Int. J. Nonlinear Anal. Appl. 14 (2023) 3, 19–32
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2023.28390.3876

A modified imperialist competitive algorithm for solving nonlinear programming problems subject to mixed fuzzy relation equations

Ali Abbasi Molai*, Hassan Dana Mazraeh

School of Mathematics and Computer Sciences, Damghan University, P.O. Box 36715-364, Damghan, Iran

(Communicated by Javad Damirchi)

Abstract

The mixed fuzzy relation programming with a nonlinear objective function and two operators of max-product and max-min composition is studied in this paper. Its feasible domain structure is investigated and some simplification procedures are presented to reduce the dimension of the original problem. We intend to modify the assimilation and revolution operators of the imperialist competitive algorithm in order to prevent the generation of infeasible solutions. The modified imperialist competitive algorithm (MICA) is compared with a real-value genetic algorithm to solve the original problem. Several test problems are presented to compare its performance with respect to the performance of the genetic algorithm. Their results show the superiority of the proposed algorithm over the genetic algorithm.

Keywords: Mixed fuzzy relation equation; Max-product and Max-min operators; Nonlinear optimization; Imperialist Competitive Algorithm
2020 MSC: Primary 90C70; Secondary 90C26

1 Introduction

Since Fuzzy Relation Equations (FREs) were firstly proposed by Sanchez [25], many researchers have studied FREs from theoretical and applied aspects [21, 22, 35, 23, 24, 19]. The theory aspects contain the solvability criteria, the solution set structure, the minimal solutions, the maximum solution, the consistency, NP-hardness, and their associated algorithms [23, 18, 11, 22, 20]. The applied aspects can be considered in various areas such as fuzzy decision-making, fuzzy symptom diagnosis, textile engineering, image compression, and reconstruction, and fuzzy optimization [21, 22, 35]. Recently, its applications can be seen in the wireless communication [29], supply chain [33], and the data transmission modelling in BitTorrent-like Peer-to-Peer (P2P) file sharing systems [30, 31, 32].

One of the most interesting topics in FREs is the objective function optimization subject to a system of FREs. The linear optimization problem provided to the max-min FRE system has firstly been investigated by Fang and Li [5] motivated in the textile engineering. Some researchers tried to improve their approach by proposing a suitable upper bound or some rules for simplifying the problem [26, 27, 28, 7]. Comprehensive overview on FREs and fuzzy relation programming has been presented by Li and Fang in [14, 15]. In some applications which the human judgment plays

*Corresponding author
Email addresses: a_abbasimolai@du.ac.ir (Ali Abbasi Molai), dana@du.ac.ir (Hassan Dana Mazraeh)

Received: September 2022 *Accepted:* January 2023

a main role, we need to variables that should satisfy FRE or FRI constraints with two composition operators, simultaneously. Such systems are called Mixed Fuzzy Relation Equations (MFREs) or Mixed Fuzzy Relation Inequalities (MFRIs), respectively. The problem of minimizing a linear objective function provided to MFRIs with two different operators of $Max-T_1$ and $Max-T_2$ was studied in [1]. The structure of its feasible domain was determined. Sufficient and necessary conditions were suggested for its solution existence. Some rules were given to reduce the size of the problem. An algorithm was designed to solve the problem based on its structure and reduction procedures. However, algorithms have been proposed to solve the nonlinear programming problems subject to FREs with special forms of objective functions [34, 38, 39, 10]. Hence, designing novel algorithms for the resolution of nonlinear programming problems with FREs, in a general case, has been developing very slowly. Lu and Fang [17] firstly focused on the resolution of such nonlinear problems. Since the resolution of these problems is a NP-hard problem in a general case, we cannot use the traditional nonlinear optimization methods. Hence, they designed a Genetic Algorithm (GA) for finding its optimal solution with regard to the structure of its feasible domain. They selected the individuals of the initial population from its feasible domain and kept them within the feasible domain during the mutation and crossover operations. The nonlinear optimization problem with the max-average operator was studied by Khorram and Hassanzadeh [12]. A modified GA was suggested to solve the problem. In the algorithm, they changed some of its components to solve the problem. Moreover, Hassanzadeh et al.[9] investigated the problem with the max-product operator. They proposed a GA to obtain a good approximate optimal solution for convex or non-convex feasible domain and evaluated its performance by several test problems. Recently, Li et al. [13] studied a special type of nonlinear optimization problem with a non-differential objective function provided to a set of constraints of MFREs with the max-min and the max-product operators. Several properties of the problem were investigated and an algorithm was designed based on the useful properties. A similar problem to the above problem with the max-min and the max-average operators was considered and some of its properties were expressed [6]. Then, an algorithm was designed by them to solve the problem. However, the objective functions of the problems in Refs. [13, 6] have a special form and the algorithms in [13, 6] cannot be applied to solve the problems in a general case. Therefore, we are motivated to design efficient evolutionary algorithms to solve the problems in a general case. In each iteration of the evolutionary algorithms, some candidate solutions are generated based on the information of the candidate solutions in the previous iteration. New generated candidate solutions may be infeasible because of the nature of some problems such as non-convexity. In this case, some techniques are applied to overcome the infeasible candidate solutions such as repairing infeasible candidate solutions, penalizing values related to infeasible candidate solutions, and eliminating infeasible candidate solutions. One of the most powerful evolutionary algorithms for resolution of various optimization problems was suggested by Atashpaz Gargari and Lucas [3] called Imperialist Competitive Algorithm (ICA). In this paper, the nonlinear optimization problem with an arbitrary objective function subject to a system of MFREs with the max-product and max-min operators is investigated. This problem is an extension of the proposed models by Lu and Fang [17] and Hassanzadeh et al. [9]. We express some properties of the system of MFREs and present some simplification procedures to reduce the size of the dimensions of the problem. A new algorithm is designed to solve the problem. The algorithm is based on Modified Imperialist Competitive Algorithm (MICA). The original ICA generates the initial empires and moves the colonies toward their imperialists (assimilation). Random changes occur in the characteristics of each colony with a determinate probability (revolution). It then updates positions of the imperialists and computes total power of an empire. In the next stage, the imperialistic competition begins and all the empires try to take possession of the colonies of other empires and eliminate the powerless empires. After some imperialist competitions, all the empires except the most powerful ones will collapse. In this case, the algorithm stops [3]. The original form of this algorithm does not take the feasible domain into account to generate new candidate solution. Therefore, some modifications have been applied on the original operators of the ICA to prevent infeasible solutions from being generated. These modifications help the algorithm to search within the feasible domain so that the execution time of the proposed algorithm decreases considerably. In fact, the innovation behind the proposed method is to change the main operators of the algorithm such that this algorithm is compatible with the problems described in this paper. We intend to change the assimilation and revolution operators of the ICA in order to prevent the generation of infeasible solutions and propose a MICA. Some test problems are given to show its performance by comparing it with the Real-Value GA (RVGA) proposed in [4]. The structure of paper is as follows. In the second section, the nonlinear optimization problem subject to mixed fuzzy relation equations with two operators of max-min and max-product composition is formulated and some simplification procedures are presented to reduce the dimension of the original problem. The third section proposes a MICA to solve the problem. Section 4 presents some test problems to illustrate its results and performance. Section 5 provides the conclusions.

2 The problem of nonlinear objective function optimization subject to MFRE constraints

This section contains two subsections. In the first subsection, we formulate the problem of nonlinear objective function optimization subject to MFRE constraints. In the second subsection, we express some useful properties of its feasible domain. Using them, some simplification procedures are proposed to reduce the size of dimensions of the problem.

2.1 Formulation of the problem

The problem of nonlinear objective function optimization subject to the system of MFRE constraints with two operators of max-min and max-product compositions is formulated as follows:

$$\begin{aligned} &Min(or Max) \quad z = f(x), && (2.1)\\ &s.t. \qquad A \circ x = b, && (2.2)\\ &\qquad\quad B \bullet x = d, && (2.3)\\ &\qquad\quad x \in [0,1]^n, && (2.4) \end{aligned}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$, $A = [a_{ij}]_{m\times n}$, $B = [b_{ij}]_{p\times n}$, $b = [b_i]_{m\times 1}$, and $d = [d_i]_{p\times 1}$. The components of the matrices denote fuzzy degrees in $[0, 1]$. Now, we want to find a vector x such that satisfies the constraints and minimizes (or maximizes) the objective function $f(x)$. Also, the notation of $\circ$ and $\bullet$ denote the max-min and max-product composition operator, respectively. The constrains (2.2)-(2.4) are equivalent to the following system:

$$\begin{aligned} &Max\{a_{i1} \wedge x_1, a_{i2} \wedge x_2, \dots, a_{in} \wedge x_n\} = b_i, \quad \forall i \in I, && (2.5)\\ &Max\{b_{i1} \times x_1, b_{i2} \times x_2, ..., b_{in} \times x_n\} = d_i, \quad \forall i \in K, && (2.6)\\ &\quad x_j \in [0,1], \quad \forall j \in J, && (2.7) \end{aligned}$$

where $I = \{1, 2, ..., m\}$, $J = \{1, 2, ..., n\}$, and $K = \{1, 2, ..., p\}$. Also, the notation of $\wedge$ denotes the minimum operator. We are now ready to investigate the structure of its feasible domain.

2.2 The solution set of system (2.5)-(2.7)

As it is well-known, the solution set of system (2.5)-(2.7) is determined by the maximum solution and a finite number of minimal solutions of system (2.5)-(2.7). The maximum solution of the sub-system (2.5) and (2.7), i.e., $\overline{x} = (\overline{x}_1, \dots, \overline{x}_n)$, is determined as follows [5]:

$$\overline{x} = A@b = [\mathop{\wedge}_{i=1}^{m}(a_{ij}@b_i)]_{j\in J}, \tag{2.8}$$

where

$$a_{ij}@b_i = \begin{cases} 1 & a_{ij} \le b_i, \\ b_i & a_{ij} > b_i, \end{cases}$$

The maximum solution of sub-system (2.6) and (2.7), i.e., $\overline{\overline{x}} = (\overline{\overline{x}}_1, \dots, \overline{\overline{x}}_n)$ is given by the following relation [16]:

$$\overline{\overline{x}} = B \circledast d = [\mathop{\wedge}_{i=1}^{p}(b_{ij} \circledast d_i)]_{j\in J}, \tag{2.9}$$

where

$$b_{ij} \circledast d_i = \begin{cases} 1 & b_{ij} \le d_i, \\ \frac{d_i}{b_{ij}} & b_{ij} > d_i, \end{cases}$$

with regard to the relations (2.8) and (2.9), the maximum solution of system (2.5)-(2.7), simultaneously, i.e., $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$, is as follows:

$$\hat{x}_j = \overline{x}_j \wedge \overline{\overline{x}}_j, \quad \forall j \in J. \tag{2.10}$$

We can easily verify the consistency of the system (2.5)-(2.7) by checking feasibility of vector $\hat{x}$ in the system (2.5)-(2.7). If vector $\hat{x}$ satisfies the system (2.5)-(2.7), then the system is consistent and its feasible domain is nonempty. Otherwise, its feasible domain is empty.

The solution set of the feasible domain of the problem (2.1)-(2.4) is determined by the maximum solution and a finite number of minimal solutions. The computation of the maximum solution with regard to formulae (2.8)-(2.10) is not difficult. However, the main difficulty is finding all the minimal solutions. We now present some simplification procedures to reduce the size of the dimensions of the problem of (2.1)-(2.4).

Lemma 2.1. Consider the system (2.5)-(2.7). Suppose that there exist $i \in I$ and $j \in J$ such that $a_{ij} < b_i$, then the component a_{ij} can be converted to zero in the system.

Proof . If $a_{ij} < b_i$, then in the i^{th} equation, the relation of $Min\{a_{ij}, x_j\}$ is less than b_i. Therefore, the i^{th} equation cannot be satisfied by the relation. Hence, we can set zero instead of a_{ij}. This topic has no effect on the system (2.5)-(2.7). □

With regard to Lemma 2.1, if $a_{ij} < b_i$, then we can easily remove the relation $Min\{a_{ij}, x_j\}$ and reduce the dimensions of the system (2.5)-(2.7).

Lemma 2.2. Consider the system (2.5)-(2.7). Suppose that there exist $i \in I$ and $j \in J$ such that $b_{ij} < d_i$, then the component b_{ij} can be converted to zero in the system.

Proof . If $b_{ij} < d_i$, then in the i^{th} equation, the relation of $b_{ij}.x_j$ is less than d_i due to $b_{ij}, x_j, d_i \in [0, 1]$. Therefore, the i^{th} equation cannot be satisfied by the relation. Hence, we can set zero instead of b_{ij}. This topic has no effect on the system (2.5)-(2.7). □

With regard to Lemma 2.2, if $b_{ij} < d_i$, then we can easily delete the relation $b_{ij} \times x_j$ and reduce the dimensions of the system (2.5)-(2.7). Also, with attention to Lemmas 2.1 and 2.2, the system (2.5)-(2.7) is simplified and the simplified problem (2.1)-(2.4) can be solved by MICA in the next section.

3 Modified imperialist competitive algorithm for resolution of the simplified problem (2.1)-(2.4)

In this section, we modify the original ICA in order to prevent from the generation of infeasible candidate solutions by changing some operators of the original imperialist competitive algorithm. Hence, we briefly explain the original imperialist competitive algorithm in Subsection 3.1. We then design a modified imperialist competitive algorithm to solve the simplified problem (2.1)-(2.4) in Subsection 3.2.

3.1 The Original Imperialist Competitive Algorithm

Imperialist Competitive Algorithm (ICA) is an evolutionary algorithm for optimization inspired by the imperialistic competition. In this algorithm, a population of individuals is called a country. The initial population in ICA is created with a random generation of initial countries. These countries are divided to two groups called imperialists and colonies based on their power. Each imperialist possesses some colonies and these colonies form an empire. When the empires compete with each other, they try to acquire more colonies. In this competition, some empires fail and lost their colonies and the powerful empires become much stronger. Hence, the weak empires collapse and all the countries become colonies of an empire [3]. In the optimization problems, the power of an imperialist depends on the objective function. In the maximization and minimization problems, it is proportional to the objective function value and its inverse, respectively. According to this point, the colonies related to each imperialist are specified and the position of the imperialists is updated. In the next step, the empires compete with each other to possess the colonies of other empires. ICA consists of four main steps as initialization, assimilation, competition, and convergence. The steps of imperialist competitive algorithm are presented below.

Algorithm 1. The Original Imperialist Competitive Algorithm [3].

Step 1. Generate Initial Empires: The position of the i^{th} country in a N_{var}-dimensional optimization problem is as: Country(i).vector = $[x_1, x_2, x_3, ..., x_{N_{var}}]$, $i = 1, 2, ..., N_{cnt}$, where N_{cnt} is total number of the countries. The cost of the i^{th} country is as:

Country(i).cost=Fitness-function(country(i).vector)

N_{imp}: The number of the most powerful countries to create empires,

N_{col}: The number of remaining countries that will be the colonies of the empires.

The initial number of colonies of an empire is directly proportional to its power and it is specified by *"Roulette Wheel"*. N_{col} vectors are generated from feasible vectors as initial countries.

Step 2. Assimilation: The colonies move toward their imperialists. Their positions in the n^{th} empire are updated as:

$$\begin{aligned}&\text{New country(i).vector}_n = \text{country(i).vector}_n + \\ &(rand * \alpha + (\text{Imperialist.vector}_n + \text{country(i).vector}_n))\end{aligned}$$

where country(i).vector$_n$: The position of the i^{th} colony in the n^{th} imperialist,

$rand$: A random vector which its components belong to $[-1, 1]$, and $\alpha = 0.1$,

Imperialist$_n$: The position of the n^{th} imperialist.

Step 3. Revolution: This stage denotes the random changes in the characteristics of each country with probability P_r.

Step 4. Position exchange between a colony and imperialist: In the previous steps, a colony may obtain a better position than the imperialist. Then their positions should be exchanged.

Step 5. Computing total power of an empire: The total power of the n^{th} empire (TC_n) is computed based on the power of the imperialist and its colonies as follows:

$$TC_n = \text{Cost(imperialist)} + \xi \times \text{mean}\{\text{Cost((colonies of empire))}\}$$

where ξ is a real number less than 1.

Step 6. Imperialistic Competition: In this step, all the empires compete with each other to possess the colonies of other empires. In this competition, some the weakest colonies are possessed by more powerful empires. In this paper, the competition is implemented by *"Roulette Wheel"* based on total powers of imperialist.

Step 7. Eliminating the weak empires: In the imperialistic competition, the weak empires will collapse. In this paper, the collapse mechanism occurs for an empire when it loses all of its colonies.

Step 8. Convergence: In this step, the most powerful empire only remains. The other empires will collapse and all of the countries under their possession are converted to colonies of the empire. Hence, the positions of all the colonies become the same. In this case, the algorithm stops.

The stages of the OICA have been illustrated in Figure 1 [3, 2, 37, 36]. Therefore, we summarize the steps of the OICA in Figure 2 as a flowchart. However, there are some drawbacks in the OICA to solve optimization problems with the non-convex constraints. In these problems, if the colonies move toward their relevant imperialists, some infeasible solutions may occur. To overcome this difficulty, we modify the OICA such that it prevents the generation of infeasible solutions. To do this, we change the *Assimilation* and the *Revolution* operators, properly. The main idea has been mentioned in Subsection 3.2.

3.2 Modified Imperialist Competitive Algorithm (MICA) for resolution of the simplified problem (2.1)-(2.4)

In this subsection, we intend to modify the OICA for prevention of generation of infeasible solutions in the optimization problems with the non-convex constrains. To do this, we change *Assimilation* and *Revolution* operators as follows:

Revolution: In the *Revolution* steps, a random vector is added to each colony with probability $P_{revolution}$. We repeat *Revolution* steps of the OICA to generate a new colony until it becomes a feasible solution. The pseudo-code of these steps are as follows:

Revolution(Empire)

For each colony of the empire do
  Pick a random value r uniformly from [0,1]

If ($r \leq P_{revolution}$) then
While (until new colony is not feasible) do
$New \quad country(i).vector = randvector + country(i).vector$
EndWhile
EndIf
EndFor
EndRevolution

Assimilation: In the non-convex optimization problems, if the colonies move toward their relevant imperialists like the OICA, some infeasible solutions may occur. Therefore, we repeat *Assimilation* steps of the OICA to prevent the generation of infeasible solutions until a feasible solution is generated. The pseudo-code of these steps are as follows:

Assimilation(Empire)
For each colony of the empire do
While(until new colony is not feasible) do
New country(i).vector=country(i).vector+(rand $* \alpha +$ (Imperialist.vector+country(i).vector))
EndWhile
EndFor
EndAssimilation

Figure 3 illustrates the main idea of the *Assimilation* steps of the MICA.

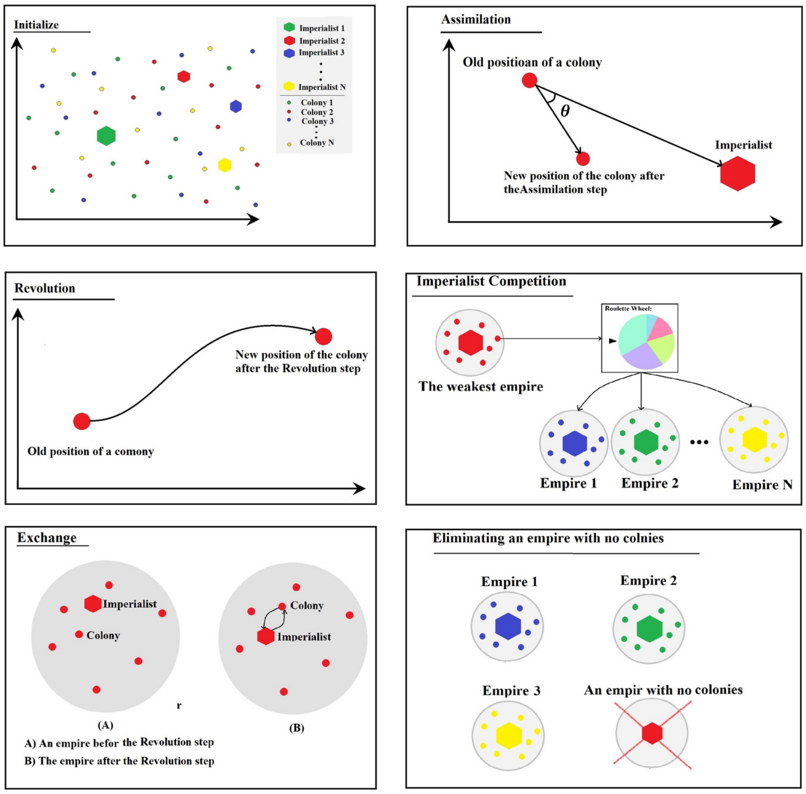

Figure 1: Steps of the Original Imperialist Competitive Algorithm.

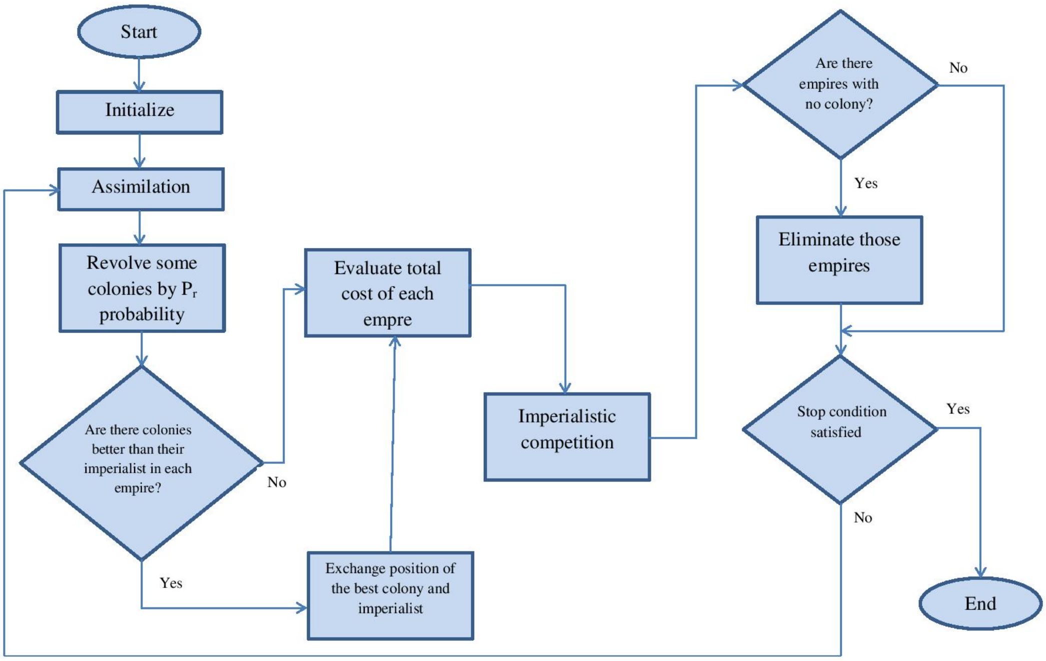

Figure 2: Flowchart of the Original Imperialist Competitive Algorithm.

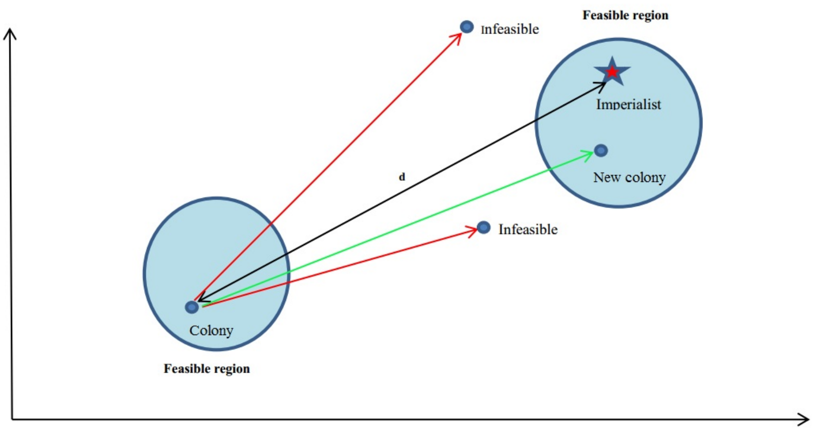

Figure 3: Modified Assimilation.

Figure 4 illustrates the main idea of *Revolution* steps of the MICA.

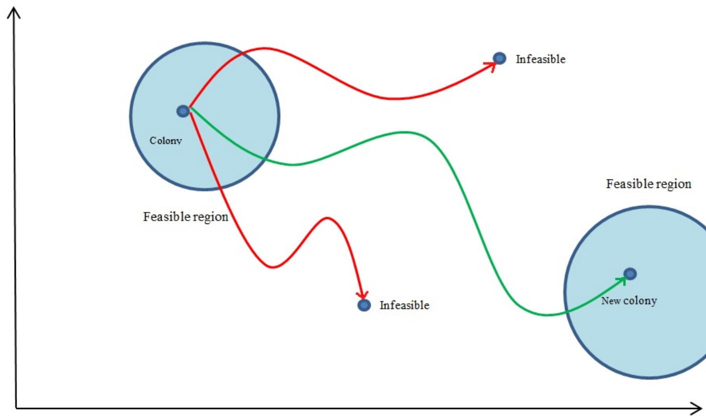

Figure 4: Modified Revolution.

Now, we present some numerical examples and solve them by the proposed RVGA in [4] and MICA to show their performances. Their results are also compared to each other. First of all, it is necessary to remind that the parameters of the RVGA are as follows:

Parameters	Value
Representation	Real-valued vectors
Population size	20
Initialization	Random
Parent selection method	Roulette wheel
Recombination	N-points crossover
Recombination probability	100%
Mutation	Random change
Mutation probability	80%
Survival selection	Replace worst
Number of offspring	2
Termination condition	Number of iterations

Table 1: The parameters of RVGA

The test problems are taken from Ref. [8]. Also, the software used to solve these examples is MATLAB, in the next section .

4 Test problems

Example 4.1. Consider problem (2.1)-(2.4) as minimization where

$$f(x_1, x_2, x_3) = 3000x_1 + 1000x_1^3 + 2000x_2 + 666.667x_2^3, \tag{4.1}$$

$$A = \begin{pmatrix} 0.34 & 0.28 & 0.64 \\ 0.5 & 0.88 & 0.18 \\ 0.71 & 0.21 & 0.3 \end{pmatrix}, b = (0.28, 0.3, 0.21)^T, \tag{4.2}$$

$$B = \begin{pmatrix} 0.15 & 0.4 & 0.26 \\ 0.32 & 0.6 & 0.73 \\ 0.16 & 0.5 & 0.42 \end{pmatrix}, d = (0.12, 0.18, 0.15)^T, \tag{4.3}$$

$$x \in [0, 1]^3. \tag{4.4}$$

We firstly display the feasible solution set by Figure 5. Then, the RVGA is run for Example 4.1. Its results are presented for $50, 100, 150, 200, 250,$ and 300 iterations in Table 2. Also, we run MICA for this example. Its results are given in Table 3 for $50, 100, 150, 200, 250$, and 300 iterations, respectively. We can now compare the obtained results from RVGA and MICA for test problem 1. This work has been done in Figure 6. Figure 6 shows that the performance of MICA is better and more efficient with respect to the performance of the proposed RVGA in [4].

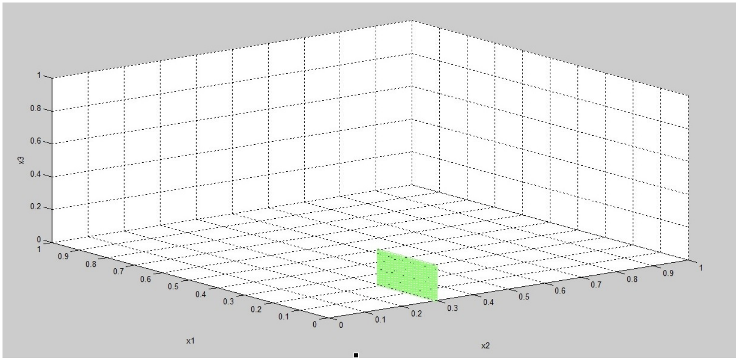

Figure 5: The green region displays the feasible solution set for test problem 1

Iterations	x_1	x_2	x_3	$f(x)$
50	0.03	0.30	0.19	708.02
100	0.02	0.30	0.16	678.01
150	0.02	0.03	0.21	678.01
200	0.01	0.03	0.16	648.00
250	0.01	0.30	0.12	648.00
300	0	0.30	0.20	618.00

Table 2: The performance of proposed RVGA with different iterations for test problem 1.

Iterations	x_1	x_2	x_3	$f(x)$
50	0.02	0.30	0.12	678.01
100	0.01	0.30	0.19	648.00
150	0	0.30	0.02	618.00
200	0.01	0.30	0.11	648.00
250	0	0.30	0.12	618.00
300	0	0.30	0.17	618.00

Table 3: The performance of proposed MICA with different iterations for test problem 1.

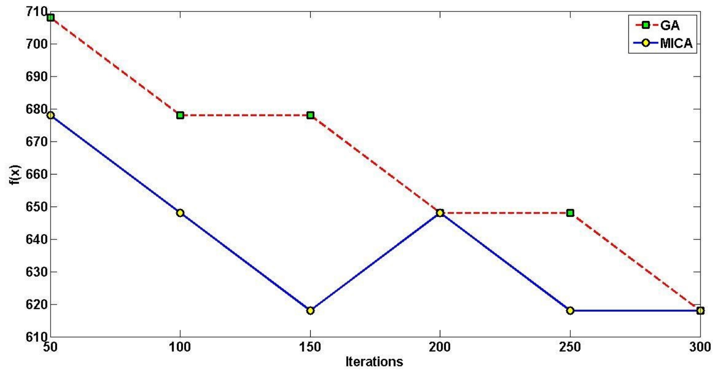

Figure 6: Performance comparison of proposed RVGA and MICA on test problem 1

Example 4.2. Consider problem (2.1)-(2.4) as minimization where

$$f(x_1,x_2,x_3,x_4) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4, \tag{4.5}$$

$$A = \begin{pmatrix} 0.5176 & 0.2278 & 0.8993 & 0.9858 \\ 0.1370 & 0.4585 & 0.6334 & 0.2790 \\ 0.4093 & 0.7399 & 0.0313 & 0.3039 \end{pmatrix}, b = (0.7208, 0.6334, 0.4725)^T, \tag{4.6}$$

$$B = \emptyset, \ and \ d = \emptyset, \tag{4.7}$$

$$x \in [0,1]^4. \tag{4.8}$$

According to Lemma 2.1, the simplification procedure is implemented on Example 4.1 as follows: Since $a_{11}, a_{12} < b_1$, the components of a_{11} and a_{12} can be converted to zero. Also, components a_{21}, a_{22}, and a_{24} are less than b_2. So, the components of a_{21}, a_{22}, and a_{24} can be converted to zero. Similarly, components a_{31}, a_{33}, and a_{34} are less than b_3. Hence, the components of a_{31}, a_{33}, and a_{34} can be converted to zero. On the other hand, the maximum solution of the feasible domain is as: $\hat{x} = \bar{x} = (1, 0.4725, 0.7208, 0.7208)$. Therefore, the simplified problem is as follows:

$$Min \ \ f(x_1,x_2,x_3,x_4) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4,$$

$$s.t. \quad Max\{0.8993 \bigwedge x_3, 0.9858 \bigwedge x_4\} = 0.7208,$$

$$Max\{0.6334 \bigwedge x_3\} = 0.6334,$$

$$Max\{0.7399 \bigwedge x_2\} = 0.4725,$$

$$x_1 \in [0,1], x_2 \in [0, 0.4725], x_3 \in [0, 0.7208], x_4 \in [0, 0.7208].$$

Then, the RVGA is run for Example 4.2. Its results are presented for $50, 100, 150, 200, 250$, and 300 iterations in Table 4. Also, we run the MICA for this example. Its results are given in Table 5 for $50, 100, 150, 200, 250$, and 300 iterations, respectively. We can now compare the obtained results from RVGA and MICA for test problem 2. This work has been done in Figure 7. The problem is a minimization problem. Figure 7 shows that the performance of MICA is better and more efficient with respect to the performance of the proposed RVGA in [4].

Iterations	x_1	x_2	x_3	x_4	$f(x)$
50	0.0389	0.4725	0.7208	0.3221	24.4359
100	0.0318	0.4725	0.7208	0.5075	24.2487
150	0.0637	0.4725	0.7208	0.5172	24.4439
200	0.0189	0.4725	0.7208	0.4602	24.1054
250	0.0303	0.4725	0.7208	0.4297	24.1731
300	0.0155	0.4725	0.7208	0.3976	24.0898

Table 4: The performance of proposed RVGA with different iterations for test problem 2.

$Iterations$	x_1	x_2	x_3	x_4	$f(x)$
50	0.0172	0.4725	0.7208	0.4194	24.0864
100	0.0167	0.4725	0.7208	0.4505	24.0852
150	0.0194	0.4725	0.7208	0.4355	24.0981
200	0.0069	0.4725	0.7208	0.3825	24.0441
250	0.0113	0.4725	0.7208	0.4632	24.0634
300	0.0062	0.4725	0.7208	0.4018	24.0200

Table 5: The performance of proposed MICA with different iterations for test problem 2.

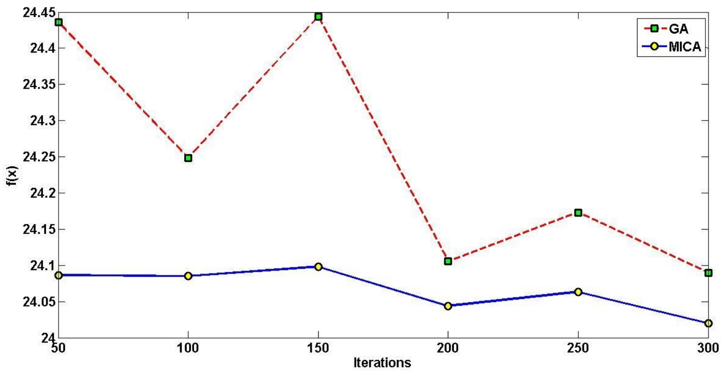

Figure 7: Performance comparison of proposed RVGA and MICA on test problem 2

Example 4.3. Consider problem (2.1)-(2.4) as minimization where

$$f(x_1,x_2,x_3,x_4,x_5) = 5x_1^{-0.2}x_2^{-0.3}x_3^2x_4^{-1}x_5^2 + 2x_1^{-0.2}x_2^{-1.5}x_3^2x_4^{-2}x_5, \tag{4.9}$$

$$A = \begin{pmatrix} 0.9 & 0.8 & 0.6 & 0.3 & 0.9 \\ 0.8 & 0.7 & 0.8 & 1 & 0.8 \\ 0.6 & 0.9 & 0.8 & 0.9 & 0.5 \\ 0.4 & 0.2 & 0.5 & 0.6 & 0.2 \\ 0.3 & 0.3 & 0.5 & 0.2 & 0.1 \\ 0.4 & 0.1 & 0.2 & 0.3 & 0.5 \end{pmatrix}, b = (0.8, 0.8, 0.5, 0.5, 0.4)^T, \tag{4.10}$$

$$B = \emptyset, d = \emptyset, \tag{4.11}$$

$$x \in [0,1]^n. \tag{4.12}$$

Then, the RVGA is run for Example 4.3. Its results are presented for $50, 100, 150, 200, 250$, and 300 iterations in Table 6. Also, we run the MICA for this example. Its results are given in Table 7 for $50, 100, 150, 200, 250$, and 300 iterations, respectively. We can now compare the obtained results from RVGA and MICA for test problem 3. This work has been done in Figure 8. The problem is a minimization problem. Figure 8 shows that the performance of MICA is better and more efficient with respect to the performance of the proposed RVGA in [4].

$Iterations$	x_1	x_2	x_3	x_4	x_5	$f(x)$
10	0.53	0.80	0.88	0.42	0.09	2.2611
20	0.80	0.53	0.94	0.22	0.02	2.4870
30	0.80	0.36	0.84	0.29	0.0200	1.9703
40	0.80	0.78	0.86	0.33	0.02	0.6648
50	0.80	0.65	0.85	0.47	0.04	0.8879
60	0.71	0.80	0.86	0.46	0	0

Table 6: The performance of proposed RVGA with different iterations for test problem 3.

$Iterations$	x_1	x_2	x_3	x_4	x_5	$f(x)$
10	0.8000	0.4800	0.8600	0.3500	0.0400	2.0695
20	0.8000	0.3700	0.8400	0.4700	0.0300	1.2077
30	0.8000	0.2300	0.8500	0.4000	0.0100	1.0029
40	0.8000	0.7600	0.9900	0.4500	0	0

Table 7: The performance of proposed MICA with different iterations for test problem 3.

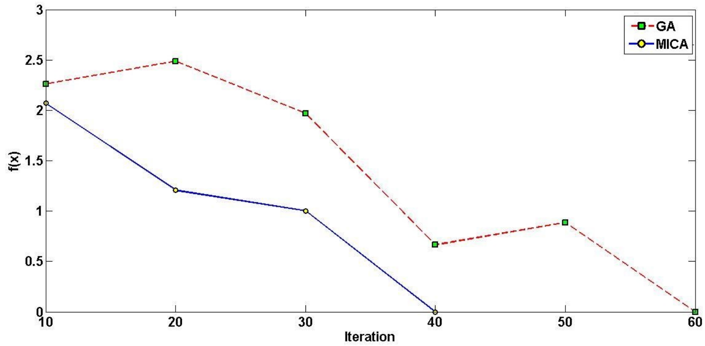

Figure 8: Performance comparison of proposed RVGA and MICA for test problem 3

5 Conclusions and future researches

The nonlinear programming problem subject to a mixed fuzzy relation equation system with two operators of max-product and max-min composition can be applied for real world applications such as wireless communication and BitTorrent-like Peer-to-Peer file sharing system. Some procedures were given to simplify the problem. A real-value genetic algorithm had been designed for its resolution in [4] where the N-points crossover and mutation operators were defined for the algorithm. The repair operator had been applied to prevent from generation of its infeasible solutions in the algorithm. In this paper, a modified imperialist competitive algorithm was proposed to solve the nonlinear problems. The algorithm modifies the assimilation and revolution operators in the original ICA to prevent the generation of infeasible solutions. The proposed MICA is compared with RVGA in [4] and some test problems are presented to compare them. The results show that the proposed MICA is more efficient with respect to the RVGA. In recent years, machine learning algorithms and deep neural networks have shown great capability in solving a wide range of problems including optimization problems. Investigating the capability of these methods to solve the problems described in this paper will be considered as a future research work. Also, we will focus on hybrid algorithms based on machine learning algorithms and meta heuristic algorithms to have an efficient algorithm.

References

[1] A. Abbasi Molai, *Linear optimization with mixed fuzzy relation inequality constraints using the pseudo-t-norms and its application*, Soft Comput. **19** (2015), 3009–3027.

[2] M. Abdellahi and M. Bahmanpour, *A novel technology for minimizing the synthesis time of nanostructured powders in planetary mills*, MATER. RES.-IBERO-AM. J. **17** (2014), no. 3, 781–791.

[3] E. Atashpaz-Gargari and C. Lucas, *Imperialist Competitive Algorithm: An algorithm for optimization inspired by imperialistic competition*, IEEE Congress on Evolutionary Computation, Singapore, 2007.

[4] H. Dana Mazraeh and A. Abbasi Molai, *Resolution of nonlinear optimization problems subject to bipolar max-min fuzzy relation equation constraints using genetic algorithm*, Iran. J. Fuzzy Syst. **15** (2018), no. 2, 109–131.

[5] S.-C. Fang and G. Li, *Solving fuzzy relation equations with a linear objective function*, Fuzzy Sets Syst. **103** (1999), 107–113.

[6] S. Feng, Y. Ma and J. Li, *A kind of nonlinear and non-convex optimization problems under mixed fuzzy relational equations constraints with max-min and max-average composition*, Proc. Eighth Int. Conf. Comput. Intel. Secur., November 17–18, 2012, pp. 154–158.

[7] F.-F. Guo and Z.-Q. Xia, *An algorithm for solving optimization problems with one linear objective function and finitely many constraints of fuzzy relation inequalities*, Fuzzy Optim. Decis. Mak. **5** (2006), no. 1, 33–47.

[8] W. Hock and K. Schittkowski, *Test Examples for nonlinear programming codes*, Lect. Notes Econ. Math. 187 Springer, New York, 1981.

[9] R. Hassanzadeh, E. Khorram, I. Mahdavi and N. Mahdavi-Amiri, *A genetic algorithm for optimization problems with fuzzy relation constraints using max-product composition*, Appl. Soft Comput. **11** (2011), 551–560.

[10] B. Hedayatfar, A. Abbasi Molai and S. Aliannezhadi, *Separable programming problems with the max-product fuzzy relation equation constraints*, Iran. J. Fuzzy Syst. **16** (2019), no. 1, 1–15.

[11] M. Higashi and G.J. Klir, *Resolution of finite fuzzy relation equations*, Fuzzy Sets Syst. **13** (1984), 65-–82.

[12] E. Khorram and R. Hassanzadeh, *Solving nonlinear optimization problems subjected to fuzzy relation equation constraints with max-average composition using a modified genetic algorithm*, Comput. Ind. Eng. **55** (2008), 1–14.

[13] J. Li, S. Feng and H. Mi, *A kind of nonlinear programming problem based on mixed fuzzy relation equations constraints*, Phys. Procedia **33** (2012), 1717–1724.

[14] P. Li and S.-C. Fang, *A survey on fuzzy relational equations, part I: classification and solvability*, Fuzzy Optim. Decis. Mak. **8** (2009), 179-–229.

[15] P. Li and S.-C. Fang, *On the resolution and optimization of a system of fuzzy relational equations with sup-T composition*, Fuzzy Optim. Decis. Mak. **7** (2008), 169-–214.

[16] J. Loetamonphong and S.-C. Fang, *Optimization of fuzzy relation equations with max-product composition*, Fuzzy Sets Syst. **118** (2001), 509–517.

[17] J. Lu and S.-C. Fang, *Solving nonlinear optimization problems with fuzzy relation equation constraints*, Fuzzy Sets Syst. **119** (2001), 1–20.

[18] A. Markovskii, *On the relation between equations with max-product composition and the covering problem*, Fuzzy Sets Syst. **153** (2005), 261—273.

[19] M. Miyakoshi and M. Shimbo, *Solutions of composite fuzzy relational equations with triangular norms*, Fuzzy Sets Syst. **16** (1985), 53-–63.

[20] K. Peeva, *Universal algorithm for solving fuzzy relational equations*, Ital. J. Pure Appl. Math. **19** (2006), 169–188.

[21] K. Peeva, *Composite fuzzy relational equations in decision making: chemistry*, Proc. 26th Summer School Appl. Math. Engin. Econ., Heron Press, Sozopol, 2001, pp. 260–264.

[22] K. Peeva and Y. Kyosev, *Fuzzy relational calculus: theory, applications and software*, World Scientific, New Jersey, 2004.

[23] W. Pedrycz, *Fuzzy relational equations with triangular norms and their resolutions*, BUSEFAL **11** (1982) 24—32.

[24] W. Pedrycz, *On generalized fuzzy relational equations and their applications*, J. Math. Anal. Appl. **107** (1985), 520-–536.

[25] E. Sanchez, *Resolution of composite fuzzy relation equations*, Inf. Control. **30** (1976), 38–48.

[26] Y.K. Wu and S.M. Guu, *Minimizing a linear function under a fuzzy max-min relational equation constraint*, Fuzzy Sets Syst. **150** (2005), 147-–162.

[27] Y.-K. Wu and S.-M. Guu, *A note on fuzzy relation programming problems with max-strict-t-norm composition*, Fuzzy Optim. Decis. Mak. **3** (2004), 271–278.

[28] Y.-K. Wu, S.-M. Guu and J.Y.-C. Liu, *An accelerated approach for solving fuzzy relation equations with a linear objective function*, IEEE Trans. Fuzzy Syst. **10** (2002), no. 4, 552–558.

[29] X.-P. Yang, X.-G. Zhou and B.-Y. Cao, *Latticized linear programming subject to max-product fuzzy relation*

inequalities with application in wireless communication, Inf. Sci. **358–359** (2016), 44–55.

[30] X.-P. Yang, H.-T. Lin, X.-G. Zhou and B.-Y. Cao, *Addition-min fuzzy relation inequalities with application in BitTorrent-like Peer-to-Peer file sharing system*, Fuzzy Sets Syst. **343** (2018), 126–140.

[31] X.-P. Yang, *Random-term-absent addition-min fuzzy relation inequalities and their lexicographic minimum solutions*, Fuzzy Sets Syst. **440** (2022), 42–61.

[32] X.-P. Yang, D.-H. Yuan and B.-Y. Cao, *Lexicographic optimal solution of the multi-objective programming problem subject to max-product fuzzy relation inequalities*, Fuzzy Sets Syst. **341** (2018), 92–112.

[33] X.-P. Yang, *Solutions and strong solutions of min-product fuzzy relation inequalities with application in supply chain*, Fuzzy Sets Syst. **384** (2020) 54–74.

[34] X. Yang, X. Zhou, B. Cao and Y. Hong, *Variable substitution method for solving single-variable term fuzzy relation geometric programming problem and its application*, Int. J. Uncertain. Fuzziness Knowlege-Based Syst. **27** (2019), 537—557.

[35] W.B. Vasantha Kandasamy and F. Smarandache, *Fuzzy relational maps and neutrosophic relational maps*, Vol. 3. Infinite Study, 2004.

[36] R. Yazdani, M. Hajimahmoodzadeh and H.R. Fallah, *Adaptive phase aberration correction based on imperialist competitive algorithm*, Appl. Opt. **53** (2014), no. 1, 132–140.

[37] M. Zanganeh and S.J. Mirabedini, *Comparing imperialist competitive algorithm with backpropagation algorithms for training feedforward neural network*, Int. J. Math. Comput. Sci. **14** (2015), 193–204.

[38] X.G. Zhou, B.Y. Cao and X.-P. Yang, *The set of optimal solutions of geometric programming problem with max-product fuzzy relational equations constraints*, Int. J. Fuzzy Syst. **18** (2016), 436—447.

[39] X.-G. Zhou, X.-P. Yang and B.-Y. Cao, *Posynomial geometric programming problem subject to max–min fuzzy relation equations*, Inf. Sci. **328** (2016), 15–25.