

A two - phase - ACO algorithm for solving nonlinear optimization problems subjected to fuzzy relational equations

A. Ghodousian ^{1,2} and S. Zal ²

¹Faculty of Engineering Science, College of Engineering, University of Tehran, P.O.Box 11365-4563, Tehran, Iran

²Department of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

a.ghodousian@ut.ac.ir, sara.zal@ut.ac.ir

Abstract

In this paper, we investigate nonlinear optimization problems whose constraints are defined as fuzzy relational equations (FRE) with max-min composition. Since the feasible solution set of the FRE is often a non-convex set and the resolution of the FREs is an NP-hard problem, conventional nonlinear approaches may involve high computational complexity. Based on the theoretical aspects of the problem, an algorithm (called FRE-ACO algorithm) is presented which benefits from the structural properties of the FREs, the ability of discrete ant colony optimization algorithm (denoted by ACO) to tackle combinatorial problems, and that of continuous ant colony optimization algorithm (denoted by ACO_R) to solve continuous optimization problems. In the current method, the fundamental ideas underlying ACO and ACO_R are combined and form an efficient approach to solve the nonlinear optimization problems constrained with such non-convex regions. Moreover, FRE-ACO algorithm preserves the feasibility of new generated solutions without having to initially find the minimal solutions of the feasible region or check the feasibility after generating the new solutions. FRE-ACO algorithm has been compared with some related works proposed for solving nonlinear optimization problems with respect to max-min FREs. The obtained results demonstrate that the proposed algorithm has a higher convergence rate and requires a less number of function evaluations compared to other considered algorithms.

Keywords: Continuous ant colony optimization, discrete ant colony optimization, fuzzy relational equations, max-min composition, nonlinear optimization.

1 Introduction

In this paper, we study the following nonlinear optimization problem in which the constraints are formed as the fuzzy relational equations defined by the minimum t-norm:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & A \varphi x = b \\ & x \in [0, 1]^n \end{aligned} \quad (1)$$

where $I = \{1, 2, \dots, m\}$ and $J = \{1, 2, \dots, n\}$. $A = (a_{ij})_{m \times n}$ is a fuzzy matrix such that $0 \leq a_{ij} \leq 1$ ($\forall i \in I$ and $\forall j \in J$), $b = (b_i)_{m \times 1}$ is a fuzzy vector such that $0 \leq b_i \leq 1$ ($\forall i \in I$) and “ φ ” is the max-min composition, i.e., $\varphi(x, y) = \min\{x, y\}$. So, if a_i ($i \in I$) denotes the i 'th row of matrix A , then the i 'th constraint of problem(1) can be expressed as $a_i \varphi x = b_i$ where $a_i \varphi x = \max_{j=1}^n \{\varphi(a_{ij}, x_j)\} = \max_{j=1}^n \{\min\{a_{ij}, x_j\}\}$.

The theory of fuzzy relational equations (FRE) with max-min composition was firstly proposed by Sanchez as a generalized version of Boolean relation equations [54]. Besides, he developed the application of max-min FRE in medical diagnosis [56]. Nowadays, it is well-known that many issues associated to the body can be treated as FRE problems

[44]. On the other hand, the fundamental result for FRE with max-product composition goes back to Pedrycz [46] and was further studied in [2, 35]. Since then, many theoretical improvements have been investigated [23, 26, 45, 51, 63] and various types of FRE with continuous triangular norms were used in applied topics [6, 43, 51]. In [49] the author demonstrated how problems of interpolation and approximation of fuzzy functions are converted with solvability of systems of FRE. Generally, when rules of inference are applied and their corresponding consequences are known, the problem of determining antecedents is simplified and mathematically reduced to solving an FRE [42].

The solvability identification and finding set of solutions are the primary, and the most fundamental, matter concerning the FRE problems. Di Nola et al. proved that the solution set for FRE (if it is nonempty), defined by max-min composition is often a non-convex set. This non-convex set is completely determined by one maximum solution and a finite number of minimal solutions [7]. In fact, the same results hold true for a more general t-norms instead of the minimum operator [18]. On the other hand, Chen and Wang [4] presented an algorithm for obtaining the logical representation of all minimal solutions and deduced that a polynomial-time algorithm with the ability to find all minimal solutions of FRE (with max-min composition) may not exist. Also, Markovskii showed that solving max-product FRE is closely related to the covering problem which is a type of NP-hard problem [40]. Additionally, Lin et al. [32] demonstrated that all systems of max-continuous t-norm fuzzy relational equations, for example, max-product, max-continuous Archimedean t-norm and max-arithmetic mean are essentially equivalent, because they are all equivalent to the set covering problem. Over the past decades, the solvability of FREs which are defined using different max-t compositions have been investigated by many researchers [15, 19, 47, 50, 53, 58, 64, 67].

The optimization problem related to FRE and FRI is one of the most interesting and on-going research topics among similar problems [3, 11, 12, 17, 18, 19, 20, 23, 24, 25, 28, 34, 56, 62, 66, 69]. For example, Fang and Li converted a linear optimization problem subjected to FRE constraints with max-min operation into an integer programming problem and solved it by branch and bound method using jump-tracking technique [12]. Wu et al. worked on improvement of the method employed by Fang and Li; this was done by decreasing the search domain and presented a simplification process by three rules which resulted from a necessary condition [65]. Chang and Shieh [3] presented new theoretical results concerning the same problem by improving an upper bound on the optimal objective value. In [27], an application of optimizing the linear objective with respect to max-min FRE constraints was employed for a minimum cost streaming media provider problem. Linear optimization problem was further investigated by numerous scholars focusing on max-product operation [24, 35]. Loetamonphong and Fang defined two sub-problems by separating negative from non-negative coefficients in the objective function, and then obtained an optimal solution by combining the two sub-problems [35]. The maximum solution of FRE is the optimum for the sub-problem having negative coefficients. Another sub-problem was converted into a binary programming problem and was solved using branch and bound method. Moreover, generalizations of the linear optimization problem with respect to FRE have been studied; this was done through replacement of max-min and max-product compositions with different fuzzy compositions [28, 56, 62].

Recently, many interesting generalizations of the linear programming problems constrained by FRE or FRI have been introduced and developed based on composite operations and fuzzy relations used in the definition of the constraints, and some developments on the objective function of the problems [5, 9, 13, 16, 17, 22, 29, 34, 65]. For instance, the linear optimization of bipolar FRE was studied by some researchers where FRE was defined with max-min composition [13] and max-Lukasiewicz composition [29, 34]. In [29] the authors introduced the optimization problem subjected to a system of FRE defined as $X(A^+, A^-, b) = \{x \in [0, 1]^m : x \circ A^+ \vee \tilde{x} \circ A^- = b\}$ where $\tilde{x}_i = 1 - x_i$ for each component of $\tilde{x} = (\tilde{x}_i)_{1 \times m}$ and the notations “ $\vee$ ” and “ $\circ$ ” denote max operation and the max-Lukasiewicz composition, respectively. They translated the problem into a 0-1 integer linear programming problem which is then solved using well-developed techniques. In [34], the foregoing problem was solved by an analytical method based on the resolution of the feasible region. In [18], the authors introduced a linear optimization problem with constraints formed by $X(A, D, b^1, b^2) = \{x \in [0, 1]^n : A\varphi x \leq b^1, D\varphi x \geq b^2\}$ where φ represents an operator with (closed) convex solutions. This problem was further investigated with φ as the Dubois-Prade t-norm [14] and φ as a continuous increasing operator [41]. Yang et al. [72] studied the optimal solution of minimizing a linear objective function with addition-min FRI constraints defined as $\sum_{j=1}^n \min\{a_{ij}, x_j\} \geq b_i$ for $i = 1, \dots, m$. Ghodousian et al. [17, 20] introduced linear and nonlinear optimization problems with FRI fuzzy constraints (FRI-FC). The feasible region of the FRI-FC problems are defined as $X(A, b) = \{x \in [0, 1]^n : A\varphi x \preceq b\}$, where φ is an arbitrary continuous t-norm and “ $\preceq$ ” denotes the relaxed or fuzzy version of the ordinary inequality “ $\leq$ ”.

The optimization problems with general nonlinear objective functions and FRE or FRI constraints were studied in [5, 9, 15, 19, 21, 30, 36, 59, 60, 65, 69, 72]. For example, Wu et al. presented an efficient method to optimize a linear fractional programming problem under FRE with max-Archimedean t-norm composition [67]. Dempe and Ruziyeva [5] generalized the fuzzy linear optimization problem by considering fuzzy coefficients. Dubey et al. studied linear programming problems involving interval uncertainty modeled using intuitionistic fuzzy set [9]. Yang et al. [70] studied the single-variable term semi-latticeized geometric programming subject to max-product fuzzy relation

equations. The proposed problem was devised from the peer-to-peer network system and the target was to minimize the biggest dissatisfaction degrees of the terminals in such system. Latticed optimization problem was introduced in [?, 61] here the constraints and the objective function were defined as max-min FRI and $z(x) = \max_{i=1}^n \{ \min \{ c_i, x_i \} \}$ with $c_i \in [0, 1]$, respectively. The latticed programming problem subjected to max-min FRI was also investigated in the works [30, 69]. In [30], based on the concept of semi-tensor product, a matrix approach was applied to handle the latticed linear programming problem subjected to max-min FRI. Also, Yang et al. [71] introduced another version of the latticed programming problem defined by minimizing objective function $z(x) = \max_{j=1}^n \{ x_j \}$ subject to $X(A, b) = \{ x \in [0, 1]^n : A \circ x \geq b \}$ where “ $\circ$ ” denotes fuzzy max-product composition. On the other hand, Lu and Fang considered the single non-linear objective function and solved it with FRE constraints and max-min operator [38]. They proposed a genetic algorithm for solving the problem. In [15], the authors improved Lu and Fang’s algorithm by introducing different crossover and mutation operators.

Generally, there are three important difficulties related to FRE or FRI problems. Firstly, in order to completely determine FREs and FRIs, we must initially find all the minimal solutions, which is an NP-hard problem. Secondly, a feasible region formed as FRE or FRI is often a non-convex set. Thirdly, FREs and FRIs as feasible regions lead to optimization problems with highly non-linear constraints. Due to the above mentioned difficulties, although the analytical methods are efficient to find exact optimal solutions, they may also involve high computational complexity for high-dimensional problems (especially, if the simplification processes cannot considerably reduce the problem). Furthermore, in spite of the fact that deterministic approaches such as feasible direction and generalized gradient descent methods are fast and able to quickly find a local minimum, they make strong assumptions on the continuity and differentiability of the constraints as well as the objective function [1]. On the other hand, heuristic algorithms, like Ant Colony Optimization (ACO) [8, 57], do not make such assumptions and they have been successfully applied for tackling constrained optimization problems [15, 19, 20, 31, 38, 52]. Particularly, we refer the readers to [10, 26, 33, 39, 48, 68] in which the powerful performance of ant colony optimization algorithm has been shown in applied optimization problems. In this paper, an ACO-based algorithm (FRE-ACO) is proposed for solving the nonlinear optimization problems constrained with max-min fuzzy relational equations. The FRE-ACO algorithm consists of two general phases that are sequentially employed until a termination criterion is satisfied. Phase (I) is based on (discrete) ACO and plays a main role in exploring the new areas of the search space. At each iteration, the purpose of Phase (I) is to find some minimal candidate solution(s) for the max-min FRE, that is a combinatorial problem. This is done by selecting FRE-paths that are attained through use of the structural properties of the FREs, and according to their corresponding pheromone values. Given a minimal candidate solution (generated by a FRE-path), the algorithm can find a subset of the feasible region that is analytically a closed convex cell. In fact, the algorithm associates to each FRE-path exactly one closed convex cell. The second phase minimizes the original objective function starting with the feasible solutions that are randomly selected from the feasible cells obtained at the end of Phase (I). Phase (II) is based on ACO_R and guarantees the exploitation of the algorithm. At the end of Phase (II), the solution archive is used to update the pheromone values of the previously found FRE-paths. Better solutions increase the probability of choosing their corresponding cells at the next iteration, by increasing the pheromone on those FRE-paths associated with their cells. So, at the next iteration, Phase (I) is biased towards FRE-paths that are associated with the most promising cells (i.e., those cells including better solutions). In conclusion, Phase (I) provides feasible regions for phase (II) to exploit, and Phase (II) in turn updates the pheromone values of the FRE-paths used by phase (I) for finding better feasible regions. As it will be shown, the proposed algorithm keeps the search inside of the feasible region without checking the feasibility of new generated solutions.

Unlike general-purpose ACO algorithms, where pheromone values on a path are updated after path construction, in the FRE-ACO algorithm, FRE-paths are not evaluated during Phase I (path construction). Instead, the quality of an FRE-path is determined in Phase II, based on the quality of a continuous solution generated within the cell formed by that path. Moreover, in the original ACOR, the best archive solutions have the highest probability of being selected during sampling. However, in FRE-ACO, the best solutions increase the likelihood of selecting the FRE-paths that generate the corresponding cells. On the other hand, in [15, 38], the selection strategy is solely based on the rank of the solutions in the population. In contrast, the FRE-ACO algorithm’s strategy depends not only on the rank of the archive solutions but also on the probability of selecting their corresponding FRE-paths, as well as the quality of the random solution added to the archive at each iteration. Furthermore, in [38], the proposed mutation operator decreases a variable to a random number between a range (later improved in [15] by setting the variable to zero). In the proposed FRE-ACO algorithm, newly generated points are always selected from the closed convex cells that are subsets of the feasible region. Additionally, the crossover operator in [38] generates four new solutions, but only one is guaranteed to be feasible, while the others require feasibility checks against the problem’s constraints. In [15], a crossover operator using a single parent produces two feasible offspring, one of which remains close to the parent. In the FRE-ACO algorithm, a new feasible solution is generated by sampling around an archive solution, ensuring both

solutions lie within the same feasible cell. Finally, in [15, 38], the initial population is generated by selecting random solutions from a small region, which limits diversity. In contrast, the FRE-ACO algorithm uses cells that include the lower bounds of feasible solutions, increasing its ability to explore a broader search space and discover new solutions. The current paper consists of three main parts. Firstly, we review some structural details of FREs defined by the minimum t-norm such as the theoretical properties of the feasible solution set and some necessary and sufficient conditions for the feasibility of the problem. Then, two preceding phases are described in detail and followed by explanation of FRE-ACO. Finally, some statistical and experimental results are provided to evaluate the performance of our algorithm. Moreover, a comparison is made between the FRE-ACO algorithm and the genetic algorithms presented in [15] and [38] for solving nonlinear optimization problems with respect to max-min FREs. The remainder of the paper is organized as follows. Section 2 takes a brief look at some basic results on the feasible solutions set of problem(1). These results are used throughout the paper and provide a proper background to design an efficient algorithm for solving the problem. The details of Phase (I) and Phase (II) are studied in Sections 3 and 4, respectively. In section 5, the FRE-ACO algorithm and its characteristics are described and a comparative study is presented. Finally, in section 7, the experimental results are demonstrated.

2 A review of the structure of max-min FRE

This section briefly describes some relevant results about the solution to system of max-min fuzzy relational equations. Let $S(A, b)$ denotes the feasible solution set of problem(1), that is, $S(A, b) = \{x \in [0, 1]^n : A \varphi x = b\}$. For $x, y \in [0, 1]^n$, we say $x \leq y$ if and only if $x_j \leq y_j, \forall j \in J$. Moreover, a solution $\bar{x} \in S(A, b)$ is said to be a maximum solution, if $x \leq \bar{x}, \forall x \in S(A, b)$. Similarly, $\underline{x} \in S(A, b)$ is called a minimal solution, if $x \leq \underline{x}$ implies $x = \underline{x}, \forall x \in S(A, b)$. Also, the notation $\underline{S}(A, b)$ represents the set of all the minimal solutions of $S(A, b)$. Furthermore, we define $I(j) = \{i \in I : a_{ij} > b_i\}, \forall i \in I$. According to [26], if $S(A, b) \neq \emptyset$, maximum solution $\bar{x}$ is easily obtained as stated in the following lemma.

Lemma 2.1. *Let $I(j) = \{i \in I : a_{ij} > b_i\}, \forall i \in I$. If $S(A, b) \neq \emptyset$, maximum solution $\bar{x}$ is obtained as follows:*

$$\bar{x}_j = \begin{cases} \min_{i \in I(j)} \{b_i\} & I(j) \neq \emptyset \\ 1 & I(j) = \emptyset \end{cases}, \forall j \in J \quad (2)$$

Definition 2.2. *Let $\bar{x}$ be the maximum solution of $S(A, b)$. For each $i \in I$, we define $\bar{J}(i) = \{j \in J : \min\{a_{ij}, \bar{x}_j\} = b_i\}$.*

Definition 2.3. *Suppose that $S(A, b) \neq \emptyset$. Let $e : I \rightarrow \bigcup_{i \in I} \bar{J}(i)$ be a function defined on I such that $e(i) \in \bar{J}(i), \forall i \in I$. Also, let E denote the set of all the functions e .*

Remark 2.4. *From Definition 2.3, each $e \in E$ can be equivalently considered as a set of ordered pairs*

$$\{(1, j_1), (2, j_2), \dots, (m, j_m)\},$$

in which $j_i = e(i), \forall i \in I$. So, for the sake of simplicity, this set is usually represented only in terms of the second entries $j_1, j_2, \dots, j_m$, and denoted by the notation $e = [j_1, j_2, \dots, j_m]$ as a finite sequence.

According to Definition 2.3, it is clear that $|E| = \prod_{i \in I} |\bar{J}(i)|$, where $|E|$ and $|\bar{J}(i)|$ ($i = 1, \dots, m$) denote the cardinality of sets E and $\bar{J}(i)$ ($i = 1, \dots, m$), respectively.

Definition 2.5. *Suppose that $S(A, b) \neq \emptyset$ and $e \in E$. Let $I_e(j) = \{i \in I : e(i) = j\}, \forall j \in J$. Let $\underline{x}(e) \in [0, 1]^n$ whose components are defined as follows:*

$$\underline{x}(e)_j = \begin{cases} \max_{i \in I_e(j)} \{b_i\} & I_e(j) \neq \emptyset \\ 0 & I_e(j) = \emptyset \end{cases}, \forall j \in J \quad (3)$$

Based on the following theorem, each $\underline{x}(e)$ ($e \in E$) is feasible to $S(A, b)$. Moreover, although a solution $\underline{x}(e)$ ($e \in E$) may not be minimal, each minimal solution $\underline{x} \in \underline{S}(A, b)$ can be written in the form of $\underline{x}(e)$ for some $e \in E$ [18, 26]. For this reason, each $\underline{x}(e)$ ($e \in E$) is generally referred to as a **minimal candidate solution**.

Theorem 2.6. Let $S_E(A, b) = \{ \underline{x}(e) : e \in E \}$. Then, $\underline{S}(A, b) \subseteq S_E(A, b) \subseteq S(A, b)$.

Proof. See proof of Lemma 9 in [26]. □

Figure 1 depicts a typical feasible region $S(A, b)$ for problem (1) (shadowed region) in the 3-dimensional space $S = [0, 1]^3$. As mentioned before, $S(A, b)$ is often a non-convex set. Figure 1 also shows the different possible cases that may arise for minimal candidate solutions in the feasible region. As shown in the figure, $\bar{x}$ is the maximum solution of $S(A, b)$, and three minimal candidate solutions $\underline{x}(e_1)$, $\underline{x}(e_2)$ and $\underline{x}(e_3)$ are both feasible and minimal. However, although the minimal candidate solution $\underline{x}(e_4)$ is feasible to $S(A, b)$, it is not a minimal solution, i.e., $\underline{x}(e_4) \notin \underline{S}(A, b)$.

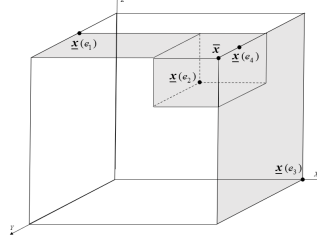


Figure 1: The feasible region of problem(1)

By the following theorem [7, 18, 26], the feasible region of problem(1) is completely determined by the unique maximum solution $\bar{x}$ and a finite number of minimal candidate solutions $\underline{x}(e)$ ($e \in E$). This theorem also shows the important role of the minimal candidate solutions in characterizing the feasible region of problem(1). Indeed, solutions $\underline{x}(e)$ ($e \in E$) are the lower bounds of the feasible solution set for problem(1).

Theorem 2.7. If $S(A, b) \neq \emptyset$, then $S(A, b) = \bigcup_{e \in E} [\underline{x}(e), \bar{x}]$, where $[\underline{x}(e), \bar{x}]$ is an n -dimensional interval including all points between $\underline{x}(e)$ and $\bar{x}$.

Proof. See Theorem 3.1 in [18]. □

Definition 2.8. [55] If $l_j, u_j \in \mathbb{R}$ and $l_j < u_j$ for $j = 1, \dots, k$, the set of all points $x = [x_1, \dots, x_k]$ in $\mathbb{R}^k$ whose coordinates satisfy the inequalities $l_j \leq x_j \leq u_j$ ($1 \leq j \leq k$) is called a k -cell. Thus, a 1-cell is an interval such as $[a, b]$, a 2-cell is a rectangle, etc. Therefore, each k -cell is a closed convex set.

By Theorem 2.7, $S(A, b)$ is the set of all points between one maximum solution $\bar{x}$, and a finite number of minimal candidate solutions $\underline{x}(e)$ (see Figure 1). In other words, $S(A, b)$ is a union of a finite number of closed convex cells $[\underline{x}(e), \bar{x}]$ ($e \in E$). In Figure 1, $S(A, b) = \bigcup_{i=1}^4 [\underline{x}(e_i), \bar{x}]$ where both $[\underline{x}(e_1), \bar{x}]$ and $[\underline{x}(e_3), \bar{x}]$ are 2-cells, $[\underline{x}(e_2), \bar{x}]$ is a 3-cell and $[\underline{x}(e_4), \bar{x}]$ is a 1-cell. As a consequence of Theorem 2.7, it turns out that $S(A, b) \neq \emptyset$ iff $\bar{x} \in S(A, b)$ [7, 18, 26]. This corollary gives a simple necessary and sufficient condition for determining the feasibility of problem(1).

Definition 2.9. A matrix $M = (m_{ij})_{m \times n}$ is said to be the minimal candidate matrix of set $S(A, b)$, if

$$m_{ij} = \begin{cases} b_i & j \in \bar{J}(i) \\ 0 & j \notin \bar{J}(i) \end{cases}, \forall i \in I, \quad (4)$$

Using the minimal candidate matrix, another feasibility condition can be obtained as follows. $S(A, b) \neq \emptyset$ iff for each $i \in I$, there exists at least some $j \in J$ such that $m_{ij} > 0$ [18].

Based on Remark 2.4 and Definitions 2.3 and 2.9, each $e = [j_1, j_2, \dots, j_m]$ in E can be interpreted as a path, denoted by $(1, j_1) \rightarrow (2, j_2) \rightarrow \dots \rightarrow (m, j_m)$, in the minimal candidate matrix M (as before, $j_i = e(i)$, $\forall i \in I$). In this respect, e (as a path in M) selects one and only one element (i, j_i) from each row i of matrix M . Also, it is important to note that each element or node (i, j_i) included in e corresponds to a positive entry m_{ij_i} in M . Hereafter, each $e \in E$ is referred to as a **FRE-path**, that is, a path in the minimal candidate matrix. As a result, from Definition 2.5, each

FRE-path $e \in E$ generates a minimal candidate solution $\underline{x}(e) \in S_E(A, b)$. So, each FRE-path e can be associated to a closed convex cell $[\underline{x}(e), \bar{x}]$ that is a subset of the feasible region $S(A, b)$ from Theorem 2.7. The following example illustrates the above-mentioned definitions.

Example 2.10. Consider the following FRE with the minimum t -norm φ

$$\begin{bmatrix} 0.7 & 0.3 & 0.8 & 0.4 & 0.8 & 0.7 \\ 0.5 & 0.9 & 0.5 & 0.4 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.5 & 0.3 & 0.0 & 0.3 \\ 0 & 0.1 & 0 & 0.6 & 0.1 & 0 \\ 0.6 & 0.5 & 0.2 & 0.5 & 0.5 & 0.6 \end{bmatrix} \varphi x = \begin{bmatrix} 0.7 \\ 0.5 \\ 0.3 \\ 0.1 \\ 0.6 \end{bmatrix}.$$

From Lemma 2.1, we have $I(1) = \emptyset$, $I(2) = \{2\}$, $I(3) = \{1, 3\}$, $I(4) = \{4\}$, $I(5) = \{1\}$ and $I(6) = \emptyset$. Hence, maximum solution is obtained as $\bar{x} = [1, 0.5, 0.3, 0.1, 0.7, 1]$. It is easy to verify that $\bar{x} \in S(A, b)$ which implies $S(A, b) \neq \emptyset$. Also, from Definition 2.2, $\bar{J}(1) = \{1, 5, 6\}$, $\bar{J}(2) = \{1, 2\}$, $\bar{J}(3) = \{3, 6\}$, $\bar{J}(4) = \{2, 4, 5\}$ and $\bar{J}(5) = \{1, 6\}$. So, $|E| = |\bar{J}(1)| \times \dots \times |\bar{J}(5)| = 72$. Therefore, the number of all FRE-paths is equal to 72. Each FRE-path $e \in E$ generates a minimal candidate solution $\underline{x}(e)$. As an example, by selecting $e' = [5, 1, 6, 5, 1]$, from Definition 2.5 the corresponding solution is obtained as $\underline{x}(e') = [0.6, 0, 0, 0, 0.7, 0.3]$. From Theorem 2.7, we know that $[\underline{x}(e'), \bar{x}] \subseteq S(A, b)$, i.e., all points between $\underline{x}(e')$ and $\bar{x}$ are feasible to problem(1). Moreover, the minimal candidate matrix $M = (m_{ij})_{5 \times 6}$ is obtained from Definition 2.9 as follows:

$$M = \begin{bmatrix} 0.7 & 0 & 0 & 0 & \boxed{0.7} & 0.7 \\ \boxed{0.5} & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & \boxed{0.3} \\ 0 & 0.1 & 0 & 0.1 & \boxed{0.1} & 0 \\ \boxed{0.6} & 0 & 0 & 0 & 0 & 0.6 \end{bmatrix}$$

As mentioned before, e' can be also denoted by $(1, 5) \rightarrow (2, 1) \rightarrow (3, 6) \rightarrow (4, 5) \rightarrow (5, 1)$. This FRE-path has been shown in matrix M . As is shown in the above matrix, each node included in FRE-path e' (marked components) corresponds to a positive entry of M . Also, FRE-path e' selects exactly one positive component from each row of M .

3 Phase (I)

The original (discrete) ant colony optimization (ACO) algorithm [8] was proposed for solving combinatorial optimization problems such as scheduling, vehicle routing, timetabling, travelling salesman problem and so on. Many of these problems, especially those of practical relevance, are NP-hard. Generally, a model of a combinatorial optimization problem may be defined as follows [59]:

Definition 3.1. A model $P = (S, \Omega, f)$ of a combinatorial optimization problem consists of a search space S defined over a finite set of discrete decision variables X_i ($i = 1, \dots, m$); a set Ω of constraints among the variables; and an objective function $f : S \rightarrow [0, \infty)$ to be minimized. Furthermore, the generic variable X_i takes values in $D_i = \{v_i^1, \dots, v_i^{|D_i|}\}$.

As mentioned in Section 1, FRE-ACO algorithm consists of two main phases. The purpose of Phase (I) is to find convex subset(s) of the feasible region, while Phase (II) produces better solutions by starting from feasible solutions selected from the convex feasible subsets. Each of the two phases is employed once during each iteration. In FRE-ACO, the pheromone is deposited on the positive entries of the minimal candidate matrix M . So, pheromone value τ_{ij} (associated with entry m_{ij}) is positive if $j \in \bar{J}(i)$ and $\tau_{ij} = 0$ otherwise. These values are summarized in the pheromone matrix $\tau = (\tau_{ij})_{m \times n}$. Pheromone matrix τ is updated at the end of each iteration, i.e., at the end of Phase (II). Moreover, based on a pheromone matrix τ in any arbitrary iteration, a probability matrix $P = (p_{ij})_{m \times n}$ is obtained as follows

$$p_{ij} = \frac{\tau_{ij}}{\sum_{k=1}^n \tau_{ik}}, \quad (5)$$

for each $i \in I$ and $j \in J$. According to Equation(4), p_{ij} is the probability of choosing (i, j) 'th entry of M (i.e., m_{ij}) from all entries in row i . In other words, for a fixed row i , value p_{ij} represents the probability of choosing column j from $J = \{1, \dots, n\}$. At the beginning, the values τ_{ij} are set as follows:

$$\tau_{ij} = \begin{cases} 1 & j \in \bar{J}(i) \\ 0 & j \notin \bar{J}(i) \end{cases} \quad (6)$$

Hence, from (4) we obtain

$$p_{ij} = \begin{cases} 1 / |\bar{J}(i)| & j \in \bar{J}(i) \\ 0 & j \notin \bar{J}(i) \end{cases} \quad (7)$$

Given a probability matrix P , Phase (I) starts with m_1 ants. In this phase, each FRE-path is interpreted as a solution, and each artificial ant incrementally builds a partial FRE-path by selecting exactly one $j_i \in \bar{J}(i)$ for each row i in M . So, at the end of $(i-1)$ 'th step of the solution construction, a partial FRE-path $[j_1, \dots, j_{i-1}]$ is obtained such that $j_s \in \bar{J}(s)$ for $s = 1, \dots, i-1$. At i 'th step, ant k visits i 'th row of M and selects column $j_i \in \bar{J}(i)$ (equivalently, entry m_{ij_i} in row i of matrix M) with probability p_{ij} as defined in (4). Each ant moves m steps to build an FRE-path $e = [j_1, \dots, j_m]$. At the end of Phase (I), with m_1 FRE-paths $e_1, \dots, e_{m_1}$ at hand we proceed to obtain the corresponding minimal candidate solutions $\underline{x}(e_1), \dots, \underline{x}(e_{m_1})$ by Definition 2.5. Now, Theorem 2.7 implies the feasibility of m_1 convex cells $[\underline{x}(e_1), \bar{x}], \dots, [\underline{x}(e_{m_1}), \bar{x}]$. So, in order to access the feasible cells $[\underline{x}(e_l), \bar{x}]$, $l = 1, \dots, m_1$, it is sufficient to know their lower bounds (i.e., $\underline{x}(e_l)$, $l = 1, \dots, m_1$); because, the upper bound of each feasible cell is the maximum solution $\bar{x}$ that can be always found by (2) at the beginning. The preceding discussions can be summarized as an algorithm shown below:

Algorithm 1: (Phase(I)).

- 1 Get m_1 and pheromone matrix τ .
 - 2 Generate the probability matrix P by (4).
 - 3 For $l = 1$ to m_1
 - 4 For $i = 1$ to m
 - 5 select j_i with the probability p_{ij} stated by (4).
 - 6 set $e_l(i) = j_i$.
 - 7 End For
 - 8 Set $e_l = [j_1, \dots, j_m]$.
 - 9 By (3), compute minimal candidate solution $\underline{x}(e_l)$ generated by the FRE-path e_l .
 - 10 End For
-

It is to be noted that by setting $S = E$, $X_i = e(i)$ and $D_i = \bar{J}(i)$, the problem of finding FRE-paths $e \in E$ can be represented as a combinatorial problem $P = (S, \Omega, f)$ as defined in Definition 3.1. The only difference is that Phase (I) considers no objective function to be optimized, because its purpose is to find some feasible subsets in the feasible region of problem (1).

4 Phase (II)

In [59], the authors proposed a way to effectively apply ACO to continuous optimization problems following the fundamental framework of original ACO. The fundamental idea underlying ACO_R is substituting the discrete probability distributions used in ACO algorithms for combinatorial problems with probability density functions in the solution construction step. This section describes Phase (II) of FRE-ACO that is based on ACO_R algorithm.

4.1 Initialization

Many optimization methods randomly generate the initial population. This strategy works well when dealing with unconstrained optimization problems. However, for a constrained optimization problem, randomly generated solutions may not be feasible. In FRE-ACO, the initial population (with population size S_{pop}) is given by randomly generating the solutions inside m_1 feasible cells $[\underline{x}(e_l), \bar{x}]$ (one solution per cell). These cells are easily accessible by considering the lower bounds $\underline{x}(e_l)$ obtained at the end of Phase (I) and the maximum solution $\bar{x}$. Each solution s_l in Phase (II) consists of three parts and it may be denoted by $s_l = [s_l(x), s_l(lb), s_l(e)]$; $s_l(x)$ is an n -dimensional point randomly selected from $[\underline{x}(e_l), \bar{x}]$, i.e., $s_l(x) \in [\underline{x}(e_l), \bar{x}]$, $s_l(lb) = \underline{x}(e_l)$ that shows the lower bound of the cell including the feasible point $s_l(x)$, and $s_l(e) = e_l$ that enables us to access the FRE-path by which the cell $[\underline{x}(e_l), \bar{x}]$ is generated. In the following, a typical feasible region $S(A, b)$ has been depicted in the 2-dimensional space $S = [0, 1]^2$. Figure 2 also shows cell $[\underline{x}(e_l), \bar{x}]$ (shadowed region) and a new randomly generated point $s_l(x)$ in $[\underline{x}(e_l), \bar{x}]$.

The algorithm for generating the initial solutions is simply obtained as follows:
For future reference, we give an example to illustrate Algorithm 2.

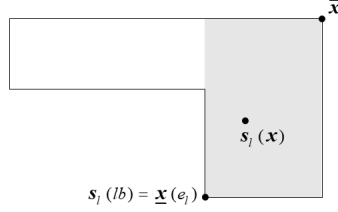


Figure 2: Initialization step.

Algorithm 2: (Initialization).

-
- 1 Get maximum solution $\bar{x}$, m_1 lower bounds $\underline{x}(e_1), \dots, \underline{x}(e_{m_1})$ and objective function $f(x)$.
 - 2 For $l = 1$ to m_1
 - 3 Select a random n -dimensional solution point $s_l(x)$ from $[\underline{x}(e_l), \bar{x}]$.
 - 4 Set $s_l(lb) = \underline{x}(e_l)$ and $s_l(e) = e_l$.
 - 5 Evaluate and add the solution $s_l = [s_l(x), s_l(lb), s_l(e)]$ to the archive T .
 - 6 End For
 - 7 Sort solutions s_l , $l = 1, \dots, m_1$, in an ascending order of objective values $f(s_l(x))$.
-

Example 4.1. Consider the maximum solution $\bar{x} = [1, 0.5, 0.3, 0.1, 0.7, 1]$, the FRE-path $e' = [5, 1, 6, 5, 1]$ and its corresponding minimal candidate solution $\underline{x}(e') = [0.6, 0, 0, 0, 0.7, 0.3]$ stated in Example 2.10. Suppose that e' has been already generated by some ant $l \in \{1, \dots, m_1\}$ in Phase (I). Therefore, we have $e_l = e'$ and $\underline{x}(e_l) = \underline{x}(e')$. Consequently, in order to generate a solution $s_l = [s_l(x), s_l(lb), s_l(e)]$, Algorithm 2 chooses a random point from the cell $[\underline{x}(e_l), \bar{x}]$, e.g., $s_l(x) = [0.8, 0.3, 0.2, 0, 0.7, 1]$. Moreover, by assuming that $f(x) = x_1x_4 - x_2x_3x_5 + x_6^2$ is the objective function of problem (1), solution s_l is evaluated by $f(s_l(x)) = 0.958$.

4.2 Sampling process

In Phase (II), the algorithm stores a set of S_{pop} so-far-best solutions in a solution archive T . The solution archive is used to create a probability distribution of promising solutions over the continuous search space. Initially, T is filled with solutions $s_l = [s_l(x), s_l(lb), s_l(e)]$, $l = 1, \dots, S_{pop}$, that are randomly generated by Algorithm 2. The j 'th variable of the l 'th solution point $s_l(x)$ is denoted by $s_l(x)_j$. Also, the quality of each solution s_l is evaluated by $f(s_l(x))$, that is, the objective function value of its corresponding feasible point. Phase (II) iteratively refines the solution archive by generating m_2 new solutions and then keeping only the best S_{pop} solutions of the $S_{pop} + m_2$ solutions that are available. The S_{pop} solutions s_l in T are always sorted according to their quality from the best to worst, that is, solution s_l has rank l . Therefore, the first solution, s_1 , is the best one in the sense that it includes the best solution point $s_1(x)$ (stored in its first part) with the smallest objective value $f(s_1(x))$. The weight of the archive solution s_l is calculated by the following Gaussian function:

$$\omega_l = \frac{1}{\sqrt{2\pi} q S_{pop}} e^{-\frac{1}{2} \left(\frac{l-1}{q S_{pop}} \right)^2}, \quad l = 1, \dots, S_{pop}, \quad (8)$$

which essentially defines the weight to be a value of the Gaussian function associated with the ranking of solution l in the archive, mean 1, and standard deviation $q S_{pop}$, where q is a parameter of the algorithm. When q is small, the best-ranked solutions are strongly preferred, and when it is large, the probability becomes more uniform.

During the solution generation process, each coordinate is treated independently. The generation of new solutions is accomplished by a sampling process as follows. First, an archive solution s_l is chosen with the following probability:

$$p_l = \frac{\omega_l}{\sum_{k=1}^{S_{pop}} \omega_k}. \quad (9)$$

Then, for $j \in J = \{1, \dots, n\}$, the algorithm samples around the j 'th component of the solution point of the selected solution, i.e., s_l^j , using a Gaussian function g_l^j , where l is a solution index and j is a coordinate index. The Gaussian function for coordinate j is given by:

$$g_l^j(x) = \frac{1}{\sigma_l^j \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x - \mu_l^j}{\sigma_l^j} \right)^2}, \quad (10)$$

with the mean $\mu_l^j = s_l(x)_j$, and the standard deviation σ_l^j equal to

$$\sigma_l^j = \xi \sum_{k=1}^{S_{pop}} \frac{|s_k(x)_j - s_l(x)_j|}{S_{pop} - 1}, \quad l = 1, \dots, S_{pop}, \quad (11)$$

which is the average distance between the j 'th variable of the feasible point $s_l(x)$ and the j 'th variable of the other feasible points in the archive. The parameter $\xi > 0$ has an effect similar to that of the pheromone evaporation rate in ACO. The higher the value of ξ , the lower the convergence speed of the algorithm. By repeating (10) for $j \in \{1, \dots, n\}$, we obtain a solution x' around $s_l(x)$. Since x' may be infeasible, we adjust its components to guarantee that both x' and $s_l(x)$ will be in the same cell, i.e., $[\underline{x}(e_l), \bar{x}]$. To this end, the algorithm uses the maximum solution $\bar{x}$ and the lower bound $\underline{x}(e_l)$ stored in the second part of $s_l(x)$, i.e., $s_l(lb) = \underline{x}(e_l)$. So, this is accomplished by the following assignment:

$$x'_j \leftarrow \min \{ \max \{ x'_j, \underline{x}(e_l)_j \}, \bar{x}_j \}, \quad j = 1, \dots, n, \quad (12)$$

Finally, since each solution archive contains three parts, we define the new solution s' by $s' = [s'(x), s'(lb), s'(e)]$ where $s'(x) = x'$, $s'(lb) = s_l(lb) = \underline{x}(e_l)$ and $s'(e) = s_l(e) = e_l$. The solution generation process is repeated m_2 times and the new solutions $s'_1, \dots, s'_{m_2}$ are added to the archive T . Each new solution that is added to T is evaluated and ranked. The sampling process is presented as the following algorithm:

Algorithm 3: (Sampling process).

- 1 Get $q, \xi, m_2, f(x)$ and sorted solutions $s_l = [s_l(x), s_l(lb), s_l(e)]$, $l = 1, \dots, S_{pop}$.
 - 2 For $k = 1$ to m_2
 - 3 According to (9), select an archive solution s_l with probability p_l .
 - 4 According to (10) and (11), sample Gaussian function $g_l^j(x)$ for $j = 1, \dots, n$.
 - 5 Generate the modified point x' by assignment (12).
 - 6 Generate new solution $s'_k = [s'_k(x), s'_k(lb), s'_k(e)]$ by setting $s'_k(x) = x'$, $s'_k(lb) = \underline{x}(e_l)$
 - 7 and $s'_k(e) = e_l$.
 - 8 End For
 - 9 Evaluate and add the new solutions s'_k , $k = 1, \dots, m_2$, to the archive T .
 - 10 Update the archive by keeping the S_{pop} best solutions and removing the m_2 worst ones.
-

4.3 Updating pheromone matrix

At the end of Phase (II), the solution archive is also used to update pheromone matrix τ . For this purpose, each archive solution $s_l = [s_l(x), s_l(lb), s_l(e)]$ updates the matrix τ as follows:

$$\tau_{i, e_l(i)} = \tau_{i, e_l(i)} + Q e^{-f(s_l(x))}, \quad \forall i \in I, \quad (13)$$

where $e_l = s_l(e)$ (stored in the third part of s_l), $\tau_{i, e_l(i)}$ denotes the entry in the i 'th row and $e_l(i)$ 'th column of the matrix τ , $f(s_l(x))$ is the objective value of the feasible point $s_l(x)$ (stored in the first part of s_l), Q is a positive adjusting constant and $Q e^{-f(s_l(x))}$ is the quality of pheromone laid on each entry $(i, e_l(i))$ of τ , $\forall i \in I$. The pheromone matrix τ is updated S_{pop} times by Equation (13). Finally, it is modified by the evaporation rate $\rho \in [0, 1)$ as follows:

$$\tau = (1 - \rho) \tau. \quad (14)$$

In the archive T , better solutions s_l include better feasible points $s_l(x)$ with less objective values $f(s_l(x))$ that imply higher values of $Q e^{-f(s_l(x))}$. Therefore, better solutions deposit higher amounts of pheromone on the components of matrix τ . Moreover, according to (13), each archive solution s_l increases the probability of choosing its corresponding FRE-path (i.e., $s_l(e) = e_l$) by Phase (I) at the next iteration. This strategy guides Phase (I) towards those FRE-paths that generate promising cells, that is, those cells that includes better solution points.

The algorithm for updating the pheromone matrix is described as follows:

Algorithm 4: (Updating pheromone matrix).

-
- 1 Get ρ , Q , $f(x)$, $s_l = [s_l(x), s_l(lb), s_l(e)]$, $l = 1, \dots, S_{pop}$, and the pheromone matrix τ .
 - 2 For $l = 1$ to S_{pop}
 - 3 Update the pheromone matrix τ by (13).
 - 4 End For
 - 5 Modify the pheromone matrix τ by (14).
-

5 FRE-ACO algorithm and a comparative study

Based on Algorithms 1-4, the different steps of FRE-ACO algorithm are expressed as follows. As is described in Algorithm 5, at the first iteration ($t = 1$), Phase (I) starts with $m_1 = S_{pop}$ ants to generate S_{pop} FRE-paths e_l and their corresponding minimal candidate solutions $\underline{x}(e_l)$ ($l = 1, \dots, S_{pop}$). Then, by using the S_{pop} cells $[\underline{x}(e_l), \bar{x}]$, Algorithm 2 creates the initial archive T with the archive size S_{pop} . At the first iteration, Phase (II) does not perform the sampling process, and therefore the initial archive is directly utilized to update the pheromone matrix τ . In contrast, at the next iterations, Phase (I) uses only one ant ($m_1 = 1$) to find just one FRE-path e_l based on the biased (by pheromone) probabilistic choice of the entries τ_{ij} . Consequently, except the first iteration, Phase (I) produces only one minimal candidate solution $\underline{x}(e_l)$. Subsequently, using the feasible cell $[\underline{x}(e_l), \bar{x}]$, Algorithm 2 generates one new solution and adds the solution to the archive T . This strategy avoids a premature convergence in FRE-ACO Algorithm and increases the ability of the algorithm in escaping from local optima in a promising cell. Moreover, Phase (II) performs the sampling process two times ($m_2 = 2$) at each iteration t such that $t > 1$. To sum up, Phase (I) provides feasible convex cells for Phase (II) to explore in proper regions for generating an initial archive. Additionally, Phase (I) also participates in the exploitation step by constructing the FRE-paths that generates promising cells. On the other hand, Phase (II) improves the solutions by sampling around the better solution points in the archive, and by increasing the probability of choosing the FRE-paths that generate the promising cells. Meanwhile, Phase (II) also participates in the exploration step by finding a random solution point in a promising cell, and by adding this solution point to the archive.

Algorithm 5: (FRE-ACO algorithm).

-
- 1 Get fuzzy matrix A , fuzzy vector b , objective function $f(x)$, population size S_{pop} , constants q , ξ , ρ and Q , and the maximum number of iterations t_{max} .
 - 2 Compute the maximum solution $\bar{x}$ by (2).
 - 3 If $\bar{x} \notin S(A, b)$, then stop (problem is infeasible).
 - 4 Generate the initial pheromone matrix τ by (5).
 - 5 **Phase (I):**
 - 6 Employ Algorithm 1 with $m_1 = S_{pop}$ to generate S_{pop} FRE-paths.
 - 7 **Phase (II):**
 - 8 Create the initial archive T by Algorithm 2.
 - 9 Update the pheromone matrix τ by Algorithm 4.
 - 10 For $t = 2$ to t_{max}
 - 11 **Phase (I):**
 - 12 Employ Algorithm 1 with $m_1 = 1$ to generate a promising FRE-path.
 - 13 **Phase (II):**
 - 14 Employ Algorithm 2 to generate a new solution and add the solution to the archive.
 - 15 Employ the sampling process with $m_2 = 2$ by Algorithm 3.
 - 16 Update the pheromone matrix τ by employing Algorithm 4.
 - 17 End For
-

The following points highlight some of the main characteristics of the proposed algorithm and some important differences between FRE-ACO and the original ACO and ACO_R algorithms.

- FRE-ACO algorithm consists of two phases that are based on discrete and continuous ant colony optimization algorithms. Employing these two ideas in the same algorithm provides a suitable tradeoff between exploration and exploitation.

- Unlike the general-purpose original ACO algorithms, FRE-ACO algorithm has been designed specifically for solving nonlinear optimization problems with fuzzy relation constraints. For this purpose, the current algorithm solves the problem by taking advantage of the structure of the solutions set that is a non-convex set.
- In FRE-ACO (Phase (I)), the purpose of ants is to find FRE-paths, i.e., the paths in the minimal candidate matrix M .
- In ACO, the pheromone value on a path is updated after the path construction while in FRE-ACO algorithm FRE-paths are not evaluated in Phase (I) where the FRE-paths are built. Indeed, the quality of an FRE-path is evaluated based on the quality of a continuous solution (generated in Phase (II)) in the cell generated by the FRE-path.
- In FRE-ACO algorithm, each archive solution includes three parts; first, a feasible solution point; second, the lower bound of the cell containing the solution point; and third, the FRE-path that generates the lower bound.
- In the original ACO_R , the best archive solutions have the highest chances to be selected in the sampling process. However, in FRE-ACO algorithm, the best solutions increase the probability of choosing the FRE-paths that generate their corresponding cells. More precisely, if $s_1 = [s_1(x), s_1(lb), s_1(e)]$ is the best archive solution at t 'th iteration and its FRE-path (i.e., $s_1(e)$) is built in Phase (I) at $(t + 1)$ 'th iteration, then Algorithm 2 adds a new solution $s_l = [s_l(x), s_l(lb), s_l(e)]$ to the archive such that $s_l(lb) = s_1(lb)$, $s_l(e) = s_1(e)$, and $s_l(x)$ and $s_1(x)$ are in the same cell $[s_1(lb), \bar{x}]$. Only in the case that $f(s_1(x)) \leq f(s_l(x))$, solution s_1 can maintain its rank in the archive with the highest probability to be selected in the sampling process.

Moreover, to see how the current algorithm is situated comparing the other algorithms designed for FRE nonlinear optimization problems, we compare theoretically our algorithm with the genetic algorithms presented in [15, 38]. In addition, an experimental comparison is given in the next section.

In [15, 38], the selection strategy is only based on the rank of the solutions in the population. But in FRE-ACO algorithm, this strategy depends on the rank of the archive solutions as well as the probability of choosing their corresponding FRE-paths and the quality of the random solution that is added to the archive at each iteration $t > 1$. In [38], the proposed mutation operator decreases one variable x_j of point x to a random number between $[0, x_j]$ each time. In this mutation operator, a decreasing variable often followed by increasing several other variables to guarantee the feasibility of a new solution. This operator was improved in [15] by finding a proper variable, and by decreasing it to zero. However, in the proposed algorithm, new generated points are always selected from the closed convex cells that are the subsets of the feasible region.

Lu and Fang [38] used the “three-point” crossover operator defined by three points (two parents x_1 and x_2 , and the maximum solution $\bar{x}$) and two operators called “contraction” and “extraction”. Both contraction and extraction operators are employed between x_1 and x_2 , and between x_i ($i = 1, 2$) and $\bar{x}$. However, from the four mentioned cases, only one case certainly results in a feasible offspring (i.e., the contraction between x_i ($i = 1, 2$) and $\bar{x}$). Therefore, for the other three cases, the feasibility of the new generated solutions must be checked by substituting them into the fuzzy relational equations as well as the constraints $x_j \in [0, 1]$, $\forall j \in J$. In [15], the authors introduced a crossover operator using only one parent each time and for producing two feasible offsprings x_{new1} and x_{new2} . Offspring x_{new1} is obtained as a random point on the line segment between x' (generated by the mutation operator) and $\bar{x}$. But, offspring x_{new2} lies close to its parent. In the current algorithm, the new point $s'(x)$ (the first part of s') is generated by sampling around $s_l(x)$ (the first part of the archive solution s_l) in such a way that both $s'(x)$ and $s_l(x)$ are in the same feasible cell $[\underline{x}(e_l), \bar{x}]$. This cell is easily accessible by considering the lower bound $\underline{x}(e_l)$ that is stored in the second part of s_l , i.e., $s_l(lb) = \underline{x}(e_l)$, and the maximum solution $\bar{x}$ that is obtained easily at the beginning of the algorithm.

In [15, 38], initial population is generated by selecting random solutions from the cell $F = [\underline{x}, \bar{x}]$, where $\underline{x}$ is obtained by the maximization on all the elements of set $S = \{\underline{x}(e) : e \in E, \underline{x}(e) \leq \bar{x}\}$. The preference of this strategy is mainly based on the finding of region F without calculating the minimal solutions. nevertheless, $\underline{x}$ has high sensitivity to each component of each minimal solution. In other words, if any component of any minimal point is large, then the same component of $\underline{x}$ will be large that restricts region F . Region F may be even smaller when S includes some solutions that are not minimal (e.g., point $\underline{x}(e_4)$ in Figure 1). In Figure 1, F is obtained as the line segment between $\underline{x}(e_4)$ and $\bar{x}$. Consequently, F is usually a small region that decreases the diversity of individuals in an initial population. In contrast, the proposed algorithm uses cells $[\underline{x}(e_l), \bar{x}]$ in which the solutions $\underline{x}(e_l)$ are the lower bounds for the feasible solutions (Theorem 2.7). This strategy increases the ability of the algorithm to expand the search space for finding new solutions.

6 Experimental results

In this section, we present the experimental results for evaluating the performance of our algorithm. As mentioned before, we can make a comparison between the FRE-ACO algorithm and the genetic algorithms proposed in [15, 38] for solving nonlinear optimization problems subjected to fuzzy relational equations. For this purpose, we apply these algorithms to 10 test problems described in Appendix. The test problems have been randomly generated in different sizes by the procedure given in [15]. Table 1 shows the optimal objective values for the ten test problems where # denotes the problem number. To perform a fair comparison, we follow the same experimental setup for the parameters $\theta = 0.5$, $\xi = 0.01$, $\lambda = 0.995$ and $\gamma = 1.005$ as suggested in [38], and $q = 0.1$ as suggested by the authors in [15]. Also, we consider $S_{pop} = 50$ for all the three algorithms. The summary of the parameters we used for FRE-ACO is presented in Table 2. Moreover, according to Algorithm 5, the number of ants used by Phase (I) at the t 'th iteration is $m_1 = S_{pop}$ for $t = 1$, and $m_1 = 1$ for $t > 1$. Also, the number of sampling processes employed by Phase (II) at the t 'th iteration is $m_2 = 0$ for $t = 1$, and $m_2 = 2$ for $t > 1$. Finally, 30 experiments are performed for all the algorithms and for each test problem, that is, each of the preceding algorithm is executed 30 times for each test problem. Also, the maximum number of iterations is equal to 100 for all the methods.

Table 1

The optimal values of the test problems.

#	Optimal values
1	- 0.0096019
2	0.8197
3	80.3752
4	- 0.39657
5	- 0.27162
6	1.2612
7	140.4693
8	- 0.10108
9	1.277
10	55.7954

Table 2

Summary of the parameters used by FRE-ACO.

Parameter	Symbol	Value
speed of convergence	ξ	1
Locality of the search process	q	0.0125
Evaporation rate	ρ	0.5
Adjusting constant	Q	1
Archive size	S_{pop}	50

The computational results of the 10 test problems are shown in Table 3, where Avg, Mdn and SD denote the average best-so-far, median best-so-far and standard deviation, respectively. In all the cases, the results that are marked with “*” indicate the better cases. In Table 3, the results have been averaged over 30 runs and the average best-so-far solution and median of the best solution in the last iterations are reported. This table also includes the best results, f_{best} , found by the algorithms over 30 runs. Table 3 demonstrates the ability of the FRE-ACO to detect the optimal solutions of problem (1). According to the results in this table, the best objective values computed by the FRE-ACO and the optimal objective values (see Table 1) match very well. Moreover, Table 3 shows that the current algorithm finds the optimal values faster than those obtained by the other related algorithms and hence has a higher convergence rate, even for the same solutions. The only exceptions are test problems 3 and 4 where all the results are the same for FRE-ACO and the GA presented by Ghodousian et. al. [15]. Also, the good convergence rate of the proposed algorithm could be concluded from Figures 3-12. Furthermore, the proposed algorithm has the smallest standard deviation (good stability) for all the test problems when compared against the other algorithms. Consequently, FRE-ACO could be a robust global search algorithm for nonlinear FRE optimization problems even though the feasible domain is generally non-convex.

Table 3 A comparison between the results found by the GA proposed by Lu and Fang, the GA proposed by Ghodousian and Babalhavaeji and FRE-ACO algorithms for the test problems. The results have been averaged over 30 runs. Maximum number of iterations=100.

#		GA(Lu and Fang)	GA(Ghodousian et. al.)	FRE-ACO
1	Avg	- 0.0075	- 0.0041	- 0.0096*
	Mdn	- 0.0089	- 0.0096*	- 0.0096*
	Sd	0.0236	0.0250	0.0017
	f_{best}	- 0.0096*	- 0.0096*	- 0.0096*
2	Avg	0.8258	0.8206	0.8197*
	Mdn	0.8212	0.8201	0.8197*
	Sd	0.2444	0.1689	0.0388
	f_{best}	0.8197*	0.81971	0.8197*
3	Avg	80.3776	80.3752*	80.3752*
	Mdn	80.3752	80.3752*	80.3752*
	Sd	0.3067	0.2474	0.0283
	f_{best}	80.3752*	80.3752*	80.3752*
4	Avg	- 0.3933	- 0.3966*	- 0.3966*
	Mdn	- 0.3957	- 0.3966*	- 0.3966*
	Sd	0.0142	0.0133	0.0021
	f_{best}	- 0.3966*	- 0.3966*	- 0.3966*
5	Avg	- 0.2411	- 0.2444	- 0.2716*
	Mdn	- 0.2432	- 0.2716*	- 0.2716*
	Sd	0.1008	0.1267	0.0188
	f_{best}	- 0.2715	- 0.2716*	- 0.2716*
6	Avg	1.4492	1.3766	1.3656*
	Mdn	1.443	1.2612*	1.2612*
	Sd	0.2652	0.2627	0.1938
	f_{best}	1.272	1.2612*	1.2612*
7	Avg	140.6311	140.5467	140.4705*
	Mdn	140.5874	140.5267	140.4699*
	Sd	0.8632	0.8382	0.3094
	f_{best}	140.4694	140.4722	140.4693*
8	Avg	- 0.0849	- 0.1005	- 0.101*
	Mdn	- 0.0850	- 0.101*	- 0.101*
	Sd	0.0147	0.0134	0.0044
	f_{best}	- 0.0995	- 0.101*	- 0.101*
9	Avg	1.2882	1.3028	1.2813*
	Mdn	1.2806	1.277*	1.277*
	Sd	0.0523	0.0555	0.0239
	f_{best}	1.277*	1.277*	1.277*
10	Avg	56.9274	56.1322	55.8338*
	Mdn	56.8232	56.0079	55.7957*
	Sd	10.7861	10.0182	0.6052
	f_{best}	55.8051	55.8166	55.7954*

Table 4 contains the average errors of the solutions found by the methods. The results have been averaged over 3000 iterations (over all the iterations). The last row of Table 4 provides the mean square error over all the test problems. Based on the table, we can note which method has the lower error on average. The results, in Table 4, demonstrate that FRE-ACO produces high quality solutions with low average error (good accuracy) compared with the solutions obtained by the other methods.

Table 4
Errors averaged over 3000 iterations (obtained from 30 runs, each of which contains 100 iterations).

#	GA(Lu and Fang)	GA(Ghodousian et. al.)	FRE-ACO
1	0.0141	0.0094	0.0002
2	0.1054	0.0366	0.0063
3	0.0882	0.0257	0.0024
4	0.0089	0.0025	0.0002
5	0.0883	0.0473	0.0034
6	0.3361	0.1692	0.1159
7	0.5888	0.4755	0.1062
8	0.0236	0.0049	0.0011
9	0.0321	0.0276	0.0050
10	5.2138	2.9753	0.2768
MSE	2.7672	0.9112	0.0101

The number of function evaluations required in the execution of each method is reported in Table 5. The results have been averaged over 30 runs. As mentioned in Section 5, Lu and Fang's GA may produce a different number of new solutions by the crossover operator at an iteration. So, in contrast to the other algorithms, Lu and Fang's GA uses a different number of function evaluations for solving the test problems. As is shown in this table, FRE-ACO requires the least number of function evaluations for all the test problems compared to the other considered algorithms.

Table 5
Number of Function Evaluations averaged over 30 runs.

#	GA (Lu & Fang)	GA (Ghodousian et al.)	FRE-ACO
1	351.2333	350	347
2	350.9667	350	347
3	350.9333	350	347
4	351.0333	350	347
5	351.1	350	347
6	351.1	350	347
7	350.9	350	347
8	350.9667	350	347
9	351.2	350	347
10	351.1	350	347

The main advantage of FRE-ACO is remarkable convergence rate, while it is competitive in terms of quality too. Particularly, FRE-ACO significantly outperforms the mentioned genetic algorithms in term of convergence rate and is better in most cases in terms of quality or is equal in the worst case. Also, by considering the notion of the complexity, suppose that algorithms A and B have the same output value (e.g., convergence rate) by performing k and $2k$ function evaluations, respectively. So, although A and B have the same output, the performance of algorithm A is reasonably two times better than that of algorithm B. Therefore, by considering the output values with respect to the number of function evaluations, we can make a fair comparison between the performances of the algorithms [59]. Table 6 shows the results of Related-Samples Wilcoxon Signed Rank Test to illustrate significant difference between the algorithms.

Table 6
RELATED TEST WILCOXON- CRITERIA ALPHA=0.05 CILEVEL=95

	Difference between FRE-ACO and GA (Lu & Fang)	Difference between FRE-ACO and GA (Ghodousian et al.)
Mean of Negative Ranks	5.50	4.50
Mean of Positive Ranks	0.00	0.00
Sum of Negative Ranks	55.00	36.00
Sum of Positive Ranks	0.00	0.00
Avg	158.2201	98.3256
SD	256.4982	162.0671
SEM	81.1118	51.2501
Z	2.803	2.521
p-value	.005	.012

According to the results, the negative and positive ranks of the difference of both groups of variables and the sum of the negative and positive marked ranks can be seen in the table. The number, average and sum of negative ranks are more than positive ranks, it can be concluded that the results obtained from the FRE-ACO are smaller than GA (Lu & Fang) and GA (Ghodousian et al.). The asymptotic probability value for the test for the hypothesis that there is no difference between GA (Lu & Fang) and FRE-ACO, it is equal to 0.005. Also, the asymptotic probability value for the test for the hypothesis that there is no difference between the GA (Ghodousian et al.) and FRE-ACO, is equal to 0.012. According to the following relationships, the null hypothesis is rejected. This means that there is a significant difference between the algorithms.

1. $\text{Sig.} = 0.005 < 0.05$

2. $\text{Sig.} = 0.012 < 0.05$

The Z with 2.803 and 2.521 values at 5 percent significance level also shows this differences.

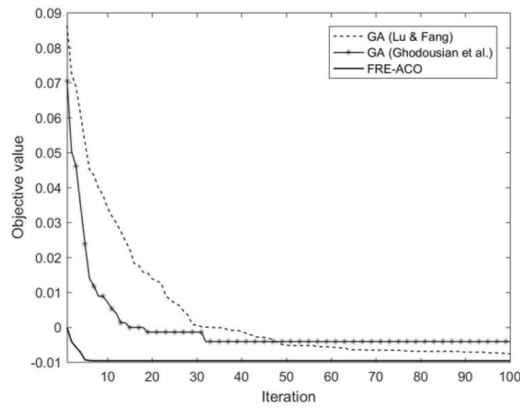


Figure 3. The performance of the algorithms on test problem 1.

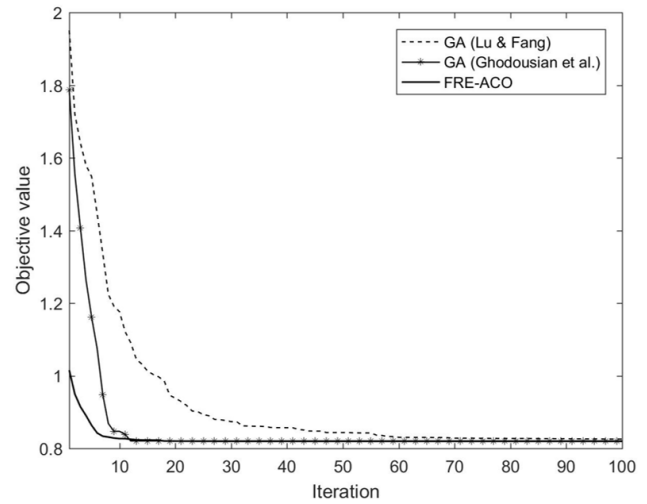


Figure 4. The performance of the algorithms on test problem 2.

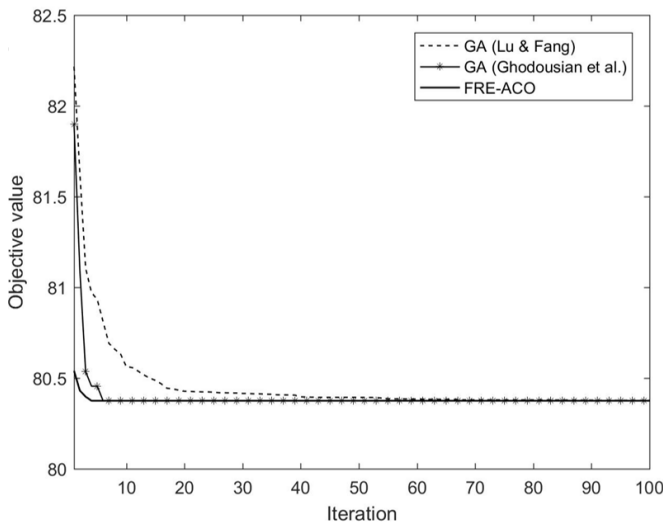


Figure 5. The performance of the algorithms on test problem 3.

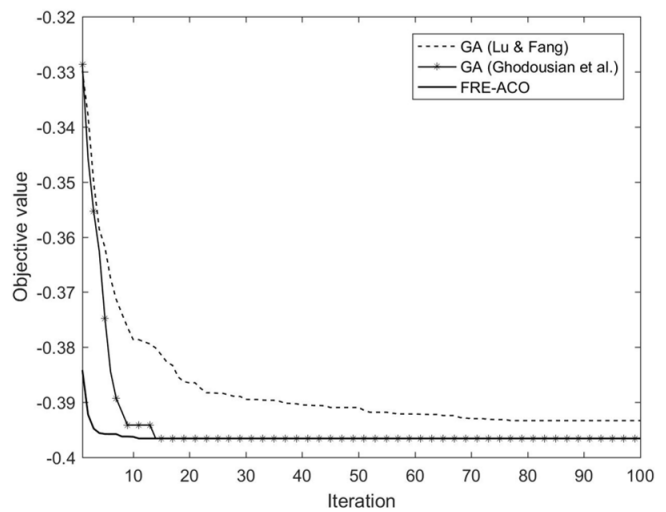


Figure 6. The performance of the algorithms on test problem 4.

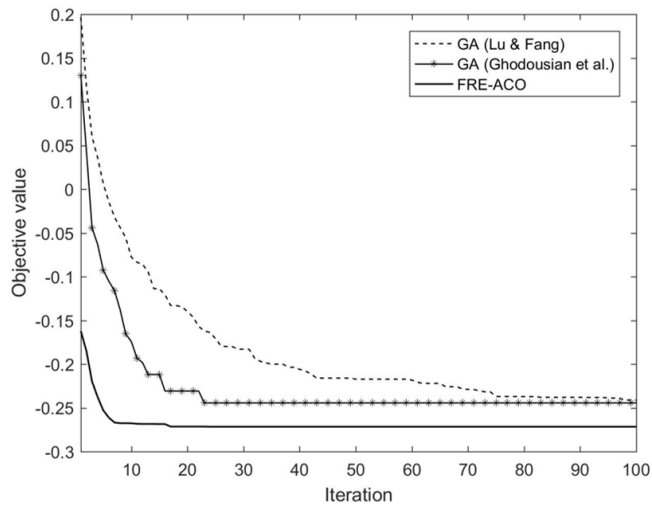


Figure 7. The performance of the algorithms on test problem 5.

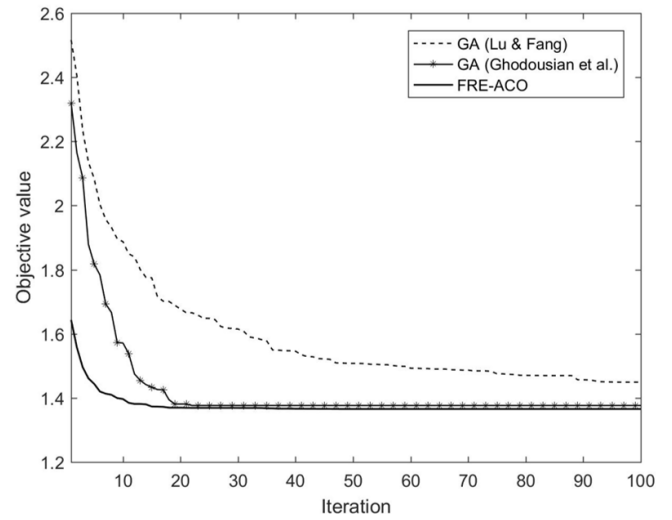


Figure 8. The performance of the algorithms on test problem 6.

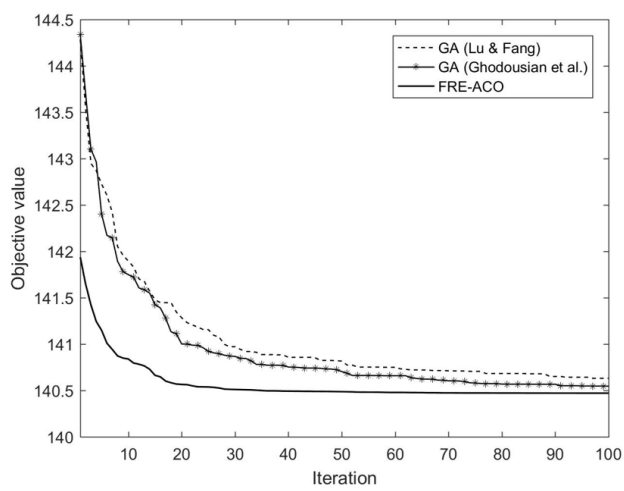


Figure 9. The performance of the algorithms on test problem 7.

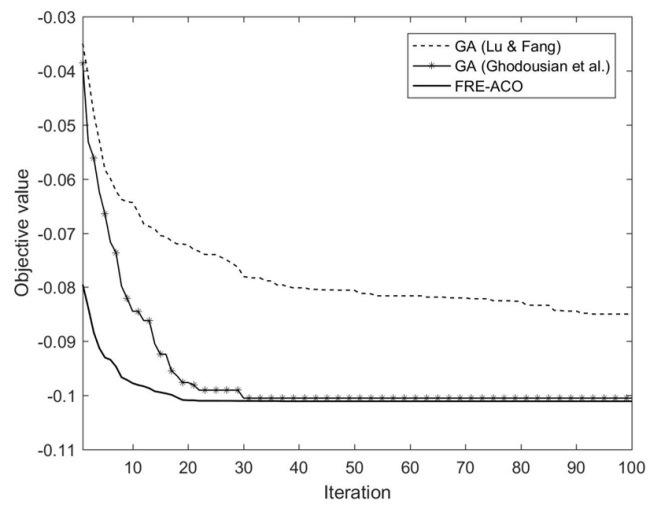


Figure 10. The performance of the algorithms on test problem 8.

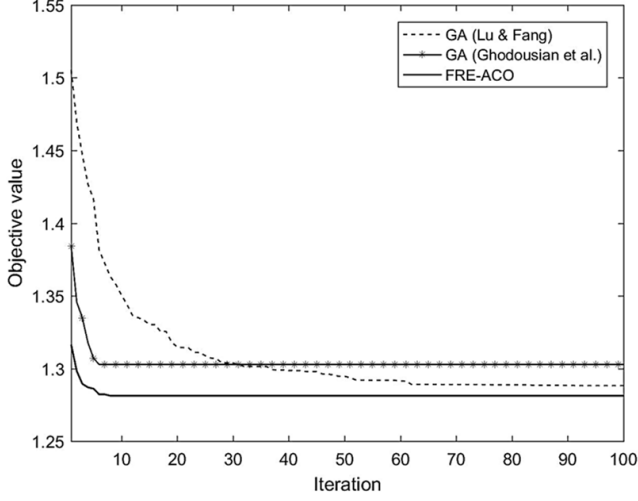


Figure 11. The performance of the algorithms on test problem 9.

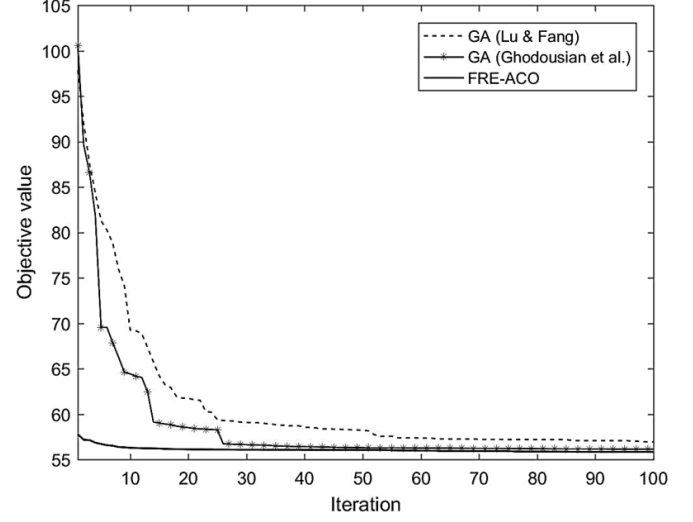


Figure 12. The performance of the algorithms on test problem 10.

7 Conclusion

In this paper, an ACO-based algorithm (FRE-ACO) was proposed for solving nonlinear optimization problems with fuzzy relational equations defined by the minimum t-norm. In the proposed algorithm, the idea of both discrete and continuous ant colony optimization algorithms were used to simultaneously tackle the resolution of the feasible region (that is an NP-hard combinatorial problem [4, 19, 32, 40, 58, 67]) and the optimization of nonlinear functions subjected to non-convex regions. Experiments were performed by the proposed method for the ten randomly generated feasible test problems. We conclude that FRE-ACO can find the optimal solutions for all the cases with a great convergence rate. Moreover, a comparison was made between FRE-ACO algorithm and two genetic algorithms (GA), Lu and Fang's GA [38] and Ghodousian and Babalhavaeji's GA [15], which solve the nonlinear optimization problems subjected to the FREs defined by max-min composition. The results showed that the proposed method finds better solutions (or the same solutions with a higher convergence rate) compared with the solutions obtained by the other algorithms. Furthermore, FRE-ACO produces high quality solutions with a less average error (good accuracy) and less standard deviation (good stability) for all the test problems. Moreover, the results demonstrated that FRE-ACO requires the least number of function evaluations when compared against the considered genetic algorithms. As the future work, we investigate other heuristic algorithms to solve the nonlinear optimization problems restricted to fuzzy relational equations.

Appendix

Test Problem 1

$$f(x) = \ln(0.5 + x_1^2 x_2 + x_3) - x_4^2 + x_5 x_6,$$

$$A = \begin{bmatrix} 0.1795 & 0.9194 & 0.6636 & 0.2774 & 0.1598 & 0.8240 \\ 0.6077 & 0.4035 & 0.7116 & 0.7843 & 0.2370 & 0.6523 \\ 0.9365 & 0.3045 & 0.2486 & 0.3578 & 0.7174 & 0.5950 \\ 0.1200 & 0.1041 & 0.0306 & 0.1692 & 0.0409 & 0.2391 \end{bmatrix},$$

$$b^T = [0.8501, 0.5064, 0.3178, 0.1263].$$

Test Problem 2

$$f(x) = \sin(x_1 x_2) + (1 - \cos(x_1 x_3)) + x_4 + x_5^2 + x_6^3,$$

$$A = \begin{bmatrix} 0.7515 & 0.0275 & 0.8476 & 0.9867 & 0.9788 & 0.8664 \\ 0.8268 & 0.1254 & 0.8037 & 0.5647 & 0.9216 & 0.0806 \\ 0.3641 & 0.5033 & 0.5515 & 0.5515 & 0.1354 & 0.0200 \\ 0.0826 & 0.5188 & 0.2551 & 0.1630 & 0.1430 & 0.1882 \\ 0.6395 & 0.7118 & 0.2717 & 0.3305 & 0.2062 & 0.0968 \\ 0.8624 & 0.0310 & 0.0027 & 0.0939 & 0.0637 & 0.1416 \end{bmatrix},$$

$$b^T = [0.8629, 0.8629, 0.5515, 0.3789, 0.3789, 0.1846].$$

Test Problem 3

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 - x_5 - x_6 + (2x_7 + x_8)^2,$$

$$A = \begin{bmatrix} 0.7715 & 0.1104 & 0.8930 & 0.2939 & 0.9973 & 0.8488 & 0.7017 & 0.7097 \\ 0.7837 & 0.9825 & 0.6753 & 0.7218 & 0.3572 & 0.1096 & 0.0955 & 0.2023 \\ 0.2266 & 0.5948 & 0.9205 & 0.1939 & 0.0579 & 0.2825 & 0.8445 & 0.8758 \\ 0.2706 & 0.2968 & 0.3066 & 0.5945 & 0.3039 & 0.9097 & 0.1923 & 0.7678 \\ 0.8174 & 0.3458 & 0.8938 & 0.6455 & 0.1174 & 0.1441 & 0.4414 & 0.5699 \\ 0.0249 & 0.0908 & 0.2651 & 0.6559 & 0.3363 & 0.2911 & 0.8178 & 0.3700 \\ 0.4353 & 0.2784 & 0.2792 & 0.9595 & 0.0716 & 0.2301 & 0.7262 & 0.2997 \\ 0.8223 & 0.1216 & 0.1137 & 0.2890 & 0.0377 & 0.0720 & 0.0247 & 0.0424 \end{bmatrix},$$

$$b^T = [0.9701, 0.8656, 0.836, 0.7911, 0.6082, 0.4634, 0.4634, 0.3614].$$

Test Problem 4

$$f(x) = x_1 - x_2 - x_3 - x_1x_3x_5 + x_1x_4x_6 + x_2x_3x_7 - x_2x_4x_8,$$

$$A = \begin{bmatrix} 0.4396 & 0.2216 & 0.1336 & 0.4682 & 0.2772 & 0.8983 & 0.0122 & 0.9332 \\ 0.9873 & 0.4027 & 0.3235 & 0.7378 & 0.1593 & 0.2108 & 0.9007 & 0.5111 \\ 0.8249 & 0.8397 & 0.0379 & 0.5054 & 0.5326 & 0.0086 & 0.5167 & 0.5803 \\ 0.6048 & 0.3073 & 0.2749 & 0.8800 & 0.9793 & 0.8859 & 0.0555 & 0.4956 \\ 0.3716 & 0.9327 & 0.4145 & 0.9292 & 0.9334 & 0.8875 & 0.1859 & 0.4239 \\ 0.2154 & 0.6681 & 0.8953 & 0.9269 & 0.9579 & 0.6790 & 0.0914 & 0.4114 \\ 0.1082 & 0.2890 & 0.0888 & 0.0474 & 0.1065 & 0.1732 & 0.0550 & 0.0853 \\ 0.0141 & 0.0057 & 0.0098 & 0.0094 & 0.0011 & 0.8573 & 0.0025 & 0.0104 \end{bmatrix},$$

$$b^T = [0.8964, 0.788, 0.788, 0.6414, 0.6414, 0.6414, 0.1439, 0.0152].$$

Test Problem 5

$$f(x) = x_1x_2x_3x_4x_5 - x_6x_7x_8 + x_9x_{10},$$

$$A = \begin{bmatrix} 0.0600 & 0.2511 & 0.9209 & 0.2276 & 0.9494 & 0.2828 & 0.3557 & 0.7488 & 0.2268 & 0.8434 \\ 0.7095 & 0.4247 & 0.7763 & 0.3525 & 0.1900 & 0.8615 & 0.8905 & 0.5769 & 0.6407 & 0.9594 \\ 0.7882 & 0.9146 & 0.9574 & 0.8091 & 0.7355 & 0.7893 & 0.2965 & 0.2424 & 0.4028 & 0.3092 \\ 0.7316 & 0.9271 & 0.1748 & 0.5008 & 0.2375 & 0.0025 & 0.2371 & 0.9715 & 0.6156 & 0.7659 \\ 0.6527 & 0.6507 & 0.0047 & 0.2099 & 0.2427 & 0.7713 & 0.2320 & 0.6045 & 0.9449 & 0.2445 \\ 0.4446 & 0.7154 & 0.9425 & 0.5271 & 0.3922 & 0.5485 & 0.1432 & 0.3793 & 0.3806 & 0.0079 \\ 0.4234 & 0.7731 & 0.7489 & 0.2736 & 0.3397 & 0.6579 & 0.2917 & 0.7944 & 0.0262 & 0.3958 \\ 0.1634 & 0.6629 & 0.6402 & 0.3769 & 0.4384 & 0.6023 & 0.2094 & 0.0423 & 0.5819 & 0.5587 \end{bmatrix},$$

$$b^T = [0.9, 0.7898, 0.7898, 0.6392, 0.6392, 0.5864, 0.5864, 0.5864].$$

Test Problem 6

$$f(x) = x_1 + 2x_2 + 4x_5 + e^{x_1x_4x_6} - x_7x_8e^{2x_9-x_{10}},$$

$$A = \begin{bmatrix} 0.3929 & 0.9766 & 0.6832 & 0.5287 & 0.3723 & 0.8749 & 0.0264 & 0.9018 & 0.6531 & 0.9819 \\ 0.3702 & 0.1480 & 0.8312 & 0.4065 & 0.1637 & 0.4962 & 0.9869 & 0.0881 & 0.621 & 0.3531 \\ 0.2097 & 0.7964 & 0.3923 & 0.4738 & 0.2785 & 0.1016 & 0.2955 & 0.3064 & 0.9609 & 0.7242 \\ 0.3012 & 0.6326 & 0.1887 & 0.9143 & 0.1486 & 0.4006 & 0.4166 & 0.8941 & 0.5663 & 0.0352 \\ 0.6560 & 0.2583 & 0.1 & 0.8502 & 0.9685 & 0.8324 & 0.959 & 0.4153 & 0.5783 & 0.347 \\ 0.6323 & 0.3277 & 0.3326 & 0.2621 & 0.9914 & 0.6775 & 0.3566 & 0.6403 & 0.3587 & 0.1329 \\ 0.0060 & 0.9824 & 0.0962 & 0.1946 & 0.7119 & 0.4264 & 0.0015 & 0.2420 & 0.1303 & 0.0409 \\ 0.0841 & 0.8233 & 0.0659 & 0.1416 & 0.1047 & 0.3487 & 0.1516 & 0.0203 & 0.0451 & 0.0250 \\ 0.0462 & 0.4195 & 0.0872 & 0.0782 & 0.3259 & 0.4444 & 0.0940 & 0.3063 & 0.0446 & 0.0207 \end{bmatrix},$$

$$b^T = [0.9264, 0.7977, 0.7389, 0.5941, 0.5941, 0.4387, 0.2327, 0.2327, 0.2327].$$

Test Problem 7

$$f(x) = \sum_{k=1}^9 [100(x_{k+1} - x_k^2)^2 + (1 - x_k)^2],$$

$$A = \begin{bmatrix} 0.812 & 0.6281 & 0.063 & 0.9803 & 0.5762 & 0.0276 & 0.7127 & 0.7648 & 0.8193 & 0.6619 \\ 0.2792 & 0.8521 & 0.3791 & 0.2889 & 0.7784 & 0.9202 & 0.2201 & 0.3630 & 0.7759 & 0.5705 \\ 0.5447 & 0.4706 & 0.8592 & 0.0901 & 0.1094 & 0.3421 & 0.2168 & 0.6405 & 0.9930 & 0.8925 \\ 0.1321 & 0.817 & 0.0413 & 0.4569 & 0.7193 & 0.2761 & 0.8406 & 0.1417 & 0.7484 & 0.2615 \\ 0.7253 & 0.5051 & 0.8847 & 0.2749 & 0.4410 & 0.4622 & 0.5527 & 0.1059 & 0.1547 & 0.8351 \\ 0.4919 & 0.2471 & 0.9033 & 0.0069 & 0.1081 & 0.1483 & 0.0753 & 0.0198 & 0.4540 & 0.3356 \\ 0.3794 & 0.5373 & 0.0599 & 0.0794 & 0.1231 & 0.1716 & 0.1662 & 0.0796 & 0.1595 & 0.8758 \end{bmatrix},$$

$$b^T = [0.9303, 0.7619, 0.6297, 0.5097, 0.4705, 0.2733, 0.2619].$$

Test Problem 8

$$f(x) = -0.5(x_1x_4 - x_2x_3 + x_2x_6 - x_5x_6 + x_4x_5 - x_6x_7 + x_8x_{10} - x_9x_{10}),$$

$$A = \begin{bmatrix} 0.2538 & 0.9943 & 0.5048 & 0.5869 & 0.1514 & 0.5405 & 0.1415 & 0.5711 & 0.4177 & 0.1420 \\ 0.2186 & 0.6899 & 0.1047 & 0.8187 & 0.7933 & 0.9628 & 0.7216 & 0.0617 & 0.1854 & 0.1476 \\ 0.4523 & 0.6951 & 0.8131 & 0.8095 & 0.1097 & 0.0215 & 0.7911 & 0.3582 & 0.635 & 0.6447 \\ 0.6647 & 0.4005 & 0.0798 & 0.8146 & 0.0883 & 0.1039 & 0.9671 & 0.4607 & 0.0305 & 0.9200 \\ 0.8746 & 0.1981 & 0.4623 & 0.7306 & 0.9437 & 0.3698 & 0.1068 & 0.5779 & 0.2977 & 0.6884 \\ 0.0370 & 0.1628 & 0.0232 & 0.0991 & 0.4626 & 0.0690 & 0.0706 & 0.0941 & 0.7398 & 0.0947 \\ 0.7203 & 0.0123 & 0.0470 & 0.0132 & 0.0576 & 0.0886 & 0.7022 & 0.5164 & 0.9357 & 0.0713 \end{bmatrix},$$

$$b^T = [0.9718, 0.878, 0.7243, 0.568, 0.568, 0.1984, 0.1006].$$

Test Problem 9

$$f(x) = e^{x_1x_2+x_3x_6+x_7x_9} - 0.5(x_1^3 + x_2^3 + x_8^3 + x_{10}^3 + 1)^2,$$

$$A = \begin{bmatrix} 0.8011 & 0.4018 & 0.7035 & 0.9869 & 0.1705 & 0.9656 & 0.9832 & 0.8364 & 0.7133 & 0.7508 \\ 0.2569 & 0.4875 & 0.222 & 0.0709 & 0.9384 & 0.3757 & 0.9174 & 0.31 & 0.9603 & 0.408 \\ 0.8807 & 0.7017 & 0.7196 & 0.1238 & 0.0061 & 0.6642 & 0.9998 & 0.7438 & 0.1750 & 0.9458 \\ 0.7750 & 0.7051 & 0.1353 & 0.4416 & 0.2704 & 0.0217 & 0.0532 & 0.6864 & 0.6955 & 0.0207 \\ 0.7518 & 0.0550 & 0.9604 & 0.3945 & 0.1807 & 0.7580 & 0.1978 & 0.0644 & 0.2631 & 0.7684 \\ 0.9177 & 0.5041 & 0.9828 & 0.2669 & 0.2915 & 0.1256 & 0.0094 & 0.9482 & 0.034 & 0.6577 \\ 0.8391 & 0.2024 & 0.6265 & 0.2043 & 0.3186 & 0.0997 & 0.0483 & 0.0232 & 0.3402 & 0.1631 \\ 0.543 & 0.9321 & 0.5742 & 0.0956 & 0.3085 & 0.3283 & 0.0358 & 0.6781 & 0.1433 & 0.1817 \\ 0.2430 & 0.6933 & 0.9497 & 0.0620 & 0.0937 & 0.0165 & 0.1313 & 0.1493 & 0.1676 & 0.1608 \\ 0.2594 & 0.8109 & 0.9014 & 0.0829 & 0.0854 & 0.2434 & 0.0184 & 0.0365 & 0.1109 & 0.2225 \end{bmatrix},$$

$$b^T = [0.9758, 0.9288, 0.9288, 0.6185, 0.5077, 0.4076, 0.3434, 0.3434, 0.2977, 0.2977].$$

Test Problem 10

$$f(x) = (x_1 - 1)^2 + (x_7 - 1)^2 + 10 \sum_{k=1}^{11} (10 - k)(x_k^2 - x_{k+1})^2,$$

$$A = \begin{bmatrix} 0.0702 & 0.1745 & 0.5925 & 0.2656 & 0.8812 & 0.6833 & 0.003 & 0.5535 & 0.7155 & 0.8598 & 0.4763 & 0.9709 \\ 0.963 & 0.5863 & 0.1001 & 0.828 & 0.2939 & 0.5813 & 0.2774 & 0.5876 & 0.5299 & 0.9017 & 0.6051 & 0.8993 \\ 0.7901 & 0.5698 & 0.8892 & 0.5924 & 0.4591 & 0.1807 & 0.3446 & 0.5325 & 0.9014 & 0.4529 & 0.7917 & 0.345 \\ 0.6321 & 0.7162 & 0.6132 & 0.2433 & 0.6247 & 0.0607 & 0.9719 & 0.1374 & 0.3581 & 0.1596 & 0.8537 & 0.3707 \\ 0.8638 & 0.8306 & 0.6024 & 0.9979 & 0.2076 & 0.5049 & 0.7025 & 0.1535 & 0.9984 & 0.5087 & 0.1401 & 0.7828 \\ 0.2246 & 0.3024 & 0.5784 & 0.1239 & 0.5353 & 0.5777 & 0.2191 & 0.466 & 0.2639 & 0.1992 & 0.8516 & 0.0107 \\ 0.6065 & 0.5062 & 0.5623 & 0.1472 & 0.0783 & 0.9427 & 0.2715 & 0.4176 & 0.2902 & 0.1834 & 0.9231 & 0.0426 \\ 0.9605 & 0.0447 & 0.794 & 0.0357 & 0.1938 & 0.3113 & 0.2229 & 0.5738 & 0.0477 & 0.1005 & 0.9174 & 0.1864 \\ 0.0254 & 0.4711 & 0.3339 & 0.2237 & 0.0934 & 0.7857 & 0.2072 & 0.1740 & 0.1953 & 0.0672 & 0.2243 & 0.2333 \\ 0.0444 & 0.1702 & 0.5993 & 0.0805 & 0.1675 & 0.8279 & 0.0647 & 0.1368 & 0.1043 & 0.1891 & 0.1084 & 0.1822 \end{bmatrix},$$

$$b^T = [0.9615, 0.9002, 0.9002, 0.9002, 0.9002, 0.4477, 0.3132, 0.3132, 0.2893, 0.2256].$$

References

- [1] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear programming: Theory and algorithms*, John Wiley and Sons, New York, NY, 2006. <https://doi.org/10.1002/0471787779>
- [2] M. M. Bourke, D. G. Fisher, *Solution algorithms for fuzzy relational equations with max-product composition*, Fuzzy Sets and Systems, **94** (1998), 61-69. [https://doi.org/10.1016/S0165-0114\(96\)00246-1](https://doi.org/10.1016/S0165-0114(96)00246-1)
- [3] C. W. Chang, B. S. Shieh, *Linear optimization problem constrained by fuzzy max-min relation equations*, Information Sciences, **234** (2013), 71-79. <https://doi.org/10.1016/j.ins.2011.04.042>
- [4] L. Chen, P. P. Wang, *Fuzzy relation equations (ii): The branch-point-solutions and the categorized minimal solutions*, Soft Computing, **11**(2) (2007), 33-40. <https://doi.org/10.1007/s00500-006-0050-1>
- [5] S. Dempe, A. Ruziyeva, *On the calculation of a membership function for the solution of a fuzzy linear optimization problem*, Fuzzy Sets and Systems, **188** (2012), 58-67. <https://doi.org/10.1016/j.fss.2011.07.014>
- [6] F. Di Martino, V. Loia, S. Sessa, *Digital watermarking in coding/decoding processes with fuzzy relation equations*, Soft Computing, **10** (2006), 238-243. <https://doi.org/10.1007/s00500-005-0477-9>
- [7] A. Di Nola, S. Sessa, W. Pedrycz, E. Sanchez, *Fuzzy relational equations and their applications in knowledge engineering*, Dordrecht: Kluwer Academic Press, 1989. <https://doi.org/10.1007/978-94-017-1650-5>
- [8] M. Dorigo, V. Maniezzo, A. Coloni, *Ant system: Optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man and Cybernetics, **26**(1) (1996), 29-41. <https://doi.org/10.1109/3477.484436>
- [9] D. Dubey, S. Chandra, A. Mehra, *Fuzzy linear programming under interval uncertainty based on IFS representation*, Fuzzy Sets and Systems, **188** (2012), 68-87. <https://doi.org/10.1016/j.fss.2011.09.008>
- [10] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, T. Desell, *Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration*, Applied Soft Computing, **73** (2018), 969-991. <https://doi.org/10.1016/j.asoc.2018.09.013>
- [11] Y. R. Fan, G. H. Huang, A. L. Yang, *Generalized fuzzy linear programming for decision making under uncertainty: Feasibility of fuzzy solutions and solving approach*, Information Sciences, **241** (2013), 12-27. <https://doi.org/10.1016/j.ins.2013.04.004>
- [12] S. C. Fang, G. Li, *Solving fuzzy relation equations with a linear objective function*, Fuzzy Sets and Systems, **103** (1999), 107-113. [https://doi.org/10.1016/S0165-0114\(97\)00184-X](https://doi.org/10.1016/S0165-0114(97)00184-X)
- [13] S. Freson, B. De Baets, H. De Meyer, *Linear optimization with bipolar max-min constraints*, Information Sciences, **234** (2013), 3-15. <https://doi.org/10.1016/j.ins.2011.06.009>

- [14] A. Ghodousian, *Optimization of linear problems subjected to the intersection of two fuzzy relational inequalities defined by Dubois-Prade family of t-norms*, Information Sciences, **503** (2019), 291-306. <https://doi.org/10.1016/j.ins.2019.06.058>
- [15] A. Ghodousian, A. Babalhavaeji, *An efficient genetic algorithm for solving nonlinear optimization problems defined with fuzzy relational equations and max-Lukasiewicz composition*, Applied Soft Computing, **69** (2018), 475-492. <https://doi.org/10.1016/j.asoc.2018.04.029>
- [16] A. Ghodousian, M. S. Chopannavaz, *Solving linear optimization problems subject to bipolar fuzzy relational equalities defined with max-strict compositions*, Information Sciences, **650** (2023), 119696. <https://doi.org/10.1016/j.ins.2023.119696>
- [17] A. Ghodousian, E. Khorram, *Fuzzy linear optimization in the presence of the fuzzy relation inequality constraints with max-min composition*, Information Sciences, **178** (2008), 501-519. <https://doi.org/10.1016/j.ins.2007.07.022>
- [18] A. Ghodousian, E. Khorram, *Linear optimization with an arbitrary fuzzy relational inequality*, Fuzzy Sets and Systems, **206** (2012), 89-102. <https://doi.org/10.1016/j.fss.2012.04.009>
- [19] A. Ghodousian, M. Naeemib, A. Babalhavaeji, *Nonlinear optimization problem subjected to fuzzy relational equations defined by Dubois-Prade family of t-norms*, Computers and Industrial Engineering, **119** (2018), 167-180. <https://doi.org/10.1016/j.cie.2018.03.038>
- [20] A. Ghodousian, M. Raeisian Parvari, *A modified PSO algorithm for linear optimization problem subject to the generalized fuzzy relational inequalities with fuzzy constraints (FRI-FC)*, Information Sciences, **418-419** (2017), 317-345. <https://doi.org/10.1016/j.ins.2017.07.032>
- [21] A. Ghodousian, B. Sepehri Rad, O. Ghodousian, *A non-linear generalization of optimization problems subjected to continuous max-t-norm fuzzy relational inequalities*, Soft Computing, **28** (2024), 4025-4036. <https://doi.org/10.1007/s00500-023-09376-2>
- [22] A. Ghodousian, F. S. Yousefi, *Linear optimization problem subjected to fuzzy relational equations and fuzzy constraints*, Iranian Journal of Fuzzy Systems, **20**(2) (2023), 1-20. <https://doi.org/10.22111/IJFS.2023.7552>
- [23] F. F. Guo, Z. Q. Xia, *An algorithm for solving optimization problems with one linear objective function and finitely many constraints of fuzzy relation inequalities*, Fuzzy Optimization and Decision Making, **5** (2006), 33-47. <https://doi.org/10.1007/s10700-005-4914-0>
- [24] S. M. Guu, Y. K. Wu, *Minimizing a linear objective function with fuzzy relation equation constraints*, Fuzzy Optimization and Decision Making, **12** (2002), 1568-4539. <https://doi.org/10.1023/A:1020955112523>
- [25] S. M. Guu, Y. K. Wu, *Minimizing a linear objective function under a max-t-norm fuzzy relational equation constraint*, Fuzzy Sets and Systems, **161** (2010), 285-297. <https://doi.org/10.1016/j.fss.2009.03.007>
- [26] M. Higashi, G. J. Klir, *Resolution of finite fuzzy relation equations*, Fuzzy Sets and Systems, **13** (1984), 65-82. [https://doi.org/10.1016/0165-0114\(84\)90026-5](https://doi.org/10.1016/0165-0114(84)90026-5)
- [27] K. Krynicki, J. Jaen, E. Navarro, *An ACO-based personalized learning technique in support of people with acquired brain injury*, Applied Soft Computing, **47** (2016), 316-331. <https://doi.org/10.1016/j.asoc.2016.04.039>
- [28] H. C. Lee, S. M. Guu, *On the optimal three-tier multimedia streaming services*, Fuzzy Optimization and Decision Making, **2**(1) (2002), 31-39. <https://doi.org/10.1023/A:1022848114005>
- [29] P. K. Li, S. C. Fang, *On the resolution and optimization of a system of fuzzy relational equations with sup-t composition*, Fuzzy Optimization and Decision Making, **7** (2008), 169-214. <https://doi.org/10.1007/s10700-008-9029-y>
- [30] P. Li, Y. Liu, *Linear optimization with bipolar fuzzy relational equation constraints using Lukasiewicz triangular norm*, Soft Computing, **18** (2014), 1399-1404. <https://doi.org/10.1007/s00500-013-1152-1>
- [31] P. Li, Y. Wang, *A matrix approach to latticized linear programming with fuzzy-relation inequality constraints*, IEEE Transactions on Fuzzy Systems, **21** (2013), 781-788. <https://doi.org/10.1109/TFUZZ.2012.2232932>

- [32] C. H. Lin, *A rough penalty genetic algorithm for constrained optimization*, Information Science (NY), **241** (2013), 119-137. <https://doi.org/10.1016/j.ins.2013.04.001>
- [33] J. L. Lin, Y. K. Wu, S. M. Guu, *On fuzzy relational equations and the covering problem*, Information Science, **181** (2011), 2951-2963. <https://doi.org/10.1016/j.ins.2011.03.004>
- [34] J. Liu, J. Liu, *Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems*, Applied Soft Computing, **74** (2019), 167-189. <https://doi.org/10.1016/j.asoc.2018.10.012>
- [35] C. C. Liu, Y. Y. Lur, Y. K. Wu, *Linear optimization of bipolar fuzzy relational equations with max-Lukasiewicz composition*, Information Sciences, **360** (2016), 149-162. <https://doi.org/10.1016/j.ins.2016.04.041>
- [36] D. Lobo, V. Lopez-Marchante, J. Medina, *On measuring the solvability of a fuzzy relation equation*, In: M. Cornejo, L. T. K'oczy, J. Medina, E. Ram'irez-Poussa, (eds) Computational Intelligence and Mathematics for Tackling Complex Problems 5. Studies in Computational Intelligence, **1127**. Springer, Cham, (2024). <https://doi.org/10.1007/978-3-031-46979-4>
- [37] J. Loetamonphong, S. C. Fang, *Optimization of fuzzy relation equations with max-product composition*, Fuzzy Sets and Systems, **118** (2001), 509-517. [https://doi.org/10.1016/S0165-0114\(98\)00417-5](https://doi.org/10.1016/S0165-0114(98)00417-5)
- [38] J. Lu, S. C. Fang, *Solving nonlinear optimization problems with fuzzy relation equations constraints*, Fuzzy Sets and Systems, **119** (2001), 1-20. [https://doi.org/10.1016/S0165-0114\(98\)00471-0](https://doi.org/10.1016/S0165-0114(98)00471-0)
- [39] L. C. Lu, T. W. Yue, *Mission-oriented ant-team ACO for min-max MTSP*, Applied Soft Computing, **76** (2019), 436-444. <https://doi.org/10.1016/j.asoc.2018.11.048>
- [40] J. Lv, X. Wang, M. Huang, *ACO-inspired ICN routing mechanism with mobility support*, Applied Soft Computing, **58** (2017), 427-440. <https://doi.org/10.1016/j.asoc.2017.04.040>
- [41] A. V. Markovskii, *On the relation between equations with max-product composition and the covering problem*, Fuzzy Sets and Systems, **153** (2005), 261-273. <https://doi.org/10.1016/j.fss.2005.02.010>
- [42] Z. Matusiewicz, J. Drewniak, *Increasing continuous operations in fuzzy max-* equations and inequalities*, Fuzzy Sets and Systems, **232** (2013), 120-133. <https://doi.org/10.1016/j.fss.2013.03.009>
- [43] M. Mizumoto, H. J. Zimmermann, *Comparison of fuzzy reasoning method*, Fuzzy Sets and Systems, **8** (1982), 253-283. [https://doi.org/10.1016/S0165-0114\(82\)80004-3](https://doi.org/10.1016/S0165-0114(82)80004-3)
- [44] H. Nobuhara, K. Hirota, F. Di Martino, W. Pedrycz, S. Sessa, *Fuzzy relation equations for compression/decompression processes of color images in the RGB and YUV color spaces*, Fuzzy Optimization and Decision Making, **4** (2005), 235-246. <https://doi.org/10.1007/s10700-005-1892-1>
- [45] W. Pedrycz, *On generalized fuzzy relational equations and their applications*, Journal of Mathematical Analysis and Applications, **107** (1985), 520-536. [https://doi.org/10.1016/0022-247X\(85\)90329-4](https://doi.org/10.1016/0022-247X(85)90329-4)
- [46] W. Pedrycz, *Granular computing: Analysis and design of intelligent systems*, CRC Press, Boca Raton, 2013. <https://doi.org/10.1201/b14862>
- [47] W. Pedrycz, A. V. Vasilakos, *Modularization of fuzzy relational equations*, Soft Computing, **6** (2002), 3-37. <https://doi.org/10.1007/s005000100125>
- [48] K. Peeva, *Resolution of fuzzy relational equations-methods, algorithm and software with applications*, Information Sciences, **234** (2013), 44-63. <https://doi.org/10.1016/j.ins.2011.04.011>
- [49] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, J. M. de la Cruz, *Ant colony optimization for multi-UAV minimum time search in uncertain domains*, Applied Soft Computing, **62** (2018), 789-806. <https://doi.org/10.1016/j.asoc.2017.09.009>
- [50] I. Perfilieva, *Fuzzy function as an approximate solution to a system of fuzzy relation equations*, Fuzzy Sets and Systems, **147** (2004), 363-383. <https://doi.org/10.1016/j.fss.2003.12.007>
- [51] I. Perfilieva, *Finitary solvability conditions for systems of fuzzy relation equations*, Information Sciences, **234** (2013), 29-43. <https://doi.org/10.1016/j.ins.2011.04.035>

- [52] I. Perfilieva, V. Novák, *System of fuzzy relation equations model of IF-THEN rules*, Information Sciences, **177**(16) (2007), 3218-3227. <https://doi.org/10.1016/j.ins.2006.11.006>
- [53] D. Praveen Kumar, A. Tarach, A. Chandra Sekhara Rao, *ACO-based mobile sink path determination for wireless sensor networks under non-uniform data constraints*, Applied Soft Computing, **69** (2018), 528-540. <https://doi.org/10.1016/j.asoc.2018.05.008>
- [54] X. B. Qu, X. P. Wang, M. H. Lei, *Conditions under which the solution sets of fuzzy relational equations over complete Brouwerian lattices form lattices*, Fuzzy Sets and Systems, **234** (2014), 34-45. <https://doi.org/10.1016/j.fss.2013.03.017>
- [55] W. Rudin, *Principles of mathematical analysis*, New York: McGraw-Hill, 1976.
- [56] E. Sanchez, *Resolution of composite fuzzy relation equations*, Information and Control, **30** (1976), 38-48. [https://doi.org/10.1016/S0019-9958\(76\)90446-0](https://doi.org/10.1016/S0019-9958(76)90446-0)
- [57] E. Sanchez, *Solution in composite fuzzy relation equations: Application to medical diagnosis in Brouwerian logic*, in: M. M. Gupta, G. N. Saridis, B. R. Games (Eds.), Fuzzy Automata and Decision Processes, North-Holland, New York, (1977), 221-234. <https://doi.org/10.1016/B978-1-4832-1450-4.50017-1>
- [58] B. S. Shieh, *Minimizing a linear objective function under a fuzzy max-t-norm relation equation constraint*, Information Sciences, **181** (2011), 832-841. <https://doi.org/10.1016/j.ins.2010.10.024>
- [59] K. Socha, M. Dorigo, *Ant colony optimization for continuous domain*, European Journal of Operational Research, **185** (2008), 1155-1173. <https://doi.org/10.1016/j.ejor.2006.06.046>
- [60] I. Stanković, Z. Jančić, M. Ćirić, I. Micić, S. Stanimirović, *Two-mode weakly linear systems of fuzzy relation equations: Structures of solutions, computation methods, and applications*, Information Sciences, **686** (2025), 121319. <https://doi.org/10.1016/j.ins.2024.121319>
- [61] F. Sun, X. P. Wang, X. B. Qu, *Minimal join decompositions and their applications to fuzzy relation equations over complete Brouwerian lattices*, Information Sciences, **224** (2013), 143-151. <https://doi.org/10.1016/j.ins.2012.10.038>
- [62] P. Z. Wang, *Latticized linear programming and fuzzy relation inequalities*, Journal of Mathematical Analysis and Applications, **159**(1) (1991), 72-87. [https://doi.org/10.1016/0022-247X\(91\)90222-L](https://doi.org/10.1016/0022-247X(91)90222-L)
- [63] Y. K. Wu, *Optimization of fuzzy relational equations with max-av composition*, Information Sciences, **177** (2007), 4216-4229. <https://doi.org/10.1016/j.ins.2007.02.037>
- [64] Y. K. Wu, S. M. Guu, *A note on fuzzy relation programming problems with max-strict-t-norm composition*, Fuzzy Optimization and Decision Making, **3** (2004), 271-278. <https://doi.org/10.1023/B:FODM.0000036862.45420.ea>
- [65] Y. K. Wu, S. M. Guu, *Minimizing a linear function under a fuzzy max-min relational equation constraints*, Fuzzy Sets and Systems, **150** (2005), 147-162. <https://doi.org/10.1016/j.fss.2004.09.010>
- [66] Y. K. Wu, S. M. Guu, *An efficient procedure for solving a fuzzy relation equation with max-Archimedean t-norm composition*, IEEE Transactions on Fuzzy Systems, **16** (2008), 73-84. <https://doi.org/10.1109/TFUZZ.2007.902018>
- [67] Y. K. Wu, S. M. Guu, J. Y. Liu, *Reducing the search space of a linear fractional programming problem under fuzzy relational equations with max-Archimedean t-norm composition*, Fuzzy Sets and Systems, **159** (2008), 3347-3359. <https://doi.org/10.1016/j.fss.2008.04.007>
- [68] Q. Q. Xiong, X. P. Wang, *Fuzzy relational equations on complete Brouwerian lattices*, Information Sciences, **193** (2012), 141-152. <https://doi.org/10.1016/j.ins.2011.12.030>
- [69] Y. Yan, H. S. Sohn, G. Reyes, *A modified ant system to achieve better balance between intensification and diversification for the traveling salesman problem*, Applied Soft Computing, **60** (2017), 256-267. <https://doi.org/10.1016/j.asoc.2017.06.049>
- [70] X. P. Yang, X. G. Zhou, B. Y. Cao, *Single-variable term semi-latticized fuzzy relation geometric programming with max-product operator*, Information Sciences, **325** (2015), 271-287. <https://doi.org/10.1016/j.ins.2015.07.015>

- [71] X. P. Yang, X. G. Zhou, B. Y. Cao, *Latticized linear programming subject to max-product fuzzy relation inequalities with application in wireless communication*, Information Sciences, **358-359** (2016), 44-55. <https://doi.org/10.1016/j.ins.2016.04.014>
- [72] X. P. Yang, X. G. Zhou, B. Y. Cao, *Min-max programming problem subject to addition-min fuzzy relation inequalities*, IEEE Transactions on Fuzzy Systems, **24**(1) (2016), 111-119. <https://doi.org/10.1109/TFUZZ.2015.2428716>