The contraction-related parameter λ is chosen randomly from $[0.5, 1]$ and the extraction-related parameter γ is chosen randomly from $[1, 1.5]$. Two individuals are generated by separate parents. One of them by contraction (extraction) of x^1 supervised by x^2 and the other by contraction (extraction) of x^2 supervised by x^1 [42]. Moreover, these two individuals may be contracted (extracted) and supervised by \hat{x} which has the probability of δ . There is also a possibility that after the operation of crossover, the individual (solution) is not feasible. That is, some of its components may be out of $[0, \hat{x}]$ or cannot change some of the restrictions to equality. In this case, the individual (solution) is modified by Algorithm 3 to change to make a feasible solution. More details of the crossover operator are available in Algorithm 5 [42].

Example 3. Consider the last example:

$$\begin{aligned}
 x^1 &= (0.6, 0.03, 0.5), \\
 x^2 &= (0.025, 0.33, 0.5).
 \end{aligned} \tag{21}$$

Create the random number r , and suppose that $r = 0.35$. Therefore, we establish the contraction operator. The created matrix is as follows:

$$\Lambda = \begin{bmatrix} 0.759 & 0 & 0 \\ 0 & 0.623 & 0 \\ 0 & 0 & 0.915 \end{bmatrix}. \tag{22}$$

Note that the values on the main diagonal have been chosen randomly from $[0.5, 1]$. Set

$$x^{-1} = x^1 \Lambda + x^2 (I - \Lambda). \tag{23}$$

For x^{-2} , create a random number r from $[0, 1]$. Suppose $r = 0.65$. Now establish the extraction and the created matrix is as follows:

- (1) Generate the random vector $r \in [0, 1]^n$. Each component of the vector r corresponds to one of the components of the solution vector.
- (2) For $i = 1, 2, \dots, n$, if $r_i \leq \epsilon$, assign a random value in $[0, \hat{x}_i]$ to x_i .
- (3) Check whether the solution is feasible. If it is not feasible, we modify the solution using Algorithm 3. This operator will give a variation to the value of the solution vector and prevent converging the population to the maximum solution. In this method, we consider the arbitrary parameter $\epsilon = 0.1$. This means each component of the solution has a 10 percent chance to be modified.

ALGORITHM 4: Mutation operator

Suppose p is the number of pairs chosen so far.

- (1) Set $p = 1$.
- (2) Choose two individuals randomly from the current population.
- (3) Generate the random $r \in [0, 1]$. If $r \geq 0.5$, go to step 5.
- (4) Generate a random $n \times n$ matrix Λ as follows:
 $a_{ii} \in [0.5, 1]$ is a random variable and $a_{ij} = 0$, for $i \neq j$.
Set $x^{-1} = x^1 \cdot \Lambda + x^2 \cdot (I - \Lambda)$ where I is the $n \times n$ identity matrix and Γ is the contraction coefficients matrix. Go to 6.
- (5) Generate a random $n \times n$ matrix Γ as follows:
 $a_{ii} \in [1, 1.5]$ is a random variable and $a_{ij} = 0$, for $i \neq j$.
Set $x^{-1} = x^1 \cdot \Gamma - x^2 \cdot (\Gamma - I)$ where I is the $n \times n$ identity matrix and Γ is the extraction coefficients matrix.
- (6) Generate the random number $r \in [0, 1]$. If $r \geq 0.5$, go to step 8.
- (7) Generate the $n \times n$ matrix Λ as follows:
 $a_{ii} \in [0.5, 1]$ is a random variable and $a_{ij} = 0$, for $i \neq j$.
Set $x^{-2} = x^2 \cdot \Lambda - x^1 \cdot (I - \Lambda)$. Go to step 9.
- (8) Define the random matrix Λ as follows:
 $a_{ii} \in [1, 1.5]$ is a random variable and $a_{ij} = 0$, for $i \neq j$.
Set $x^{-2} = x^2 \cdot \Gamma - x^1 \cdot (\Gamma - I)$.
- (9) Choose the random variable $r_1 \in [0, 1]$. If $r_1 \geq \delta$, then go to the next step. Otherwise, generate $r_2 \in [0, 1]$. If $r_2 \leq 0.5$, generate a new matrix Λ and set $x^{-1} = x^{-1} \cdot \Lambda - \hat{x} \cdot (I - \Lambda)$. Otherwise, generate a new Γ and set $x^{-1} = x^{-1} \cdot \Gamma - \hat{x} \cdot (\Gamma - I)$.
- (10) Choose the random variable $r_1 \in [0, 1]$. If $r_1 \geq \delta$, then go to the next step. Otherwise, generate $r_2 \in [0, 1]$. If $r_2 \leq 0.5$, generate a new matrix Λ and set $x^{-2} = x^2 \cdot \Lambda - \hat{x} \cdot (I - \Lambda)$. Otherwise, generate a new Γ and set $x^{-2} = x^2 \cdot \Gamma - \hat{x} \cdot (\Gamma - I)$.
- (11) If x_i^{-1} (x_i^{-2}) does not belong to $[0, \hat{x}_i]$, for some $i \in I''$, change the value of x_i^{-2} (x_i^{-1}) to a random variable in $[0, \hat{x}_i]$.
- (12) Check the feasibility of x_i^{-1} and x_i^{-2} . If any of them is not feasible, modify them and make them feasible using Algorithm 3.
- (13) Put x_i^{-1} and x_i^{-2} in the new population.
- (14) Set $p = p + 1$. If $p \leq ((\text{pop} - \text{size})/2)$, then go to step 2. Otherwise, stop.

ALGORITHM 5: The crossover operator.

$$\Gamma = \begin{bmatrix} 1.013 & 0 & 0 \\ 0 & 1.28 & 0 \\ 0 & 0 & 1.379 \end{bmatrix}. \quad (24)$$

Set

$$x^{-2} = x^2 \Gamma - x^1 (\Gamma - I), \quad (25)$$

and suppose that the default value for δ is 0.1. In step 9, suppose that $r_1 = 0.309$ for x^{-1} . Since $r_1 > \delta$, there would neither be contraction nor extraction for x^{-1} under the supervision of \hat{x} . We generate another random number, for example, 0.078. Since $r_1 < \delta$, we generate r_2 . Since $r_2 = 0.27$, we generate a new matrix Λ and set

$$x^{-2} = x^{-2} \Lambda + \hat{x} (I - \Lambda). \quad (26)$$

If x^{-1} and x^{-2} are not feasible via Algorithm 3, we can modify them. Therefore, the solutions obtained by this method

generate the members of the future generation. We continue establishing the operating operator until the society is full.

3.3.5. Local Improvement. The local improvement method is used to improve the solutions obtained via the genetic algorithm [42]. In each iteration, one point of a neighborhood of the current point is selected. Basically, we perturb the solutions inside E . The rate of this perturbation has a specific restriction, for example, 0.1. Therefore, we generate a random vector using the values of $[-0.05, 0.05]$; then, we add this vector to the solution vector. As soon as this is done, we check if the vector is feasible. If so, the reverse action will give the solution vector. Further, we update set E as follows.

If any new solution has domination on other solutions, these solutions will be deleted from the set E and we replace them with this new solution. If any new solution does not have domination on other solutions, we shall not add it to E as we are looking for a better solution, not just a good solution. Each of the solutions in

E will be perturbed finitely many times, for example, 100 times. This method is very useful for deleting unsuitable solutions and closes the solutions to an optimized solution.

3.4. *A Summary on Proposed GA Approach.* The proposed genetic algorithm for solving a MOOP with fuzzy relation constraints is as follows:

4. Demonstration of Applications

The aim of proposing the genetic algorithm is to use it in solving difficult problems. Using this method, we obtain the optimized Pareto solutions. However, before using this method, we should make sure of the applicability of the method. Hence, we use this algorithm to solve some simple problems at first.

Example 4. Optimization using two variables and a linear objective function while the Pareto set contains only one point.

This problem has four variables and two linear objective functions. This problem can be transferred to a two-objective unrestricted problem after reduction. The information provided for this problem is as follows:

$$A = \begin{bmatrix} 0.3 & 0.5 & 0.85 \\ 0.7 & 0.1 & 0.8 \\ 0.4 & 0.2 & 0.77 \\ 0.75 & 0.6 & 0.9 \end{bmatrix}, \quad (27)$$

$$b = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.95 \end{bmatrix}.$$

The objective function of the problem is

$$\min f = (0.1x_1 - 0.5x_2 - 0.3x_3 + 0.8x_4 - 0.2x_1 - 0.3x_3 + x_4), \quad (28)$$

and the maximum solution is

$$\hat{x} = (0.6000 \ 0.667 \ 0.7500 \ 0.5),$$

$$I_1 = \{2\},$$

$$I_2 = \{1, 3, 4\},$$

$$I_3 = \{4\}. \quad (29)$$

Using reduction methods, we may delete the second and fourth variables and the restrictions of the problem. Therefore, the problem changes to an unrestricted optimized problem with two variables. By the fact that the coefficients of x_1 and x_3 in each of the objective functions are negative, they must get their maximum values. That means, the optimized set is a one-point problem and contains only the maximum solution. Applying the genetic algorithm for this problem, the same result is obtained. Figure 2 presents the Pareto optimal set contains one point.

Example 5. Optimization using two variables and a linear objective function while the Pareto set is on one edge.

The specifications of the problem are the same as the previous problem. Only the differences are the coefficients of the objective function. The objective functions are

$$\min = \begin{cases} f_1 = 0.1x_1 + 0.6x_2 + 0.3x_3, \\ f_2 = -0.2x_1 - 0.3x_2 + 0.3x_3 - 0.1x_4. \end{cases} \quad (30)$$

The coefficient of x_3 is positive in both objective functions, and variable x_3 has to get its minimum, that is, zero. The coefficient of x_1 in one of the objective functions is positive, and in the other case, it is negative. Therefore, a modification for one case causes a weakness in the other case and vice versa. So, the optimized Pareto set lies on the edge $(x_1, 0): 0 \leq x_1 \leq \hat{x}_1$. Using the genetic algorithm gives the same result. Figure 3 presents The Pareto optimal set is on one edge.

Example 6. Optimization with two variables along with linear objective functions while the Pareto set lies on two connected edges.

In this case, we consider again the previous problem with the following objective functions:

$$\min = \begin{cases} f_1 = 0.1x_1 + 0.7x_2 - 0.3x_3 + 0.6x_4, \\ f_2 = -0.2x_1 - 0.9x_2 + 0.3x_3 - 0.1x_4. \end{cases} \quad (31)$$

The Pareto optimal solution which is obtained analytically is

$$\{x_1 = 0, x_2 = 0.667, 0 \leq x_3 \leq 0.76, x_4 = 0.5\} \cup \{0 \leq x_1 \leq 0.6000, x_2 = 0.667, x_3 = 0, x_4 = 0.5\}. \quad (32)$$

The genetic algorithm yields the same solution. Figure 4 presents The Pareto optimal set is on two connected edges.

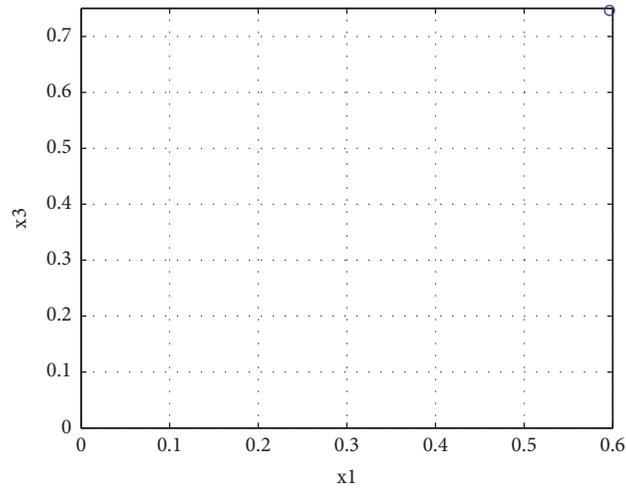


FIGURE 2: The Pareto optimal set contains one point.

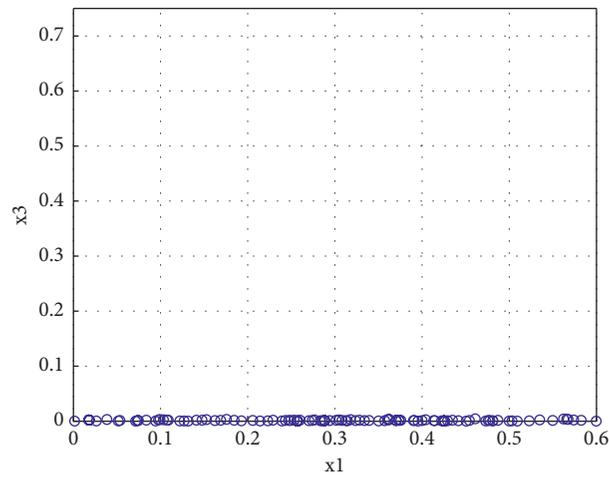


FIGURE 3: The Pareto optimal set is on one edge.

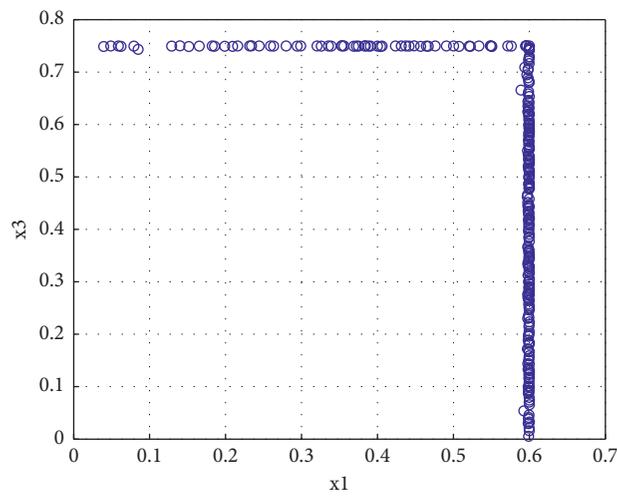


FIGURE 4: The Pareto optimal set is on two connected edges.

Example 7. Optimization in three-dimensional space along with three objective functions while the Pareto set lies on two connected edges.

The problem we are going to solve is an expansion of the previous problem. The problem is specified as follows:

$$\begin{aligned} \hat{x} &= (0.6000 \ 0.667 \ 0.75 \ 0.5 \ 0.55), \\ b &= \begin{pmatrix} 0.9 \\ 0.8 \\ 0.95 \end{pmatrix}, \\ A &= \begin{bmatrix} 0.3 & 0.5 & 0.85 \\ 0.7 & 0.1 & 0.8 \\ 0.4 & 0.2 & 0.77 \\ 0.75 & 0.6 & 0.9 \\ 0.75 & 0.5556 & 0.4 \end{bmatrix}, \\ I_1 &= \{2\}, \\ I_2 &= \{1, 3, 4, 5\}, \\ I_3 &= \{4\}. \end{aligned} \tag{33}$$

Using the reduction process, all three restrictions are omitted and x_1, x_3, x_5 remain. The objective functions of the problem are

$$\min = \begin{cases} f_1 = 2x_1 + 5x_2 + 2x_3 - x_4 + x_5, \\ f_2 = -3x_1 + 6x_3 - 3x_4 + 2x_5, \\ f_3 = 8x_1 + 7x_2 - 3x_3 + 3x_5. \end{cases} \tag{34}$$

Because the coefficient of x_5 is positive in f_i for $i \in \{1, 2, 3\}$, it must be zero. Therefore, the analytic solution to the problem is

$$\{x_1 = 0.6, x_2 = 0.6667, 0 \leq x_3 \leq 0.75, x_4 = 0.5, x_5 = 0\} \cup \{0 \leq x_1 \leq 0.6, x_2 = 0.667, x_3 = 0.75, x_4 = 0.5, x_5 = 0\}. \tag{35}$$

Using the genetic algorithm, the accuracy is checked. Figure 5 presents the Pareto optimal set is on two connected edges.

Example 8. Optimization in three-dimensional vector space while the Pareto set lies on one edge.

Once again consider the previous problem but with different objective functions:

$$\min \begin{cases} f_1 = 2x_1 + 3x_2 + 2x_3 - x_4 + x_5, \\ f_2 = 3x_1 - 5x_2 + 6x_3 + x_4 + 2x_5, \\ f_3 = 8x_1 + 3x_3 + 2x_4 - 3x_5. \end{cases} \tag{36}$$

Noting that the coefficients of x_1 and x_3 for all three cases are positive shows that these variables have to be zero. The coefficient of x_5 is negative in the first and third objective functions and positive in the second one; therefore, it may choose any value between zero and 0.55. This result satisfies the result we get using the algorithm as well. Figure 6 presents the Pareto optimal set is on two connected edges.

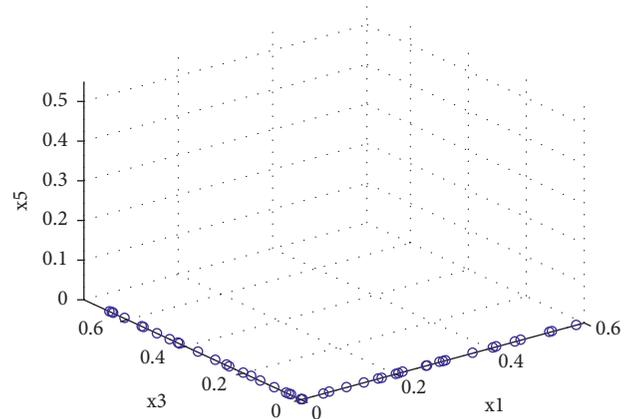


FIGURE 5: The Pareto optimal set is on two connected edges.

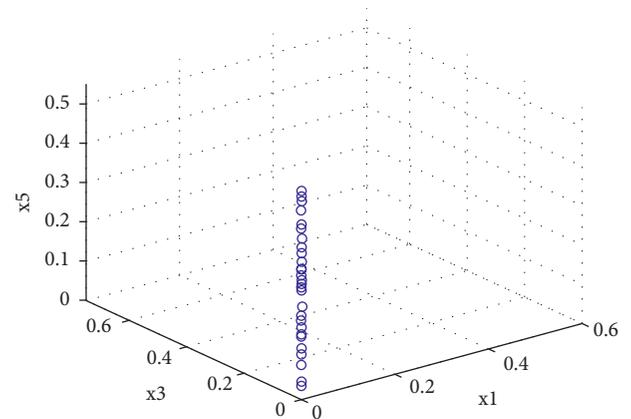


FIGURE 6: The Pareto optimal set is on one edge.

Example 9. Optimization in three-dimensional vector space while the Pareto set contains only one point.

We use the same problem with different objective functions again. Figure 6 presents The Pareto optimal set is on two connected edges.

$$\min \begin{cases} f_1 = 2x_1 - 3x_2 + 2x_3 + x_4 + x_5, \\ f_2 = 3x_1 + x_2 + 6x_3 - 5x_4 + 2x_5, \\ f_3 = 6x_1 - 4x_2 + 3x_3 - 10x_4 + 3x_5. \end{cases} \tag{37}$$

Noting that the coefficients of x_1, x_3, x_5 are positive, they must be zero. Therefore, the Pareto set contains only one point which satisfies the solution by the genetic algorithm too. Figure 7 presents the Pareto optimal set is on two connected edges.

Example 10. Optimization with a nonlinear objective function.

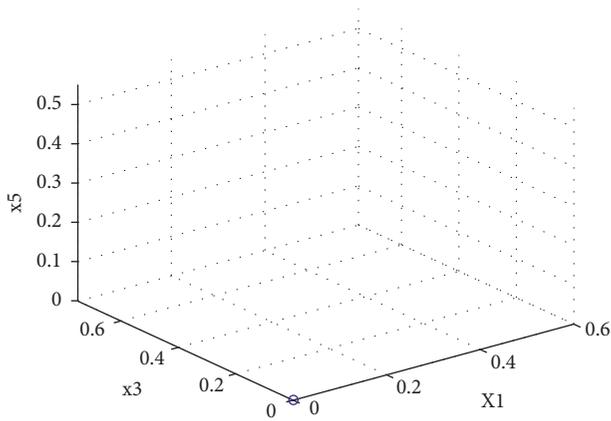


FIGURE 7: The Pareto optimal set is on two connected edges.

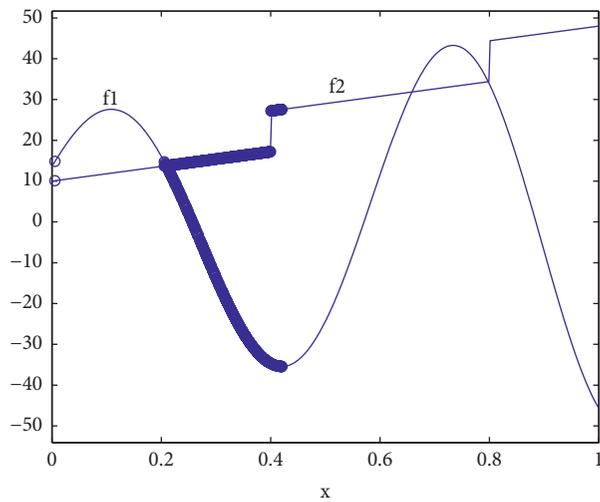


FIGURE 8: The Pareto optimal set for a nonlinear objective function.

The objective function for this case is chosen from

$$A = \begin{bmatrix} 0.5 & 0.85 \\ 0.6 & 0.9 \end{bmatrix},$$

$$b = \begin{bmatrix} 0.8 \\ 0.95 \end{bmatrix},$$

$$I_1 = [1, 2],$$

$$I_2 = [2],$$

$$\hat{x} = (0.6000, 0.5),$$

$$\min \begin{cases} f_1 = -25(x_1 + 1) \times \sin(10x_1 + 10), \\ f_2 = 10\left(\frac{x_1}{0.4} + 1\right) + 18x_1. \end{cases} \quad (38)$$

Using the reduction process, both restrictions of the problem are omitted. Moreover, the variable x_2 is also omitted and we only dealt with a one-variable problem. The problem may be solved by drawing and finding out the related Pareto set. The Pareto set obtained by the genetic algorithm is a good estimation of the real optimized Pareto set.

The results obtained so far show that the proposed genetic algorithm is a very applicable method for solving MOOPs and may be used to solve some difficult and complex problems as well. Figure 8 presents the Pareto optimal set for a nonlinear objective function.

For future work, this method can be developed to overcome some problems in 6G. Generally, the 6G can be described as a full-region perfect network enabled through a data rate of 100 Gbps, representing a very wider bandwidth in comparison with existing networks [43]. This is best understood when it comes to basic channel coding. New challenging issues for modulation techniques and basic channel coding should be considered. For example, nonorthogonal wave, next-generation channel coding, free duplex/full-duplex spectrum, multiple access system, dynamic spectrum sharing technology, Internet of Things in 6G, and the blockchain technology altogether are the potential problems in 6G that can be solved with the proposed method.

5. Conclusion

In this paper, a modified GA-based technique has been introduced and validated to be used as a powerful tool for solving fuzzy MOOPs. At the first stage, the problem is reduced to be solved more easily. The size reduction of the problem is done using methods that can determine the constant solutions to the problem. Designing the mutation and crossover operators is based on the information obtained from the structure of the problem. Regarding uncertainties and nonlinearities within emerging communication technologies, this feature is helpful to solve a great number of MOOPs in MEC, 5G, IoT, and many other engineering problems, which can be described with FRE constraints. The feasibility of the problem is always protected and the efficient set is updated. The size of the efficient set is being managed throughout the steps and by the fuzzy clustering method. This magnificently shows the applicability of the algorithm. Finally, a localization improvement for enhancing the efficient set was proposed. We did some numerical experiments to show the extent of applicability of the proposed algorithm.

Although 5G networks are more suitable to be utilized in communicating between the Internet of Things and users, 6G networks could be more productive in communicating among businesses, corporations, individuals, and, of course, multiple intelligent objects, which will arise numerous challenges and problems. Therefore, using more efficient and practical techniques based on evolutionary algorithms, we will alleviate the difficulties associated with such complex structures. Thus, for future work, particularly in 6G, introducing such a problem in the form of a MOOP will provide us with this opportunity to employ the proposed method in conquering these nonlinearities and uncertainties as well as finding the best possible solutions.

Data Availability

There are no data for sharing.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Taif University Research Supporting Project (no. TURSP-2020/239), Taif University, Taif, Saudi Arabia. This work was supported by the Research Center for Informatics through OP VVV project CZ.02.1.01/0.0/0.0/16 019/0000765 and also the Ministry of Education, Youth and Sports of the Czech Republic through the Project OP VVV Electrical Engineering Technologies with High-Level of Embedded Intelligence under Grant CZ.02.1.01/0.0/0.0/18-069/0009855.

References

- [1] M. E. Cornejo, D. Lobo, and J. Medina, "On the solvability of bipolar max-product fuzzy relation equations with the standard negation," *Fuzzy Sets and Systems*, vol. 410, pp. 1–18, 2021.
- [2] G. Xiao, T. Zhu, Y. Chen, and X. Yang, "Linear searching method for solving approximate solution to system of max-min fuzzy relation equations with application in the instructional information resources allocation," *IEEE Access*, vol. 7, pp. 65019–65028, 2019.
- [3] E. Turunen, "Necessary and sufficient conditions for the existence of solution of generalized fuzzy relation equations $A \Leftrightarrow X = B$," *Information Sciences*, vol. 536, pp. 351–357, 2020.
- [4] V. Sellam, N. Kannan, and H. A. Basha, "An effective fuzzy logic based clustering scheme for edge-computing based internet of medical things systems," in *Cognitive Internet of Medical Things for Smart Healthcare*, pp. 105–116, Springer, Cham, Switzerland, 2021.
- [5] S. K. Malchi, S. Kallam, F. Al-Turjman, and R. Patan, "A trust-based fuzzy neural network for smart data fusion in internet of things," *Computers and Electrical Engineering*, vol. 89, Article ID 106901, 2021.
- [6] M. Jamshidi, R. Farhadi, M. Jamshidi, Z. Shamsi, and S. Naseh, "Using a soft computing method for impedance modelling of li-ion battery current," *International Journal of Advanced Intelligence Paradigms*, vol. 16, no. 1, pp. 18–29, 2020.
- [7] T. Vaiyapuri, V. S. Parvathy, V. Manikandan, N. Krishnaraj, D. Gupta, and K. Shankar, "A novel hybrid optimization for cluster-based routing protocol in information-centric wireless sensor networks for iot based mobile edge computing," *Wireless Personal Communications*, vol. 116, pp. 1–24, 2021.
- [8] S. Roshani, M. B. Jamshidi, F. Mohebi, and S. Roshani, "Design and modeling of a compact power divider with squared resonators using artificial intelligence," *Wireless Personal Communications*, vol. 117, no. 3, pp. 2085–2096, 2021.
- [9] M. B. Jamshidi, S. Roshani, J. Talla, S. Roshani, and Z. Peroutka, "Size reduction and performance improvement of a microstrip Wilkinson power divider using a hybrid design technique," *Scientific Reports*, vol. 11, no. 1, pp. 1–15, 2021.
- [10] M. B. Jamshidi, M. Gorjiankhanzad, A. Lalbakhsh, and S. Roshani, "A novel multiobjective approach for detecting money laundering with a neuro-fuzzy technique," in *Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 454–458, IEEE, Alberta, Canada, December 2019.
- [11] K. Purushothaman and V. Nagarajan, "Evolutionary multi-objective optimization algorithm for resource allocation using deep neural network in 5G multi-user massive mimo," *International Journal of Electronics*, pp. 1–20, 2020.
- [12] M. B. Jamshidi and N. Alibeigi, "Neuro-fuzzy system identification for remaining useful life of electrolytic capacitors," in *Proceedings of the 2017 2nd International Conference on System Reliability and Safety (ICSRS)*, pp. 227–231, IEEE, Milan, Italy, December 2017.
- [13] H. Zhang, Z. Chen, J. Wu, and K. Liu, "FRRF: a fuzzy reasoning routing-forwarding algorithm using mobile device similarity in mobile edge computing-based opportunistic mobile social networks," *IEEE Access*, vol. 7, pp. 35874–35889, 2019.
- [14] C. Wang, N. Tian, Z. Ji, and Y. Wang, "Multi-objective fuzzy flexible job shop scheduling using memetic algorithm," *Journal of Statistical Computation and Simulation*, vol. 87, no. 14, pp. 2828–2846, 2017.
- [15] B. Gupta, M. Tiwari, and S. Singh Lamba, "Visibility improvement and mass segmentation of mammogram images using quantile separated histogram equalisation with local contrast enhancement," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 73–79, 2019.
- [16] H. S. Basavegowda and G. Dagnev, "Deep learning approach for microarray cancer data classification," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 22–33, 2020.
- [17] S. Ghosh, P. Shivakumara, P. Roy, U. Pal, and T. Lu, "Graphology based handwritten character analysis for human behaviour identification," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 55–65, 2020.
- [18] M. Jamshidi, A. Lalbakhsh, J. Talla et al., "Artificial intelligence and covid-19: deep learning approaches for diagnosis and treatment," *IEEE Access*, vol. 8, pp. 109581–109595, 2020.
- [19] M. B. Jamshidi, A. Lalbakhsh, J. Talla et al., "Deep learning techniques and covid-19 drug discovery: fundamentals, state-of-the-art and future directions," *Emerging Technologies During the Era of COVID-19 Pandemic*, vol. 348, p. 9, 2021.
- [20] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks* Springer, Berlin, Germany, 2019.
- [21] A. Choudhary, M. Nizamuddin, M. Zadoo, and V. K. Sachan, "Multi-objective optimization framework complying IEEE 802.15.6 communication standards for wireless body area networks," *Wireless Networks*, vol. 26, no. 6, pp. 4339–4362, 2020.
- [22] B. V. Natesha and R. M. R. Guddeti, "Adopting elitism-based Genetic algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, Article ID 102972, 2021.
- [23] J. G. Andrews, S. Buzzi, W. Choi et al., "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.
- [24] S. K. Goudos, P. D. Diamantoulakis, and G. K. Karagiannidis, "Multi-objective optimization in 5G wireless networks with massive mimo," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2346–2349, 2018.

- [25] Y. Q. Hei, C. Zhang, and G. M. Shi, "Trade-off optimization between energy efficiency and spectral efficiency in large scale mimo systems," *Energy*, vol. 145, pp. 747–753, 2018.
- [26] G. Lee and Y. Sung, "A new approach to user scheduling in massive multi-user mimo broadcast channels," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1481–1495, 2017.
- [27] H. R. Chi and A. Radwan, "Multi-objective optimization of green small cell allocation for IoT applications in smart city," *IEEE Access*, vol. 8, pp. 101903–101914, 2020.
- [28] L. Liu, Z. Chang, X. Guo, and T. Ristaniemi, "Multi-objective optimization for computation offloading in mobile-edge computing," in *Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 832–837, IEEE, Heraklion, Greece, July 2017.
- [29] M. M. Bourke and D. G. Fisher, "Solution algorithms for fuzzy relational equations with max-product composition," *Fuzzy Sets and Systems*, vol. 94, no. 1, pp. 61–69, 1998.
- [30] S.-C. Fang, *Linear Optimization and Extensions: Theory and Algorithms*, ACM Digital Library, New York, NY, USA, 1993.
- [31] E. Khorram and H. Zarei, "Multi-objective optimization problems with fuzzy relation equation constraints regarding max-average composition," *Mathematical and Computer Modelling*, vol. 49, no. 5-6, pp. 856–867, 2009.
- [32] S. Narayanan and S. Azarm, "On improving multiobjective genetic algorithms for design optimization," *Structural Optimization*, vol. 18, no. 2, pp. 146–155, 1999.
- [33] M. Li, S. Yang, and X. Liu, "Pareto or non-pareto: bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2015.
- [34] C. Mariano and E. Morales, "A new distributed reinforcement learning algorithm for multiple objective optimization problems," in *Advances in Artificial Intelligence*, pp. 290–299, Springer, Berlin, Germany, 2000.
- [35] S. Dutta and K. N. Das, "A survey on pareto-based EAs to solve multi-objective optimization problems," *Advances in Intelligent Systems and Computing*, vol. 817, pp. 807–820, 2019.
- [36] E. Khorram, R. Ezzati, and Z. Valizadeh, "Linear fractional multi-objective optimization problems subject to fuzzy relational equations with a continuous archimedean triangular norm," *Information Sciences*, vol. 267, pp. 225–239, 2014.
- [37] A. A. Molai and E. Khorram, "Minimizing a linear objective function subject to fuzzy relation equations constraints with Max-Hamacher product composition," *Asia Pacific Journal of Operational Research*, vol. 25, no. 2, pp. 243–266, 2008.
- [38] H. Tamaki, H. Kita, and S. Kobayashi, "Multi-objective optimization by genetic algorithms: a review," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 517–522, IEEE, Nagoya, Japan, May 1996.
- [39] J. Loetamonphong and S.-C. Fang, "Optimization of fuzzy relation equations with max-product composition," *Fuzzy Sets and Systems*, vol. 118, no. 3, pp. 509–517, 2001.
- [40] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Science & Business Media, Berlin, Germany, 2013.
- [41] J. Magalhaes-Mendes, "A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem," *WSEAS Transactions on Computers*, vol. 12, no. 4, pp. 164–173, 2013.
- [42] A. Thapar, D. Pandey, and S. K. Gaur, "Satisficing solutions of multi-objective fuzzy optimization problems using genetic algorithm," *Applied Soft Computing*, vol. 12, no. 8, pp. 2178–2187, 2012.
- [43] Y. Lu and X. Zheng, "6G: a survey on technologies, scenarios, challenges, and the related issues," *Journal of Industrial Information Integration*, vol. 19, Article ID 100158, 2020.