Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

On the optimization of Dombi non-linear programming

Amin Ghodousian*[1] and Fatemeh Elyasimohammadi†[2]

[1,2]Faculty of Engineering Science, College of Engineering, University of Tehran, P.O.Box 11365-4563, Tehran, Iran.

ABSTRACT

Dombi family of t-norms includes a parametric family of continuous strict t-norms, whose members are increasing functions of the parameter. This family of t-norms covers the whole spectrum of t-norms when the parameter is changed from zero to infinity. In this paper, we study a nonlinear optimization problem in which the constraints are defined as fuzzy relational equations (FRE) with the Dombi family of t-norms. We firstly investigate the resolution of the feasible solutions set when it is defined with max-Dombi composition and present some necessary and sufficient conditions for determining the feasibility. Also, some procedures are presented for simplifying the problem. Since the feasible solutions set of FREs is non-convex, conventional nonlinear programming methods may not be directly employed to solve the problem. Based on some theoretical properties of the problem, a genetic algorithm is

Keyword: Fuzzy relational equations, nonlinear optimization, genetic algorithm.

ARTICLE INFO

Article history:
Received 11, May 2018
Received in revised form 24, November 2019
Accepted 20 December 2019
Available online 01, June 2020

AMS subject Classification: 05C78.

*Corresponding author:A. Ghodousian. Email: a.ghodousian@ut.ac.ir
†Fatemeelyasi49@gmail.com

1 Abstract continued

presented, which preserves the feasibility of new generated solutions. Moreover, a method is presented to generate feasible max-Dombi FREs as test problems for evaluating the performance of our algorithm. The proposed method has been compared with some related works. The obtained results confirm the high performance of the proposed method in solving such nonlinear problems.

2 Introduction

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

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

where $I = \{1,2,...,m\}$, $J = \{1,2,...,n\}$, $A = (a_{ij})_{m\times n}$, $0 \le a_{ij} \le 1$ ($\forall i \in I$ and $\forall j \in J$), is a fuzzy matrix, $b = (b_i)_{m\times 1}$,$0 \le b_i \le 1$ ($\forall i \in I$), is an m–dimensional fuzzy vector, and "φ" is the max-Dombi composition, that is,

$$\varphi(x,y) = T_D^\lambda(x,y) = \begin{cases} 0 & x = 0 \ or \ y = 0 \\ \frac{1}{1+\left(\left(\frac{1-x}{x}\right)^\lambda + \left(\frac{1-y}{y}\right)^\lambda\right)^{1/\lambda}} & otherwise \end{cases}$$

in which $\lambda > 0$.
If a_i is the i'th row of matrix A, then problem (1) can be expressed as follows:

$$\begin{aligned} &\min \ f(x) \\ &\varphi(a_i, x) = b_i \ , \ i \in I \\ &x \in [0,1]^n \end{aligned}$$

where the constraints mean:

$$\varphi(a_i,x) = \max_{j\in J}\{\varphi(a_{ij},x_j)\} = \max_{j\in J}\{T_D^\lambda(a_{ij},x_j)\} = b_i \quad ,\forall i \in I$$

and

$$T_D^\lambda(a_{ij},x_j) = \begin{cases} 0 & a_{ij} = 0 \ or \ x_j = 0 \\ \frac{1}{1+\left(\left(\frac{1-a_{ij}}{a_{ij}}\right)^\lambda + \left(\frac{1-x_j}{x_j}\right)^\lambda\right)^{1/\lambda}} & otherwise \end{cases}$$

As mentioned, the family $\left\{T_D^\lambda\right\}$ is increasing in λ. On the other hand, Dombi t-norm $T_D^\lambda(x,y)$ converges to the basic fuzzy intersection $\min\{x,y\}$ as λ goes to infinity and converges to Drastic product t-norm as λ approaches zero. Therefore, Dombi t-norm covers the whole spectrum of t-norms [8]. In [40], the Dombi operations of single-valued neutrosophic numbers (SVNNs) were presented based on the operations of the Dombi T-norm and T-conorm. The authors proposed the single-valued neutrosophic Dombi weighted arithmetic average (SVNDWAA) operator and the single-valued neutrosophic Dombi weighted geometric average (SVNDWGA) operator to deal with the aggregation of SVNNs. In [41], a fuzzy morphological approach was presented to detect the edges of real time images in order to preserve their features, where Dombi t-norm and t-conorm was used for computing morphological dilation and erosion. In [42], the authors studied the connection with Dombi aggregative operators, uninorms, strict t-norms and t-conorms. They presented a new representation theorem of strong negations that explicitly contains the neutral value.

The theory of fuzzy relational equations was firstly proposed by Sanchez [51,52]. He introduced a FRE whit max-min composition and applied the model to medical diagnosis in Brouwerian logic. Nowadays, it is well-known that many issues associated with a body knowledge can be treated as FRE problems [43]. In addition to such applications, FRE theory has been applied in many fields including fuzzy control, discrete dynamic systems, prediction of fuzzy systems, fuzzy decision making, fuzzy pattern recognition, fuzzy clustering, image compression and reconstruction, and so on. Pedrycz [44] categorized and extended two ways of the generalizations of FRE in terms of sets under discussion and various operations which are taken into account. Since then, many theoretical improvements have been investigated and many applications have been presented [5,11,18,22,37,45,48,57,59,65]. For example, Klement et al. [28] presented the basic analytical and algebraic properties of triangular norms and important classes of fuzzy operators' generalization such as Archimedean, strict and nilpotent t-norms. In [47] the author demonstrates how problems of interpolation and approximation of fuzzy functions are converted with solvability of systems of fuzzy relational equations.

The solvability and the finding of solutions set are the primary (and the most fundamental) subject concerning FRE problems. Many studies have reported fuzzy relational equations with max-min and max-product compositions. Both compositions are special cases of the max-triangular-norm (max-t-norm). Di Nola et al. proved that the solution set of FRE (if it is nonempty) defined by continuous max-t-norm composition is often a non-convex set that is completely determined by one maximum solution and a finite number of minimal solutions [6]. This non-convexity property is one of two bottlenecks making major contribution to the increase in complexity of problems that are related to FRE, especially in the optimization problems subjected to a system of fuzzy relations. The other bottleneck is concerned with detecting the minimal solutions for FREs. Chen and Wang [2,3] presented an algorithm for obtaining the logical representation of all minimal solutions and deduced that a polynomial-time algorithm to find all minimal solutions of FRE (with max-min compositions) may not exist. Also, Markovskii showed that solving max-product FRE is closely related to the covering problem which is an NP-hard problem [39]. In fact, the same result holds true for more general t-norms instead of the minimum and product operators [33,34]. Lin et al. [34] 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 last decades, the solvability of FRE defined with different max-t compositions has been investigated by many researches [46, 49, 50, 53, 55, 56, 60, 64, 68]. It is worth to mention that Li and Fang [32] provided a complete survey and a detailed discussion on fuzzy relational equations. They studied the relationship among generalized logical operators involved in the construction of FRE and introduced the classification of basic fuzzy relational equations.

Optimizing an objective function subjected to a system of fuzzy relational equations or inequalities (FRI) is one of the most interesting and on-going topics among the problems related to the FRE (or FRI) theory [1,9,13-21,25-27,30,35,54,61,66]. By far the most frequently studied aspect is the determination of a minimizer of a linear objective function and the use of the max-min composition [1,14]. So, it is an almost standard approach to translate this type of problem into a corresponding 0-1 integer linear programming problem, which is then solved using a branch and bound method [10,62]. In [29] an application of optimizing the linear objective with max-min composition was employed for the streaming media provider seeking a minimum cost while fulfilling the requirements assumed by a three-tier framework. Chang and Shieh [1] presented new theoretical results concerning the linear optimization problem constrained by fuzzy max-min relation equations by improving an upper bound on the optimal objective value. The topic of the linear optimization problem was also investigated with max-product operation [13,20,36]. Loetamonphong and Fang defined two sub-problems by separating negative and non-negative coefficients in the objective function and then obtained the optimal solution by combining those of the two sub-problems [36]. Also, in [20] and [13], some necessary conditions of the feasibility and simplification techniques were presented for solving FRE with max-product composition. Moreover, some studies have determined a more general operator of linear optimization with replacement of max-min and max-product compositions with a max-t-norm composition [19, 30, 54], max-average composition [26,61] or max-star composition [16,27].

Recently, many interesting generalizations of the linear and non-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 [4,7,12,14,31,35,63]. For instance, the linear optimization of bipolar FRE was studied by some researchers where FRE was defined with max-min composition [12] and max-Lukasiewicz composition [31,35]. In [31] the authors introduced the optimization problem subjected to a system of bipolar 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 [35], the foregoing problem was solved by an analytical method based on the resolution and some structural properties of the feasible region (using a necessary condition for characterizing an optimal solution and a simplification process for reducing the problem). Ghodousian and khorram [15] focused on the algebraic structure of two fuzzy relational inequalities $A\varphi x \leq b^1$ and $D\varphi x \geq b^2$, and studied a mixed fuzzy system formed by the two preceding FRIs, where φ is an operator with (closed) convex solutions. Yang [67] studied the optimal solution of minimizing a linear objective function subject to fuzzy relational inequalities where the constraints defined as $a_{i1}\wedge x_1+a_{i2}\wedge x_2+...+a_{in}\wedge x_n \geq b_i$ for $i = 1,...,m$ and $a\wedge b = \min\{a,b\}$. He presented an algorithm based on some properties of the minimal solutions of the FRI. In [14], the authors introduced

FRI-FC problem min{ $c^T x \;:\; A\varphi x \tilde{\leq} b \;,\; x \in [0,1]^n$ }, where φ is max-min composition and " $\tilde{\leq}$" denotes the relaxed or fuzzy version of the ordinary inequality "$\leq$".
Another interesting generalizations of such optimization problems are related to objective function. Wu et al. [63] represented an efficient method to optimize a linear fractional programming problem under FRE with max-Archimedean t-norm composition. Dempe and Ruziyeva [4] 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 [7]. If the objective function is $z(x) = \max_{i=1}^{n}\{\min\{c_i, x_i\}\}$ with $c_i \in [0,1]$, the model is called the latticized problem [58]. Also, Yang et al. [66] introduced another version of the latticized programming problem subject to max-prod fuzzy relation inequalities with application in the optimization management model of wireless communication emission base stations. The latticized problem was defined by minimizing objective function $z(x) = x_1 \vee x_1 \vee ... \vee x_n$ subject to feasible region $X(A,b) = \{x \in [0,1]^n \;:\; A \circ x \geq b\}$ where ''$\circ$'' denotes fuzzy max-product composition. They also presented an algorithm based on the resolution of the feasible region. 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. Hassanzadeh et al. [23] used the same GA proposed by Lu and Fang to solve a similar nonlinear problem constrained by FRE and max-product operator.
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, and the finding of all the minimal solutions is an NP-hard problem. Secondly, a feasible region formed as FRE or FRI [15] is often a non-convex set. Finally, 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).
In this paper, we propose a genetic algorithm for solving problem (1), which keeps the search inside of the feasible region without finding any minimal solution and checking the feasibility of new generated solutions. For this purpose, the paper consists of three main parts. Firstly, we describe some structural details of FREs defined by the Dombi t-norm such as the theoretical properties of the solutions set, necessary and sufficient conditions for the feasibility of the problem, some simplification processes and the existence of an especial convex subset of the feasible region. By utilizing the convex subset, the proposed GA can easily generate a random feasible initial population. These results are used throughout the paper and provide a proper background to design an efficient GA by taking advantage of the structure of the feasible region. Then, our algorithm is presented based on the obtained theoretical properties. The proposed GA is designed especially for solving nonlinear optimization problems with fuzzy relational equations constraints. It is shown that all the operations used by the algorithm such as mutation and crossover are also kept within the feasible region. Finally, we provide some statistical and experimental results to evaluate the performance of our algorithm. Since the feasibility of problem (1) is essentially dependent on the t-norm (Dombi t-norm) used in the definition of the constraints, a method is also presented to construct feasible test problems. More precisely,

we construct a feasible problem by randomly generating a fuzzy matrix A and a fuzzy vector b according to some criteria resulted from the necessary and sufficient conditions. It is proved that the max-Dombi fuzzy relational equations constructed by this method is not empty. Moreover, a comparison is made between the proposed GA and the genetic algorithms presented in [23] and [38].
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). In section 3, the proposed GA and its characteristics are described, finally in section 4 the experimental results are demonstrated.

2. Basic properties of max-Dombi FRE

2.1. Characterization of feasible solutions set
This section describes the basic definitions and structural properties concerning problem (1) that are used throughout the paper. For the sake of simplicity, let $S_{T_D^\lambda}(a_i,b_i)$ denote the feasible solutions set of i'th equation, that is, $S_{T_D^\lambda}(a_i,b_i) = \left\{x \in [0,1]^n \ : \ \max_{j=1}^{n} \left\{T_D^\lambda(a_{ij},x_j)\right\} = b_i\right\}$. Also, let $S_{T_D^\lambda}(A,b)$ denote the feasible solutions set of problem (1). Based on the foregoing notations, it is clear that $S_{T_D^\lambda}(A,b) = \bigcap_{i\in I} S_{T_D^\lambda}(a_i,b_i)$.

Definition 1. For each $i \in I$, we define $J_i = \left\{j \in J \ : \ a_{ij} \geq b_i\right\}$.
According to definition 1, we have the following lemmas.

Lemma 1. Let $i \in I$. If $j \notin J_i$, then $T_D^\lambda(a_{ij},x_j) < b_i$, $\forall x_j \in [0,1]$.
Proof. From the monotonicity and identity law of t-norms, we have $T_D^\lambda(a_{ij},x_j) \leq T_D^\lambda(a_{ij},1) = a_{ij}$, $\forall x_j \in [0,1]$. Now, the result follows from the assumption (i.e., $j \notin J_i$) and definition 1.
Lemma 2. Suppose that $i \in I$ and $j \in J_i$. Also, let

$$V(b_i,a_{ij}) = \frac{1}{1+\left(\left(\frac{1-b_i}{b_i}\right)^\lambda - \left(\frac{1-a_{ij}}{a_{ij}}\right)^\lambda\right)^{1/\lambda}}.$$

(a) If $x_j > V(b_i,a_{ij})$ and $b_i \neq 0$, then $T_D^\lambda(a_{ij},x_j) > b_i$.
(b) If $x_j = V(b_i,a_{ij})$ and $b_i \neq 0$, then $T_D^\lambda(a_{ij},x_j) = b_i$.
(c) If $x_j < V(b_i,a_{ij})$ and $b_i \neq 0$, then $T_D^\lambda(a_{ij},x_j) < b_i$.
(d) If $a_{ij} = b_i = 0$, then $T_D^\lambda(a_{ij},x_j) = b_i$, $\forall x_j \in [0,1]$.
(e) If $a_{ij} > b_i = 0$, then $T_D^\lambda(a_{ij},x_j) = b_i$ for $x_j = 0$, and $T_D^\lambda(a_{ij},x_j) > b_i$ for $0 < x_j \leq 1$.

Proof. The proof is easily obtained from the definition of Dombi t-norm and definition 1. ?
Lemma 3 below gives a necessary and sufficient condition for the feasibility of sets $S_{T_D^\lambda}(a_i,b_i)$, $\forall i \in I$.
Lemma 3. For a fixed $i \in I$, $S_{T_D^\lambda}(a_i,b_i) \neq \emptyset$ if and only if $J_i \neq \emptyset$.
Proof. Suppose that $S_{T_D^\lambda}(a_i,b_i) \neq \emptyset$. So, there exists $x \in [0,1]^n$ such that $\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij},x_j)\right\} = b_i$. Therefore, we must have $T_D^\lambda(a_{ij_0},x_{j_0}) = b_i$ for some $j_0 \in J$. Now, lemma 1 implies $j_0 \in J_i$

that means $J_i \neq \emptyset$. Conversely, suppose that $J_i \neq \emptyset$ and let $j_0 \in J_i$. Now, if $b_i = 0$, then we set $\dot{x} = \mathbf{0}_{1\times n}$ where $\mathbf{0}_{1\times n}$ is an n-dimensional zero vector. Otherwise, if $b_i \neq 0$, we define $\dot{x} = [\dot{x}_1, \dot{x}_2, ..., \dot{x}_n] \in [0,1]^n$ where

$$\dot{x}_j = \begin{cases} V(b_i, a_{ij}) & j = j_0 \\ 0 & j \neq j_0 \end{cases}, \forall j \in J$$

and $V(b_i, a_{ij})$ is defined as in lemma 2. By this definition and lemma 2 (parts (b)-(e)), we have $T_D^\lambda(a_{ij_0}, \dot{x}_{j_0}) = b_i$ and $T_D^\lambda(a_{ij}, \dot{x}_j) = 0 \leq b_i$ for each $j \in J - \{j_0\}$. Therefore,

$$\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} = \max \left\{ T_D^\lambda(a_{ij_0}, \dot{x}_{j_0}), \max_{\substack{j \in J \\ j \neq j_0}} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} \right\} = T_D^\lambda(a_{ij_0}, \dot{x}_{j_0}) = b_i$$

The above equality shows that $\dot{x} \in S_{T_D^\lambda}(a_i, b_i)$. This completes the proof. ?

Definition 2. Suppose that $i \in I$ and $S_{T_D^\lambda}(a_i, b_i) \neq \emptyset$ (hence, $J_i \neq \emptyset$ from lemma 3). Let $\hat{x}_i = [(\hat{x}_i)_1, (\hat{x}_i)_2, ..., (\hat{x}_i)_n] \in [0,1]^n$ where the components are defined as follows:

$$(\hat{x}_i)_k = \begin{cases} V(b_i, a_{ik}) & k \in J_i,\ b_i \neq 0 \\ 0 & k \in J_i,\ a_{ik} > b_i = 0 \\ 1 & otherwise \end{cases}, \forall k \in J$$

and $V(b_i, a_{ij})$ is defined as in lemma 2. Also, for each $j \in J_i$, we define $\check{x}_i(j) = [\check{x}_i(j)_1, \check{x}_i(j)_2, ..., \check{x}_i(j)_n] \in [0,1]^n$ such that

$$\check{x}_i(j)_k = \begin{cases} V(b_i, a_{ik}) & b_i \neq 0 \ and \ k = j \\ 0 & otherwise \end{cases}, \forall k \in J$$

The following theorem characterizes the feasible region of the i'th relational equation ($i \in I$).
Theorem 1. Let $i \in I$. If $S_{T_D^\lambda}(a_i, b_i) \neq \emptyset$, then $S_{T_D^\lambda}(a_i, b_i) = \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i]$.
Proof. Firstly, we show that $\bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i] \subseteq S_{T_D^\lambda}(a_i, b_i)$. Then, we prove that $x \notin \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i]$ implies $x \notin S_{T_D^\lambda}(a_i, b_i)$. The second statement is equivalent to $S_{T_D^\lambda}(a_i, b_i) \subseteq \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i]$, and then the result follows. Let, $\dot{x} \in \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i]$. Thus, there exists some $j_0 \in J_i$ such that $\dot{x} \in [\check{x}_i(j_0), \hat{x}_i]$ (i.e., $\check{x}_i(j_0) \leq \dot{x} \leq \hat{x}_i$). In the first case, suppose that $b_i \neq 0$. So, definition 2 implies $\dot{x}_{j_0} = V(b_i, a_{ij_0})$, $\dot{x}_j = [0, V(b_i, a_{ij})]$, $\forall j \in J_i - \{j_0\}$, and $\dot{x}_j \in [0, 1]$, $\forall j \notin J_i$. Therefore, $T_D^\lambda(a_{ij}, \dot{x}_j) < b_i$, $\forall j \notin J_i$ (resulted from lemma 1), and then $\max_{j \notin J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} < b_i$. Also, $T_D^\lambda(a_{ij}, \dot{x}_j) \leq b_i$, $\forall j \in J_i - \{j_0\}$(resulted from lemma 2, parts (b) and (c)), which implies $\max_{j \in J_i - \{j_0\}} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} \leq b_i$. Additionally, $T_D^\lambda(a_{ij_0}, \dot{x}_{j_0}) = b_i$ from lemma 2 (part (b)). Hence, we have

$$\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} = \max \left\{ T_D^\lambda(a_{ij_0}, \dot{x}_{j_0}), \max_{j \in J_i - \{j_0\}} T_D^\lambda(a_{ij}, \dot{x}_j), \max_{j \notin J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} \right\} = b_i$$

Otherwise, suppose that $b_i = 0$. In this case, definition 2 implies $\dot{x}_j = 0$, $\forall j \in J_i$ such that $a_{ij} > 0$, and $\dot{x}_j \in [0, 1]$, otherwise. By similar arguments we have $\max_{j \notin J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} < b_i$. Also, $\max_{j \in J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} = b_i$ (resulted from lemma 2, parts (d) and (e)). Therefore, $\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} = \max \left\{\max_{j \in J_i} T_D^\lambda(a_{ij}, \dot{x}_j), \max_{j \notin J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\}\right\} = b_i$. Thus, for each case $\dot{x} \in S_{T_D^\lambda}(a_i, b_i)$ that implies $\bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i] \subseteq S_{T_D^\lambda}(a_i, b_i)$. Conversely, assume that $\dot{x} \notin \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i]$. Hence, either $\dot{x}$ is not less than $\hat{x}_i$ (i.e., $\dot{x} \nleq \hat{x}_i$) or $\dot{x}$ is not greater than $\check{x}_i(j)$, $\forall j \in J_i$ (i.e., $\dot{x} \ngeq \check{x}_i(j), \forall j \in J_i$). If $\dot{x} \nleq \hat{x}_i$, there must exists some $k \in J$ such that $\dot{x}_k > (\hat{x}_i)_k$. Then, from definition 2 we must have one of the three following cases: $\dot{x}_k > V(b_i, a_{ik})$ (if $k \in J_i$ and $b_i \neq 0$); $\dot{x}_k > 0$ (if $k \in J_i$ and $a_{ik} > b_i = 0$); $\dot{x}_k > 1$ (if $k \notin J_i$ or $a_{ik} = b_i = 0$). In the first and second cases, we have $T_D^\lambda(a_{ik}, \dot{x}_k) > b_i$ from lemma 2, parts (a) and (e), respectively. Therefore, $\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} > b_i$ that means $\dot{x} \notin S_{T_D^\lambda}(a_i, b_i)$. In the third case, the infeasibility of $\dot{x}$ is obvious. Otherwise, suppose that $\dot{x} \ngeq \check{x}_i(j), \forall j \in J_i$. Since each solution $\check{x}_i(j)$ $(j \in J_i)$ has at most one positive component $\check{x}_i(j)_j$ (from definition 2), we conclude $\dot{x}_j < \check{x}_i(j)_j$ $(\forall j \in J_i)$. So, for each $j \in J_i$ we have $\dot{x}_j < 0$, if $b_i = 0$, and $\dot{x}_j < V(b_i, a_{ij})$, if $b_i \neq 0$. In the former case, the result trivially follows. In the latter case, lemma 2 (part (c)) implies $T_D^\lambda(a_{ij}, \dot{x}_j) < b_i$ $(\forall j \in J_i)$. Therefore, $\max_{j \in J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} < b_i$, and then we have

$$\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\} = \max \left\{\max_{j \in J_i} T_D^\lambda(a_{ij}, \dot{x}_j), \max_{j \notin J_i} \left\{T_D^\lambda(a_{ij}, \dot{x}_j)\right\}\right\} < b_i$$

Thus, $\dot{x} \notin S_{T_D^\lambda}(a_i, b_i)$ that completes the proof. ?

From theorem 1, $\hat{x}_i$ is the unique maximum solution and $\check{x}_i(j)$'s $(j \in J_i)$ are the minimal solutions of $S_{T_D^\lambda}(a_i, b_i)$.

Definition 3. Let $\hat{x}_i$ $(i \in I)$ be the maximum solution of $S_{T_D^\lambda}(a_i, b_i)$. We define $\overline{X} = \min_{i \in I}\{\hat{x}_i\}$.
Definition 4. Let $e : I \to J_i$ so that $e(i) = j \in J_i$, $\forall i \in I$, and let E be the set of all vectors e. For the sake of convenience, we represent each $e \in E$ as an m–dimensional vector $e = [j_1, j_2, ..., j_m]$ in which $j_k = e(k)$.
Definition 5. Let $e = [j_1, j_2, ..., j_m] \in E$. We define $\underline{X}(e) = [\underline{X}(e)_1, \underline{X}(e)_2, ..., \underline{X}(e)_n] \in [0, 1]^n$, where $\underline{X}(e)_j = \max_{i \in I}\left\{\check{x}_i(e(i))_j\right\} = \max_{i \in I}\left\{\check{x}_i(j_i)_j\right\}$, $\forall j \in J$.
Theorem 2 below completely determines the feasible solutions set of problem (1).
Theorem 2. $S_{T_D^\lambda}(A, b) = \bigcup_{e \in E} [\underline{X}(e), \overline{X}]$.
Proof. Since $S_{T_D^\lambda}(A, b) = \bigcap_{i \in I} S_{T_D^\lambda}(a_i, b_i)$, from theorem 1 we have

$$S_{T_D^\lambda}(A, b) = \bigcap_{i \in I} \bigcup_{j \in J_i} [\check{x}_i(j), \hat{x}_i] = \bigcap_{i \in I} \bigcup_{e \in E} [\check{x}_i(e(i)), \hat{x}_i] = \bigcup_{e \in E} \bigcap_{i \in I} [\check{x}_i(e(i)), \hat{x}_i]$$
$$= \bigcup_{e \in E} [\max_{i \in I}\{\check{x}_i(e(i))\}, \min_{i \in I}\{\hat{x}_i\}] = \bigcup_{e \in E} [\underline{X}(e), \overline{X}]$$

where the last equality is obtained by definitions 3 and 5. ?

As a consequence, it turns out that $\overline{X}$ is the unique maximum solution and $\underline{X}(e)$'s $(e \in E)$ are the minimal solutions of $S_{T_D^\lambda}(A,b)$. Moreover, we have the following corollary that is directly resulted from theorem 2.

Corollary 1(first necessary and sufficient condition). $S_{T_D^\lambda}(A,b) \neq \emptyset$ if and only if $\overline{X} \in S_{T_D^\lambda}(A,b)$.

The following example illustrates the above-mentioned definitions.

Example 1. Consider the problem below with Dombi t-norm

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

Where

$$\varphi(x,y) = T_D^2(x,y) = \begin{cases} 0 & x = 0 \; or \; y = 0 \\ \frac{1}{1+\sqrt{\left(\frac{1-x}{x}\right)^2+\left(\frac{1-y}{y}\right)^2}} & otherwise \end{cases}$$

(i.e.,$\lambda = 2$). By definition 1, we have $J_1 = \{1\}$, $J_2 = \{1,5\}$, $J_3 = \{2,5\}$, $J_4 = \{1,4,5\}$ and $J_5 = \{1,2,3,4,5,6\}$. The unique maximum solution and the minimal solutions of each equation are obtained by definition 2 as follows:

$$\hat{x}_1 = [0.7073\,,\,1\,,\,1\,,\,1\,,\,1\,,\,1], \hat{x}_2 = [1\,,\,1\,,\,1\,,\,1\,,\,1\,,\,1], \hat{x}_3 = [1\,,\,0.6180\,,\,1\,,\,1\,,\,1\,,\,1],$$

$$\hat{x}_4 = [0.8170\,,\,1\,,\,1\,,\,1\,,\,1\,,\,1], \hat{x}_5 = [1\,,\,1\,,\,1\,,\,0\,,\,1\,,\,1].$$

$$\check{x}_1(1) = [0.7073\,,\,0\,,\,0\,,\,0\,,\,0\,,\,0],$$

$$\check{x}_2(1) = [1\,,\,0\,,\,0\,,\,0\,,\,0\,,\,0], \check{x}_2(5) = [0\,,\,0\,,\,0\,,\,0\,,\,1\,,\,0]$$

$$\check{x}_3(2) = [0\,,\,0.6180\,,\,0\,,\,0\,,\,0\,,\,0], \check{x}_3(5) = [0\,,\,0\,,\,0\,,\,0\,,\,1\,,\,0]$$

$$\check{x}_4(1) = [0.8170\,,\,0\,,\,0\,,\,0\,,\,0\,,\,0], \check{x}_4(4) = [0\,,\,0\,,\,0\,,\,1\,,\,0\,,\,0], \check{x}_4(5) = [0\,,\,0\,,\,0\,,\,0\,,\,1\,,\,0]$$

$$\check{x}_5(j) = [0\,,\,0\,,\,0\,,\,0\,,\,0\,,\,0]\,,\; j \in \{1,2,3,4,5,6\}$$

Therefore, by theorem 1 we have $S_{T_D^\lambda}(a_1,b_1) = [\check{x}_1(1)\,,\,\hat{x}_1]$, $S_{T_D^\lambda}(a_2,b_2) = [\check{x}_2(1)\,,\,\hat{x}_2] \bigcup [\check{x}_2(5)\,,\,\hat{x}_2]$, $S_{T_D^\lambda}(a_3,b_3) = [\check{x}_3(2)\,,\,\hat{x}_3] \bigcup [\check{x}_3(5)\,,\,\hat{x}_3]$ and$S_{T_D^\lambda}(a_4,b_4) = [\check{x}_4(1)\,,\,\hat{x}_4] \bigcup [\check{x}_4(4)\,,\,\hat{x}_4] \bigcup [\check{x}_4(5)\,,\,\hat{x}_4]$ and $S_{T_D^\lambda}(a_5,b_5) = [\mathbf{0}_{1\times 6}\,,\,\hat{x}_5]$ where $\mathbf{0}_{1\times 6}$ is a zero vector. From definition 3,
$\overline{X} = [0.7073\,,\,0.6180\,,\,1\,,\,0\,,\,1\,,\,1]$. It is easy to verify that $\overline{X} \in S_{T_D^\lambda}(A,b)$. Therefore, the above problem is feasible by corollary 1. Finally, the cardinality of set E is equal to 24 (definition 4). So, we have 24 solutions $\underline{X}(e)$ associated to 24 vectors e. For example, for $e = [1,5,5,5,5]$, we obtain $\underline{X}(e) = \max\{\check{x}_1(1), \check{x}_2(5), \check{x}_3(5), \check{x}_4(5), \check{x}_5(5)\}$ from definition 5 that means $\underline{X}(e) = [0.7073\,,\,0\,,\,0\,,\,0\,,\,1\,,\,0]$.

2.2. Simplification processes

In practice, there are often some components of matrix A that have no effect on the solutions to problem (1). Therefore, we can simplify the problem by changing the values of these components to zeros. For this reason, various simplification processes have been proposed by researchers. We refer the interesting reader to [15] where a brief review of such these processes is given. Here, we present two simplification techniques based on the Dombi t-norm.

Definition 6. If a value changing in an element, say a_{ij}, of a given fuzzy relation matrix A has no effect on the solutions of problem (1), this value changing is said to be an equivalence operation.

Corollary 2. Suppose that $T_D^\lambda(a_{ij_0}, x_{j_0}) < b_i, \forall x \in S_{T_D^\lambda}(A,b)$. In this case, it is obvious that $\max_{j=1}^{n} \left\{T_D^\lambda(a_{ij}, x_j)\right\} = b_i$ is equivalent to $\max_{\substack{j=1 \\ j \neq j_0}}^{n} \left\{T_D^\lambda(a_{ij}, x_j)\right\} = b_i$, that is, "resetting a_{ij_0} to zero" has no effect on the solutions of problem (1) (since component a_{ij_0} only appears in the i'th constraint of problem (1)). Therefore, if $T_D^\lambda(a_{ij_0}, x_{j_0}) < b_i, \forall x \in S_{T_D^\lambda}(A,b)$, then "resetting a_{ij_0} to zero" is an equivalence operation.

Lemma 4 (first simplification). Suppose that $j_0 \notin J_i$, for some $i \in I$ and $j_0 \in J$. Then, "resetting a_{ij_0} to zero" is an equivalence operation.

Proof. From corollary 2, it is sufficient to show that $T_D^\lambda(a_{ij_0}, x_{j_0}) < b_i, \forall x \in S_{T_D^\lambda}(A,b)$. But, from lemma 1 we have $T_D^\lambda(a_{ij_0}, x_{j_0}) < b_i$, $\forall x_{j_0} \in [0,1]$. Thus, $T_D^\lambda(a_{ij_0}, x_{j_0}) < b_i, \forall x \in S_{T_D^\lambda}(A,b)$. ?

Lemma 5 (second simplification). Suppose that $j_0 \in J_{i_1}$ and $b_{i_1} \neq 0$, where $i_1 \in I$ and $j_0 \in J$. If at least one of the following conditions hold, then "resetting $a_{i_1 j_0}$ to zero" is an equivalence operation:

(a) There exists some $i_2 \in I(i_1 \neq i_2)$ such that $j_0 \in J_{i_2}$, $b_{i_2} \neq 0$ and $V(b_{i_2}, a_{i_2 j_0}) < V(b_{i_1}, a_{i_1 j_0})$, where $V(b_i, a_{ij})$ is defined as in lemma 2.

(b) There exists some $i_2 \in I(i_1 \neq i_2)$ such that $b_{i_2} = 0$ and $a_{i_2 j_0} > 0$.

Proof. (a) Similar to the proof of lemma 4, we show that $T_D^\lambda(a_{i_1 j_0}, x_{j_0}) < b_i, \forall x \in S_{T_D^\lambda}(A,b)$. Consider an arbitrary feasible solution $x \in S_{T_D^\lambda}(A,b)$. Since $x \in S_{T_D^\lambda}(A,b)$, it turns out that $T_D^\lambda(a_{i_1 j_0}, x_{j_0}) > b_{i_1}$ never holds. So, assume that $T_D^\lambda(a_{i_1 j_0}, x_{j_0}) = b_{i_1}$. Since $b_{i_1} \neq 0$, from lemma 2 we conclude that $x_{j_0} = V(b_{i_1}, a_{i_1 j_0})$. So, by the assumption, we have $V(b_{i_2}, a_{i_2 j_0}) < x_{j_0}$. Therefore, lemma 2 (part (a)) implies $T_D^\lambda(a_{i_2 j_0}, x_{j_0}) > b_{i_2}$ that contradicts $x \in S_{T_D^\lambda}(A,b)$.

(b) By the assumption, we have $j_0 \in J_{i_2}$. Now, the result similarly follows by a simpler argument. ?

We give an example to illustrate the above two simplification processes.

Example 2. Consider the problem presented in example 1. From the first simplification (lemma 4), "resetting the following components a_{ij} to zeros" are equivalence operations: a_{12},a_{13},a_{14},a_{15},a_{16}; a_{22},a_{23},a_{24},a_{26}; a_{31},a_{33},a_{34},a_{36}; a_{42},a_{43},a_{46}; in all of these cases, $a_{ij} < b_i$, that is, $j \notin J_i$. Also, from the second simplification (lemma 5, part (a)), we can change the values of components a_{21}

and $a_{41}$to zeros. For example, $a_{21} = b_2$ (i.e., $1 \in J_2$), $b_2 \neq 0$, $a_{11} > b_1$ (i.e., $1 \in J_1$), $b_1 \neq 0$ and $0.7073 = V(b_1, a_{11}) < V(b_2, a_{21}) = 1$.
Moreover, from lemma 5 (part (b)), we can also change the value of component a_{44} to zero with no effect on the solutions set of the problem (since $4 \in J_4$, $b_4 \neq 0$, $b_5 = 0$ and $a_{54} > 0$).
In addition to simplifying the problem, a necessary and sufficient condition is also derived from lemma 5. Before formally presenting the condition, some useful notations are introduced. Let $\tilde{A}$ denote the simplified matrix resulted from A after applying the simplification processes (lemmas 4 and 5). Also, similar to definition 1, assume that $\tilde{J}_i = \left\{j \in J : \tilde{a}_{ij} \geq b_i\right\}$ $(i \in I)$ where $\tilde{a}_{ij}$ denotes (i, j)'th component of matrix $\tilde{A}$. The following theorem gives a necessary and sufficient condition for the feasibility of problem (1).
Theorem 3 (second necessary and sufficient condition). $S_{T_D^\lambda}(A,b) \neq \emptyset$ if and only if $\tilde{J}_i \neq \emptyset$, $\forall i \in I$.
Proof. Since $S_{T_D^\lambda}(A,b) = S_{T_D^\lambda}(\tilde{A},b)$ from lemmas 4 and 5, it is sufficient to show that $S_{T_D^\lambda}(\tilde{A},b) \neq \emptyset$ if and only if $\tilde{J}_i \neq \emptyset$, $\forall i \in I$. Let $S_{T_D^\lambda}(\tilde{A},b) \neq \emptyset$. Therefore, $S_{T_D^\lambda}(\tilde{a}_i,b_i) \neq \emptyset$, $\forall i \in I$, where $\tilde{a}_i$ denotes i'th row of matrix $\tilde{A}$. Now, lemma 3 implies $\tilde{J}_i \neq \emptyset$, $\forall i \in I$. Conversely, suppose that $\tilde{J}_i \neq \emptyset$, $\forall i \in I$. Again, by using lemma 3 we have$\tilde{J}_i \neq \emptyset$, $\forall i \in I$. By contradiction, suppose that $S_{T_D^\lambda}(\tilde{A},b) = \emptyset$. Therefore, $\overline{X} \notin S_{T_D^\lambda}(\tilde{A},b)$ from corollary 1, and then there exists $i_0 \in I$ such that $\overline{X} \notin S_{T_D^\lambda}(\tilde{a}_{i_0},b_{i_0})$. Since $\max_{j \notin \tilde{J}_i}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} < b_{i_0}$ (from lemma 1), we must have either $\max_{j \in \tilde{J}_i}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} > b_{i_0}$ or $\max_{j \in \tilde{J}_i}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} < b_{i_0}$. Anyway, since $\overline{X} \leq \hat{x}_{i_0}$ (i.e., $\overline{X}_j \leq (\hat{x}_{i_0})_j$, $\forall j \in J$), we have $\max_{j \in \tilde{J}_{i_0}}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} \leq \max_{j \in \tilde{J}_{i_0}}\left\{T_D^\lambda(\tilde{a}_{i_0 j},(\hat{x}_{i_0})_j)\right\} = b_{i_0}$, and then the former case (i.e., $\max_{j \in \tilde{J}_i}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} > b_{i_0}$) never holds. Therefore, $\max_{j \in \tilde{J}_i}\left\{T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j)\right\} < b_{i_0}$ that implies $b_{i_0} \neq 0$ and $T_D^\lambda(\tilde{a}_{i_0 j},\overline{X}_j) < b_{i_0}$, $\forall j \in \tilde{J}_{i_0}$. Hence, by lemma 2, we must have $\overline{X}_j < V(b_{i_0},\tilde{a}_{i_0 j})$, $\forall j \in \tilde{J}_{i_0}$. On the other hand, $V(b_{i_0},\tilde{a}_{i_0 j}) \leq 1$, $\forall j \in \tilde{J}_{i_0}$. Therefore, $\overline{X}_j < 1$, $\forall j \in \tilde{J}_{i_0}$, and then from definitions 2 and 3, for each $j \in \tilde{J}_{i_0}$there must exists $i_j \in I$ such that either $j \in \tilde{J}_{i_j}$ and $\overline{X}_j = (\hat{x}_{i_j})_j = V(b_{i_j},\tilde{a}_{i_j j})$ or $j \in \tilde{J}_{i_j}$and $\tilde{a}_{i_j j} > b_{i_j} = 0$. Until now, we proved that $b_{i_0} \neq 0$ and for each $j \in \tilde{J}_{i_0}$, there exist $i_j \in I$ such that either $j \in \tilde{J}_{i_j}$ and $V(b_{i_j},\tilde{a}_{i_j j}) < V(b_{i_0},\tilde{a}_{i_0 j})$ (because, $V(b_{i_j},\tilde{a}_{i_j j}) = \overline{X}_j < V(b_{i_0},\tilde{a}_{i_0 j})$) or $b_{i_j} = 0$ and $\tilde{a}_{i_j j} > 0$. But in both cases, we must have $\tilde{a}_{i_0 j} = 0$ ($\forall j \in \tilde{J}_{i_0}$) from the parts (a) and (b) of lemma 5, respectively. Therefore, $\tilde{a}_{i_0 j} < b_{i_0} \neq 0$ ($\forall j \in \tilde{J}_{i_0}$) that is a contradiction. ?

Remark 1. Since $S_{T_D^\lambda}(A,b) = S_{T_D^\lambda}(\tilde{A},b)$ (from lemmas 4 and 5), we can rewrite all the previous definitions and results in a simpler manner by replacing $\tilde{J}_i$ with $J_i (i \in I)$.

3. The proposed GA

Genetic algorithms (GAs) are metaheuristics inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). In a genetic algorithm, a population of solutions (called individuals) to an optimization problem is iteratively evolved toward better solutions (the population in each iteration called a generation). The evolution usually starts from a population of randomly generated individuals and progress to improve solutions by emulating some bio-inspired operators such as mutation, crossover and selection. In each generation, the fitness (performance) of every individual in the population is evaluated, and based on the performance, the relatively good solutions are retained and the relatively bad solutions are replaced with some newly generated offsprings. The fitness is usually the value of the objective function in the optimization problem being solved. The new generation of solutions is then used in the next iteration of the algorithm.

In this section, a genetic algorithm is presented for solving problem (1). Since the feasible region of problem (1) is non-convex, a convex subset of the feasible region is firstly introduced. Consequently, the proposed GA can easily generate the initial population by randomly choosing individuals from this convex feasible subset. The mutation and crossover operators are also designed to keep the feasibility of the individuals without checking the feasibility of the new generated solutions. Solutions with better objective values will have higher opportunities to survive and the algorithm terminates after taking a pre-determined number of generations. At the last part of this section, a method is presented to generate random feasible max-Dombi fuzzy relational equations.

3.1. Representation

Similar to the mentioned related literatures [23,38], we use the floating-point representation in which each variable (gene) x_j in a solution (individual) $x = [x_1, x_2, ..., x_n]$ belongs to the interval $[0,1]$. There are several reasons for using the floating-point representation instead of binary strings. For example, all components of every solution in problem (1) are nonnegative numbers that are less than or equal to one. Also, the floating-point representation is faster, more consistence, and provides high precision [38].

3.2. Initialization

As mentioned before, GAs 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 the proposed GA, the initial population is given by randomly generating the individuals inside the feasible region. For this purpose, we firstly find a convex subset of the feasible solutions set, that is, we find set F such that $F \subseteq S_{T_D^\lambda}(A,b)$ and F is convex. Then, the initial population is generated by randomly selecting individuals from set F.

Definition 7. Suppose that $S_{T_D^\lambda}(\tilde{A},b) \neq \emptyset$. For each $i \in I$, let $\check{x}_i = [(\check{x}_i)_1, (\check{x}_i)_2, ..., (\check{x}_i)_n] \in [0,1]^n$ where the components are defined as follows:

$$(\check{x}_i)_k = \begin{cases} V(b_i, a_{ik}) & b_i \neq 0 \; and \; k \in \tilde{J}_i \\ 0 & otherwise \end{cases} , \forall k \in J$$

Also, we define $\underline{X} = \max_{i \in I} \{\check{x}_i\}$.

Remark 2. According to definition 2 and remark 1, it is clear that for a fixed $i \in I$ and $j \in \tilde{J}_i$, $\check{x}_i(j)_k \leq (\check{x}_i)_k$ ($\forall k \in J$). Therefore, from definitions 5 and 7 we have $\underline{X}(e)_k = \max_{i \in I} \{\check{x}_i(e(i))_k\} = \max_{i \in I} \{\check{x}_i(j_i)_k\} \leq \max_{i \in I} \{(\check{x}_i)_k\} = \underline{X}_k$, $\forall k \in J$ and $\forall e \in E$. Thus, $\underline{X}(e) \leq \underline{X}$, $\forall e \in E$.

Lemma 6 (a Convex subset of the feasible region). Suppose that $S_{T_D^\lambda}(\tilde{A},b) \neq \emptyset$ and $F = \left\{x \in [0,1]^n \; : \; \underline{X} \leq x \leq \overline{X}\right\}$. Then, $F \subseteq S_{T_D^\lambda}(\tilde{A},b)$ and F is a convex set.

Proof. From theorem 2, we have $S_{T_D^\lambda}(\tilde{A},b) = S_{T_D^\lambda}(A,b) = \bigcup_{e\in E}[\underline{X}(e), \overline{X}]$. To prove the lemma, we show that $\underline{X}(e) \le \underline{X} \le \overline{X}$, $\forall e \in E$. Then, we can conclude $[\underline{X}, \overline{X}] \subseteq [\underline{X}(e), \overline{X}]$, $\forall e \in E$, that implies both $F \subseteq S_{T_D^\lambda}(A,b)$ and the convexity of F. But from remark 2, $\underline{X}(e) \le \underline{X}$, $\forall e \in E$. Therefore, it is sufficient to prove $\underline{X} \le \overline{X}$. By contradiction, suppose that $\underline{X}_{j_0} > \overline{X}_{j_0}$ for some $j_0 \in J$. So, from definitions 2, 3 and 7, there must exist $i_1 \in I$ and $i_2 \in I$ such that either $\underline{X}_{j_0} = (\check{x}_{i_1})_{j_0} = V(b_{i_1}, \tilde{a}_{i_1 j_0})$, $\overline{X}_{j_0} = (\hat{x}_{i_2})_{j_0} = V(b_{i_2}, \tilde{a}_{i_2 j_0})$ and $\overline{X}_{j_0} < \underline{X}_{j_0}$, i.e.,

$$V(b_{i_2}, \tilde{a}_{i_2 j_0}) < V(b_{i_1}, \tilde{a}_{i_1 j_0})(*)$$

or $\underline{X}_{j_0} = (\check{x}_{i_1})_{j_0} = V(b_{i_1}, \tilde{a}_{i_1 j_0})$, $\overline{X}_{j_0} = (\hat{x}_{i_2})_{j_0} = 0$ and $\underline{X}_{j_0} > 0$. But the former case occurs only when $b_{i_1} \ne 0$ and $j_0 \in \tilde{J}_{i_1} \bigcap \tilde{J}_{i_2}$. These facts together with (*) imply $\tilde{a}_{i_1 j_0} = 0$ from lemma 5, part (a). Also, the latter case occurs only when $b_{i_1} \ne 0$, $b_{i_2} = 0$ and $\tilde{a}_{i_2 j_0} > 0$. However, in this case, part (b) of lemma 5 again gives the same result, i.e., $\tilde{a}_{i_1 j_0} = 0$. Therefore, in all the cases we have $\tilde{a}_{i_1 j_0} < b_{i_1}$ that contradicts $j_0 \in \tilde{J}_{i_1}$. ?
To illustrate definition 7 and lemma 6, we give the following example.
Example 3. Consider the problem presented in example 1, where
$\overline{X} = [0.7073\ ,\ 0.6180\ ,\ 1\ ,\ 0\ ,\ 1\ ,\ 1]$. Also, according to example 2, the simplified matrix $\tilde{A}$ is

$$\tilde{A} = \begin{bmatrix} 0.9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0.8 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0 \end{bmatrix}$$

From definition 7, we have

$$\check{x}_1 = [0.7073\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0], \check{x}_2 = [0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0], \check{x}_3 = [0\ ,\ 0.6180\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0],$$

$$\check{x}_4 = [0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0], \check{x}_5 = [0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0\ ,\ 0]$$

, and then $\underline{X} = \max_{i=1}^{5}\{\check{x}_i\} = [0.7073\ ,\ 0.6180\ ,\ 0\ ,\ 0\ ,\ 1\ ,\ 0]$. Therefore, set $F = [\underline{X}, \overline{X}]$ is obtained as a collection of intervals:

$$F = [\underline{X}, \overline{X}] = [0.7073\ ,\ 0.6180\ ,\ [0,1]\ ,\ 0\ ,\ 1\ ,\ [0,1]]$$

By generating random numbers in the corresponding intervals, we acquire one initial individual: $x = [0.7073\ ,\ 0.6180\ ,\ 0.001\ ,\ 0\ ,\ 1\ ,\ 0.88]$.
According to lemma 6, the algorithm for generating the initial population is simply obtained as follows:
Algorithm 1 (Initial Population).

1. Get fuzzy matrix A, fuzzy vector b and population size S_{pop}.
2. If $\overline{X} \notin S_{T_D^\lambda}(A,b)$, then stop; the problem is infeasible (corollary1).
3. For i = 1, 2, ..., S_{pop}
   Generate a random n – dimensional solution $pop(i)$ in the interval $[\underline{X}, \overline{X}]$.
   End

3.3. Selection strategy

Suppose that the individuals in the population are sorted according to their ranks from the best to worst, that is, individual $pop(r)$ has rank r. Therefore, the first individual is the best one with the smallest objective value in problem (1). The weight of the individual $pop(r)$ is calculated by the following formula:

$$W_r = \frac{1}{\sqrt{2\pi} q S_{pop}} e^{-\frac{1}{2}\left(\frac{r-1}{q S_{pop}}\right)^2} \tag{2}$$

which essentially defines the weight to be a value of the Gaussian function with argument r, mean 1, and standard deviation qS_{pop}, where q is a parameter of the algorithm. When q is small, the best-ranked individuals are strongly preferred, and when it is large, the probability becomes more uniform. Based on relation (2), the probability P_r of choosing the r'th individual is given by:

$$P_r = \frac{W_r}{\sum_{k=1}^{S_{pop}} W_k} \tag{3}$$

3.4. Mutation operator

Although various mutation operators have been proposed for handling the constrained optimization problems, there seldom is any mutation operator available for the non-convex problem [38]. In this section, a mutation operator is presented, which preserves the feasibility of new individuals in the non-convex feasible domain. As usual, suppose that $S_{T_D^\lambda}(A, b) \neq \emptyset$. So, from theorem 3 we have $\tilde{J}_i \neq \emptyset$, $\forall i \in I$, where $\tilde{J}_i = \left\{ j \in J \ : \ \tilde{a}_{ij} \geq b_i \right\}$, $\forall i \in I$(see definition1 and remark 1).

Definition 8. Let $I^+ = \{i \in I \ : \ b_i \neq 0\}$. So, we define $D = \left\{ j \in J \ : \ \text{if } \exists i \in I^+ \text{ such that } j \in \tilde{J}_i \Rightarrow \left| \tilde{J}_i \right| > 1 \right\}$, where $\left|\tilde{J}_i\right|$ denotes the cardinality of set $\tilde{J}_i$. For a given individual $x = [x_1, x_2, ..., x_n]$, we define an operator that mutates the individual by randomly choosing an element $j_0 \in D$ and decreasing x_{j_0} from its current value to zero. Therefore, for the new individual $x' = [x'_1, x'_2, ..., x'_n]$ we have $x'_{j_0} = 0$, and $x'_j = x_j$, $\forall j \in J - \{j_0\}$. If x' is infeasible, the mutation operator will neglect this decreasing operation and find another x_j ($j \in D$ and $j \neq j_0$) to decrease.

Remark 3. Suppose that $i_0 \in I^+$, $j_0 \in \tilde{J}_{i_0}$ and $\left| \tilde{J}_{i_0} \right| = 1$. Therefore, $\tilde{J}_{i_0} = \{j_0\}$ and according to definition 8 we have $j_0 \notin D$. So, if we decide to set $x_{j_0} = 0$, then from lemma 1

$$\max_{k=1}^{n} \left\{ T_D^{\lambda}(\tilde{a}_{i_0 k}, x_k) \right\} = \max \left\{ \max_{j \in \tilde{J}_{i_0}} \left\{ T_D^{\lambda}(\tilde{a}_{i_0 k}, x_k) \right\}, \max_{j \notin \tilde{J}_{i_0}} \left\{ T_D^{\lambda}(\tilde{a}_{i_0 k}, x_k) \right\} \right\}$$

$$= \max \left\{ T_D^{\lambda}(\tilde{a}_{i_0 j_0}, x_{j_0}), \max_{\substack{k=1 \\ k \neq j_0}}^{n} \left\{ T_D^{\lambda}(\tilde{a}_{i_0 k}, x_k) \right\} \right\}$$

$$= \max \left\{ T_D^{\lambda}(\tilde{a}_{i_0 j_0}, 0), \max_{\substack{k=1 \\ k \neq j_0}}^{n} \left\{ T_D^{\lambda}(\tilde{a}_{i_0 k}, x_k) \right\} \right\} < b_{i_0}$$

In this case, the new individual violates i_0'th equation. This is the reason why the reduction process only selects those elements j belonging to the set D.
Based on definition 8 and remark 3, we present the mutation operator as follows:
Algorithm 2 (Mutation operator).

1. Get the matrix $\tilde{A}$, vector b and a selected solution $\dot{x} = [\dot{x}_1, ..., \dot{x}_n]$.
2. While $D \neq \emptyset$
   2.1. Set $x' \leftarrow x$.
   2.2. Randomly choose $j_0 \in D$, and set $x'_{j_0} = 0$.
   2.3. IF x' is feasible, go to Crossover operator; otherwise, set $D = D - \{j_0\}$.

Remark 4. From theorem 2, if $x \in S_{T_D^{\lambda}}(A,b)$, then there exists some $e \in E$ such that $x \in [\underline{X}(e), \overline{X}]$. Therefore, if $x \neq \underline{X}(e)$, it is always possible for algorithm 2 to find an element $j_0 \in D$ and generate a feasible solution x' by setting $x_{j_0} = 0$. The only exceptions are the minimal solutions. The minimal solutions are actually the lower bounds of the feasible region, and therefore any reduction in their variables results in an infeasible point. Hence, if the While-loop of the above algorithm is terminated with $D = \emptyset$, it turns out that $\dot{x}$ must be a minimal solution.
3.5. Crossover operator
In section 2, it was proved that $\overline{X}$ is the unique maximum solution of $S_{T_D^{\lambda}}(A,b)$. By using this result, the crossover operator is stated as follows:
Algorithm 3 (Crossover operator).

1. Get the maximum solution $\overline{X}$, the new solution x' (generated by algorith 2) and one parent $pop(k)$ (for some $k = 1, 2, ..., S_{pop}$).
2. Generate a random number $\lambda_1 \in [0,1]$. Set $x_{new1} = \lambda_1 x' + (1 - \lambda_1)\overline{X}$.
3. Let $\lambda_2 = \min_{\substack{j=1 \\ j \neq k}}^{S_{pop}} \|pop(k) - pop(j)\|$ and $d = \overline{X} - pop(k)$.
   Set $x_{new2} = pop(k) + \min\{\lambda_2, 1\}\, d$.

Remark 6. From the above algorithm, the new individual x_{new1} is generated by the convex combination of x' and $\overline{X}$. Since $x' \in S_{T_D^\lambda}(A,b)$, theorem 2 implies $x' \in [\underline{X}(e), \overline{X}]$, for some $e \in E$. Thus, since $[\underline{X}(e), \overline{X}]$ is a closed cell, the generated offsprings x_{new1} is always feasible, and therefore we have no need to check its feasibility. Similar argument is also true for x_{new2}. The only difference is that the offspring x_{new2} usually locates close to its parent (i.e., $pop(k)$). It is because of the step length λ_2 computed as the minimum distances between the parent $pop(k)$ and other individual. This strategy increases the ability of the algorithm to find the optima around a good solution.

3.6. Construction of test problems

There are usually several ways to generate a feasible FRE defined with different t-norms. In what follows, we present a procedure to generate random feasible max- Dombi fuzzy relational equations:

Algorithm 4 (construction of feasible Max-Dombi FRE).

1. Randomly select m columns $\{j_1, j_2, ..., j_m\}$ from $J = \{1,2,...,n\}$.
2. Generate vector b whose elements are random numbers from $[0,1]$.
3. For $i \in \{1,2,...,m\}$
   Assign a random number from $[b_i, 1]$ to a_{ij_i}.
   End
4. For $i \in \{1,2,...,m\}$
   For each $k \in \{1,2,...,m\} - \{i\}$
   If $b_k = 0$
   Set $a_{kj_i} = 0$.
   Else, generate a random number θ from $[0,1]$.
   If $\theta \le 0.5$, assign a random number from $[0, b_k)$ to a_{kj_i}.
   Else
   If $(\frac{1-b_k}{b_k})^\lambda < (\frac{1-b_i}{b_i})^\lambda - (\frac{1-a_{ij_i}}{a_{ij_i}})^\lambda$, assign a random number from $[b_k, 1]$ to a_{kj_i}.
   Else
   Set $L = \frac{1}{1+\left(\left(\frac{1-b_k}{b_k}\right)^\lambda - \left(\frac{1-b_i}{b_i}\right)^\lambda + \left(\frac{1-a_{ij_i}}{a_{ij_i}}\right)^\lambda\right)^{1/\lambda}}$.
   Assign a random number from $[0\ , L]$ to a_{kj_i}.
   End
5. For each $i \in \{1,2,...,m\}$ and each $j \notin \{j_1, j_2, ..., j_m\}$
   Assign a random number from $[0,1]$ to a_{ij}.
   End

From step 4 of the above algorithm, we note that if $\theta \le 0.5$, then we will have $a_{kj_i} \in [0, b_k)$, and therefore $j_i \notin J_k$. Also, if $\theta > 0.5$ and $(\frac{1-b_k}{b_k})^\lambda < (\frac{1-b_i}{b_i})^\lambda - (\frac{1-a_{ij_i}}{a_{ij_i}})^\lambda$, then $a_{kj_i} \in [b_k, 1]$. In this case, after applying the algorithm we will have $V(b_k, a_{kj_i}) > V(b_i, a_{ij_i})$. By the following theorem, it is proved that algorithm 4 always generates random feasible max-Dombi fuzzy relational equations.

Theorem 4. The solutions set $S_{T_D^\lambda}(A,b)$ of FRE (with Dombi t-norm) constructed by algorithm

4 is not empty.

Proof. According to step 3 of the algorithm, $j_i \in J_i$, $\forall i \in I$. Therefore, $J_i \neq \emptyset$, $\forall i \in I$. To complete the proof, we show that $j_i \in \tilde{J}_i$, $\forall i \in I$. By contradiction, suppose that the second simplification process reset a_{ij_i} to zero, for some $i \in I$. So, $b_i \neq 0$ and there must exists some $k \in I(k \neq i)$ such that either $j_i \in J_k$, $b_k \neq 0$ and $V(b_k, a_{kj_i}) < V(b_i, a_{ij_i})$ or $b_k = 0$ and $a_{kj_i} > 0$. In the former case, the inequality $V(b_k, a_{kj_i}) < V(b_i, a_{ij_i})$ implies

$$\left(\left(\frac{1-b_i}{b_i}\right)^{\lambda} - \left(\frac{1-a_{ij_i}}{a_{ij_i}}\right)^{\lambda}\right)^{1/\lambda} < \left(\left(\frac{1-b_k}{b_k}\right)^{\lambda} - \left(\frac{1-a_{kj_i}}{a_{kj_i}}\right)^{\lambda}\right)^{1/\lambda}$$

Therefore, we must have $(\frac{1-b_i}{b_i})^{\lambda} - (\frac{1-a_{ij_i}}{a_{ij_i}})^{\lambda} < (\frac{1-b_k}{b_k})^{\lambda}$, and then $a_{kj_i} > L$ in which

$$L = \frac{1}{1 + \left(\left(\frac{1-b_k}{b_k}\right)^{\lambda} - \left(\frac{1-b_i}{b_i}\right)^{\lambda} + \left(\frac{1-a_{ij_i}}{a_{ij_i}}\right)^{\lambda}\right)^{1/\lambda}}.$$

Anyway, both cases contradict step 4. ?

4. Experimental results and comparison with related works

In this section, we present the experimental results for evaluating the performance of our algorithm. Firstly, we apply our algorithm to 8 test problems described in Appendix A. The test problems have been randomly generated in different sizes by algorithm 4 given in section 3. Since the objective function is an ordinary nonlinear function, we take some objective functions from the well-known source: Test Examples for Nonlinear Programming Codes [24]. In section 5.2, we make a comparison against the related GAs proposed in [23] and [38]. 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 by the authors in [23] and [38]. Since the authors did not explicitly reported the size of the population, we consider $S_{pop} = 50$ for all the three GAs. As mentioned before, we set $q = 0.1$ in relation (2) for the current GA. Moreover, in order to compare our algorithm with max-min GA [38] (max-product GA [23]), we modified all the definitions used in the current GA based on the minimum t-norm (product t-norm). For example, we used the simplification process presented in [38] for minimum, and the simplification process given in [13, 23] for product. Finally, 30 experiments are performed for all the GAs and for eight test problems reported in Appendix B, that is, each of the preceding GA is executed 30 times for each test problem. All the test problems included in Appendix A, have been defined by considering $\lambda = 2$ in T_D^{λ}. Also, the maximum number of iterations is equal to 100 for all the methods.

5.1. Performance of the max-Dombi GA

To verify the solutions found by the max-Dombi GA, the optimal solutions of the test problems are also needed. Since $S_{T_D^{\lambda}}(A, b)$ is formed as the union of the finite number of convex closed cells (theorem 2), the optimal solutions are also acquired by the following procedure:

1. Computing all the convex cells of the Dombi FRE.
2. Searching the optimal solution for each convex cell.

3. Finding the global optimum by comparing these local optimal solutions.
The computational results of the eight test problems (see Appendix A) are shown in Table 1 and Figures 1-8. In Table 1, the results are averaged over 30 runs and the average best-so-far solution, average mean fitness function and median of the best solution in the last iteration are reported.
Table 2 includes the best results found by the max-Dombi GA and the above procedure. According to Table 2, the optimal solutions computed by the max- Dombi GA and the optimal solutions obtained by the above procedure match very well. Tables 1 and 2, demonstrate the attractive ability of the max-Dombi GA to detect the optimal solutions of problem (1). Also, the good convergence rate of the max-Dombi GA could be concluded from Table 1 and figures 1-8.

Table 1. Results of applying the max-Dombi GA to the eight test problems of Appendix A. The results have been averaged over 30 runs. Maximum number of iterations=100.

Test problems	Average best-so-far	Median best-so-far	Average mean fitness
A.1	15.699631	15.699631	15.770439
A.2	0.105175	0.105175	0.105181
A.3	-0.946994	-0.946994	-0.946987
A.4	4.462970	4.462970	4.462995
A.5	118.169452	118.169454	118.175547
A.6	-0.176134	-0.176134	-0.175935
A.7	0.370392	0.370392	0.371217
A.8	58.57292	58.57291	58.57313

Table 2. Comparison of the solutions found by Max-Dombi GA
and the optimal values of the test problems described in Appendix A.

Test problems	Solutions of max-Dombi GA	Optimal values
A.1	15.699631 0.105175	15.699631
A.2	-0.946994	0.105170
A.3	4.462970	-0.946995
A.4	118.169437	4.462970
A.5	-0.176134	118.169437
A.6	0.370392	-0.176134
A.7	58.5729	0.370392 58.5726
A.8		

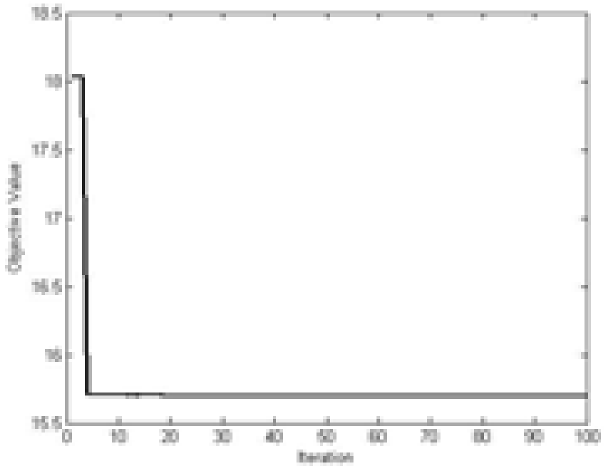

Figure 1: The performance of the max-Dombi GA on test problem A.2.

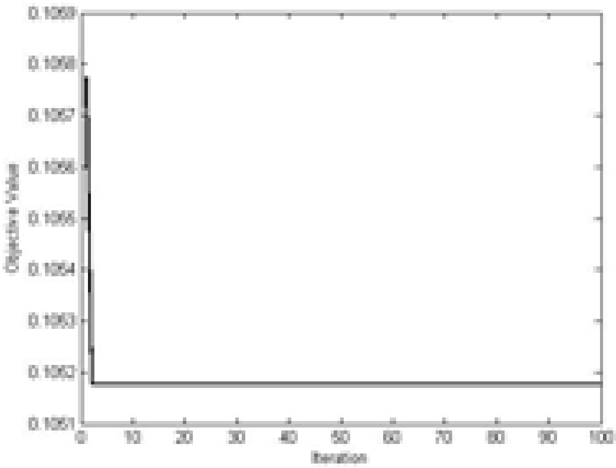

Figure 2: The performance of the max-Dombi GA on test problem A.2.

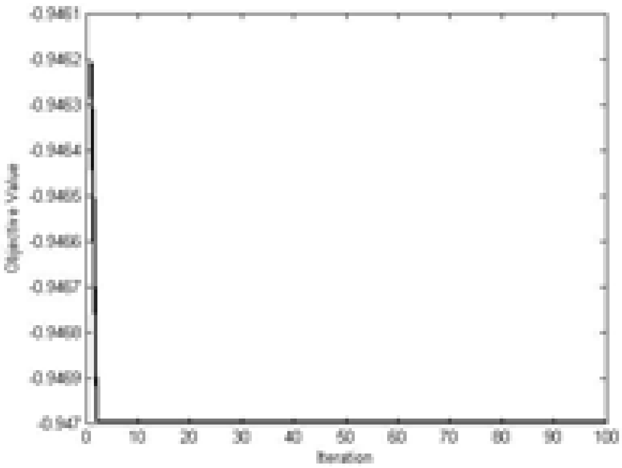

Figure 3: The performance of the max-Dombi GA on test problem A.3.

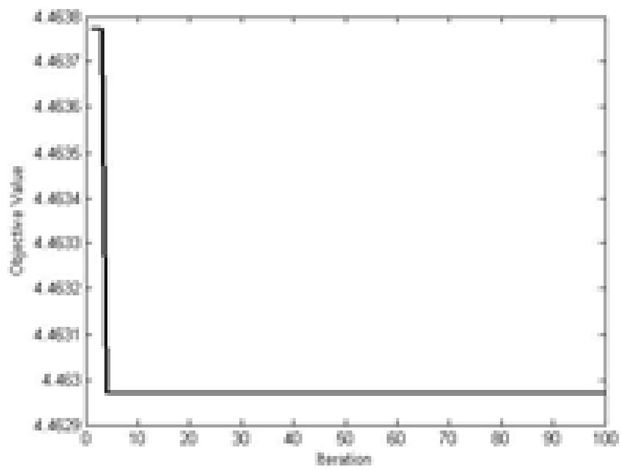

Figure 4: The performance of the max-Dombi GA on test problem A.4.

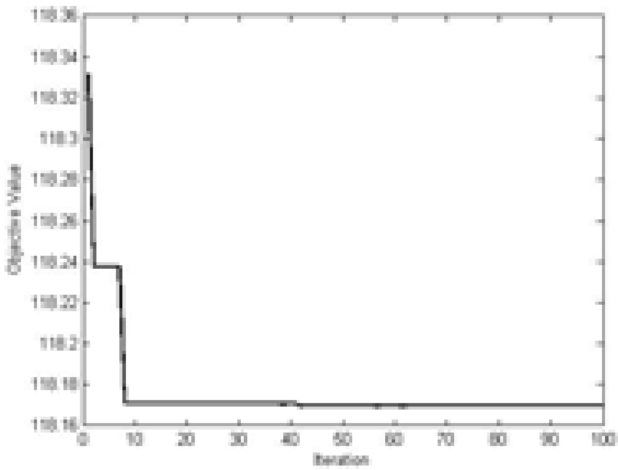

Figure 5: The performance of the max-Dombi GA on test problem A.5.

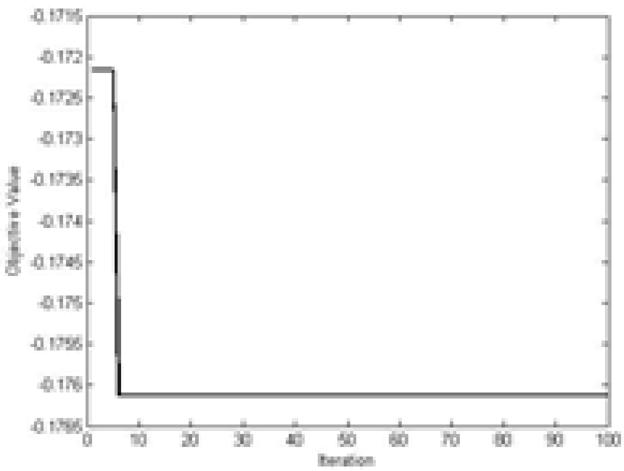

Figure 6: The performance of the max-Dombi GA on test problem A.6.

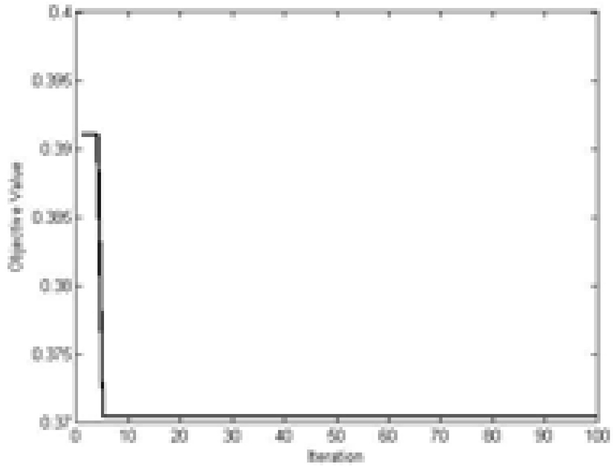

Figure 7: The performance of the max-Dombi GA on test problem A.7.

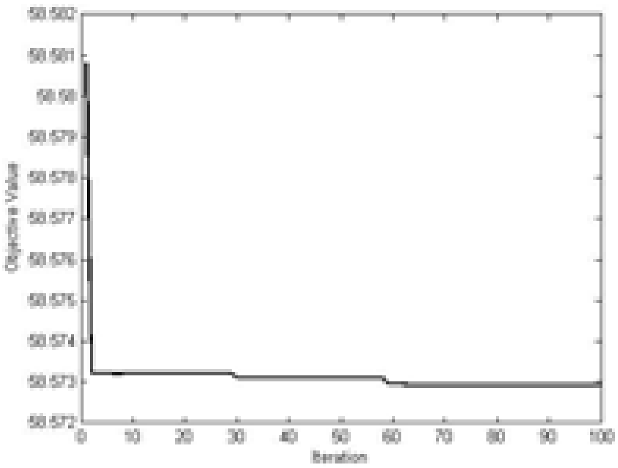

Figure 8: The performance of the max-Dombi GA on test problem A.8.

5.2. Comparisons with other works

As mentioned before, we can make a comparison between the current GA, max-min GA [38] and max-product GA [23]. For this purpose, all the test problems described in Appendix B have been designed in such a way that they are feasible for both the minimum and product t-norms.

The first comparison is against max-min GA, and we apply our algorithm (modified for the minimum t-norm) to the test problems by considering φ as the minimum t-norm. The results are shown in Table 3 including the optimal objective values found by the current GA and max-min GA. As is shown in this table, the current GA finds better solutions for test problems 1, 5 and 6, and the same solutions for the other test problems.

Table 4 shows that the current GA finds the optimal values faster than max-min GA and hence has a higher convergence rate, even for the same solutions. The only exception is test problem 8 in which all the results are the same. In all the cases, results marked with "*" indicate the better cases.

The second comparison is against the max-product GA. In this case, we apply our algorithm (modified for the product t-norm) to the same test problems by considering φ as the product t-norm (Tables 5 and 6).

The results, in Tables 5 and 6, demonstrate that the current GA produces better solutions (or the same solutions with a higher convergence rate) when compared against max-product GAs for all the test problems.

Table 3. Best results found by our algorithm and max-min GA.

Test problems	Lu and Fang	Our algorithm
B.1	8.4296755	8.4296754*
B.2	-1.3888	-1.3888
B.3	0	0
B.4	5.0909	5.0909
B.5	71.1011	71.0968*
B.6	-0.3291	-0.4175*
B.7	-0.6737	-0.6737
B.8	93.9796	93.9796

Table 4. A Comparison between the results found by the current GA and max-min GA.

Test problems		Max-min GA	Our GA
B.1	Average best-so-far Median best-so-far Average mean fitness	8.4297014 8.4296755 8.4308865	8.4296796* 8.4296755 8.4298745*
B.2	Average best-so-far Median best-so-far Average mean fitness	-1.3888 -1.3888 -1.3877	-1.3888 -1.3888 -1.3886*
B.3	Average best-so-far Median best-so-far Average mean fitness	0 0 7.1462e-07	0 0 0*
B.4	Average best-so-far Median best-so-far Average mean fitness	5.0909 5.0909 5.0910	5.0909 5.0909 5.0908
B.5	Average best-so-far Median best-so-far Average mean fitness	71.1011 71.1011 71.1327	71.0969* 71.0968* 71.1216*
B.6	Average best-so-far Median best-so-far Average mean fitness	-0.3291 -0.3291 -0.3287	-0.4175* -0.4175* -0.4162*
B.7	Average best-so-far Median best-so-far Average mean fitness	-0.6737 -0.6737 -0.6736	-0.6737 -0.6737 -0.6737*
B.8	Average best-so-far Median best-so-far Average mean fitness	93.9796 93.9796 93.9796	93.9796 93.9796 93.9796

Table 5. Best results found by our algorithm and max-product GA.

Test problems	Hassanzadeh et al.	Our algorithm
B.1	13.61740269	13.61740246*
B.2	-1.5557	-1.5557
B.3	0	0
B.4	5.8816	5.8816
B.5	45.0650	45.0314*
B.6	-0.3671	-0.4622*
B.7	-2.470232	-2.470232
B.8	38.0195	38.0150*

Table 6. A Comparison between the results found by the current GA and max-product GA.

Test problems		Max-product GA	Our GA
B.1	Average best-so-far Median best-so-far Average mean fitness	13.61745044 13.61740371 13.61785924	13.61740502* 13.61740260* 13.61781613*
B.2	Average best-so-far Median best-so-far Average mean fitness	-1.5557 -1.5557 -1.5524	-1.5557 -1.5557 -1.5557**
B.3	Average best-so-far Median best-so-far Average mean fitness	0 0 1.5441e-05	0 0 0*
B.4	Average best-so-far Median best-so-far Average mean fitness	5.8816 5.8816 5.8823	5.8816 5.8816 5.8816*
B.5	Average best-so-far Median best-so-far Average mean fitness	45.0650 45.0650 45.1499	45.0315* 45.0314* 45.0460*
B.6	Average best-so-far Median best-so-far Average mean fitness	-0.3671 -0.3671 -0.3668	-0.4622* -0.4622* -0.4614*
B.7	Average best-so-far Median best-so-far Average mean fitness	-2.470232 -2.470232 -2.470175	-2.470232 -2.470232 -2.470213*
B.8	Average best-so-far Median best-so-far Average mean fitness	-38.0195 38.0195 38.0292	38.0150* 38.0150* 38.0171*

Conclusion

In this paper, we studied the resolution of FRE defined by Dombi family of t-norms. Two necessary and sufficient conditions to determine the feasibility of the problem were presented. Moreover, two simplification approaches (depending on the Dombi t-norm) were proposed to simplify the problem. A nonlinear optimization problem was introduced in which the constraints were defined by the max- Dombi fuzzy relational equations, and then a genetic algorithm was designed for solving the nonlinear optimization problems constrained by the max-Dombi FRE. Additionally, we presented a method for generating feasible max-Dombi FREs. These feasible max-Dombi FREs were used as test problems for the performance evaluation of the proposed algorithm. Experiments were performed with the proposed method in the generated feasible test problems. We conclude that the proposed GA can find the optimal solutions for all the cases with a great convergence rate. On the other hand, a comparison was made between the proposed method and max-min and max-product GAs, which solve the nonlinear optimization problems subjected to the FREs defined by max-min and max-product compositions, respectively. The results showed that the proposed method (modified by minimum and product t-norms) finds better solutions compared with the solutions obtained by the other algorithms.

As future works, we aim at testing our algorithm in other type of nonlinear optimization problems whose constraints are defined as FRE or FRI with other well-known t-norms.

Appendix A

Test Problem A.1:

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

$$b^T = [0.6593\ ,\ 0.9251\ ,\ 0.9907]$$

$$A = \begin{bmatrix} 0.5082 & 0.6828 & 0.0823 & 0.7285 \\ 0.0871 & 0.9377 & 0.9281 & 0.4201 \\ 0.9972 & 0.9986 & 0.6336 & 0.6708 \end{bmatrix}$$

Test Problem A.2:

$$f(x) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4 + x_4x_5,$$

$$b^T = [0.7033\ ,\ 0.0236\ ,\ 0.3997\ ,\ 0.5589]$$

$$A = \begin{bmatrix} 0.3871 & 0.1002 & 0.5191 & 0.7801 & 0.0665 \\ 0.0015 & 0.9770 & 0.0165 & 0.0172 & 0.2430 \\ 0.0340 & 0.2792 & 0.4441 & 0.0811 & 0.5393 \\ 0.9359 & 0.4002 & 0.6339 & 0.2232 & 0.4569 \end{bmatrix}$$

Test Problem A.3:

$$f(x) = x_1x_2 - Ln(1 + x_3x_4x_5) - x_6,$$

$$b^T = [0.9429\ ,\ 0.8851\ ,\ 0.0264\ ,\ 0.1020]$$

$$A = \begin{bmatrix} 0.9726 & 0.9923 & 0.9459 & 0.2384 & 0.9836 & 0.9776 \\ 0.9526 & 0.4838 & 0.2885 & 0.4926 & 0.0984 & 0.6302 \\ 0.0022 & 0.7135 & 0.3188 & 0.0080 & 0.9447 & 0.0012 \\ 0.0506 & 0.9041 & 0.6870 & 0.2511 & 0.5076 & 0.0524 \end{bmatrix}$$

Test Problem A.4:

$$f(x) = x_1 + 2x_2 + 4x_5 + e^{x_1 x_4 - x_6},$$

$$b^T = [0.3832\ ,\ 0.1350\ ,\ 0.6031\ ,\ 0.7859\ ,\ 0.1066]$$

$$A = \begin{bmatrix} 0.5101 & 0.1286 & 0.8728 & 0.7647 & 0.1861 & 0.3285 \\ 0.1372 & 0.7924 & 0.1917 & 0.0065 & 0.0458 & 0.0140 \\ 0.7710 & 0.2655 & 0.1213 & 0.7554 & 0.4540 & 0.6565 \\ 0.5321 & 0.3390 & 0.3605 & 0.9896 & 0.8661 & 0.9327 \\ 0.0396 & 0.1126 & 0.9172 & 0.3115 & 0.0572 & 0.0404 \end{bmatrix}$$

Test Problem A.5:

$$f(x) = \sum_{k=1}^{6} [100(x_{k+1} - x_k^2)^2 + (1 - x_k)^2],$$

$$b^T = [0.2802\ ,\ 0.8195\ ,\ 0.5566\ ,\ 0.3531\ ,\ 0.2913]$$

$$A = \begin{bmatrix} 0.6954 & 0.1031 & 0.6531 & 0.1181 & 0.1560 & 0.0548 & 0.2789 \\ 0.8380 & 0.9445 & 0.9994 & 0.4365 & 0.5134 & 0.8513 & 0.9068 \\ 0.3760 & 0.5579 & 0.2102 & 0.5141 & 0.5867 & 0.1125 & 0.9765 \\ 0.8964 & 0.1761 & 0.4983 & 0.4131 & 0.1949 & 0.3573 & 0.0184 \\ 0.1851 & 0.0230 & 0.0991 & 0.7249 & 0.1189 & 0.2683 & 0.0854 \end{bmatrix}$$

Test Problem A.6:

$$f(x) = -0.5(x_1 x_4 - x_2 x_3 + x_2 x_6 - x_5 x_6 + x_5 x_4 - x_6 x_7),$$

$$b^T = [0.4922\ ,\ 0.4999\ ,\ 0.4241\ ,\ 0.7686\ ,\ 0.4312\ ,\ 0.4625]$$

$$A = \begin{bmatrix} 0.4270 & 0.9334 & 0.2550 & 0.1860 & 0.8836 & 0.0122 & 0.6802 \\ 0.2118 & 0.9286 & 0.3612 & 0.3449 & 0.9171 & 0.8042 & 0.6817 \\ 0.2532 & 0.5943 & 0.4964 & 0.0255 & 0.2118 & 0.0525 & 0.0607 \\ 0.8783 & 0.6133 & 0.8915 & 0.8936 & 0.8823 & 0.3126 & 0.9859 \\ 0.0023 & 0.8014 & 0.4443 & 0.0138 & 0.2130 & 0.2901 & 0.1190 \\ 0.1476 & 0.6292 & 0.3243 & 0.7276 & 0.3705 & 0.0882 & 0.7825 \end{bmatrix}$$

Test Problem A.7:

$$f(x) = e^{x_1x_2x_3x_4x_5} - 0.5(x_1^3 + x_2^3 + x_6^3 + 1)^2 + 2x_7x_8,$$

$$b^T = [0.9447 , 0.5297 , 0.9050 , 0.6069 , 0.3409 , 0.3532]$$

$$A = \begin{bmatrix} 0.5447 & 0.7634 & 0.2596 & 0.9834 & 0.9762 & 0.6425 & 0.9880 & 0.9859 \\ 0.2786 & 0.2582 & 0.2392 & 0.3074 & 0.9867 & 0.9599 & 0.0567 & 0.7831 \\ 0.7423 & 0.0034 & 0.9897 & 0.4332 & 0.7499 & 0.5841 & 0.9805 & 0.5393 \\ 0.1791 & 0.5852 & 0.9007 & 0.5493 & 0.9481 & 0.2806 & 0.1038 & 0.9601 \\ 0.4272 & 0.5637 & 0.3236 & 0.0698 & 0.2060 & 0.4693 & 0.1328 & 0.9478 \\ 0.0601 & 0.7308 & 0.3557 & 0.1443 & 0.2689 & 0.8754 & 0.1450 & 0.0190 \end{bmatrix}$$

Test Problem A.8

$$f(x) = (x_1-1)^2+(x_7-1)^2+10\sum_{k=1}^{7}(10-k)(x_k^2-x_{k+1})^2 b^T = [0.2992 , 0.7490 , 0.3393 , 0.6067 , 0.3062 , 0.6485 ,$$

$$A = \begin{bmatrix} 0.5702 & 0.4757 & 0.8816 & 0.1956 & 0.2975 & 0.2049 & 0.0977 & 0.0816 \\ 0.8757 & 0.8618 & 0.3322 & 0.8994 & 0.0104 & 0.0250 & 0.9379 & 0.6662 \\ 0.1859 & 0.3957 & 0.5540 & 0.2316 & 0.0413 & 0.2350 & 0.7120 & 0.2155 \\ 0.4771 & 0.3078 & 0.0616 & 0.5245 & 0.5947 & 0.7889 & 0.8649 & 0.0086 \\ 0.8202 & 0.7877 & 0.3791 & 0.0691 & 0.3408 & 0.1502 & 0.1618 & 0.0610 \\ 0.7950 & 0.1145 & 0.9573 & 0.5570 & 0.7174 & 0.3553 & 0.3520 & 0.7773 \\ 0.8853 & 0.7159 & 0.1801 & 0.1777 & 0.1001 & 0.1286 & 0.1916 & 0.1229 \end{bmatrix}$$

Appendix B
Test Problem B.1:

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

$$b^T = [0.2077 , 0.4709 , 0.8443]$$

$$A = \begin{bmatrix} 0.4302 & 0.4464 & 0.0741 & 0.0751 \\ 0.1848 & 0.1603 & 0.4628 & 0.5929 \\ 0.9049 & 0.1707 & 0.8746 & 0.4210 \end{bmatrix}$$

Test Problem B.2:

$$f(x) = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4,$$

$$b^T = [0.4228\ ,\ 0.9427\ ,\ 0.9831]$$

$$A = \begin{bmatrix} 0.1280 & 0.7390 & 0.2852 & 0.2409 \\ 0.9991 & 0.7011 & 0.1688 & 0.9667 \\ 0.1711 & 0.6663 & 0.9882 & 0.6981 \end{bmatrix}$$

Test Problem B.3:

$$f(x) = x_1 x_2 x_3 x_4 x_5,$$

$$b^T = [0.6714\ ,\ 0.5201\ ,\ 0.1500]$$

$$A = \begin{bmatrix} 0.4424 & 0.3592 & 0.6834 & 0.6329 & 0.9150 \\ 0.6878 & 0.7363 & 0.7040 & 0.6869 & 0.2002 \\ 0.6482 & 0.3947 & 0.4423 & 0.0769 & 0.0175 \end{bmatrix}$$

Test Problem B.4:

$$f(x) = x_1 + 2x_2 + 4x_5 + e^{x_1 x_4},$$

$$b^T = [0.6855\ ,\ 0.5306\ ,\ 0.5975\ ,\ 0.2992]$$

$$A = \begin{bmatrix} 0.1025 & 0.7780 & 0.3175 & 0.9357 & 0.7425 \\ 0.0163 & 0.2634 & 0.5542 & 0.4579 & 0.9213 \\ 0.7325 & 0.2481 & 0.8753 & 0.2405 & 0.4193 \\ 0.1260 & 0.2187 & 0.6164 & 0.7639 & 0.2962 \end{bmatrix}$$

Test Problem B.5:

$$f(x) = \sum_{k=1}^{6} [100(x_{k+1} - x_k^2)^2 + (1 - x_k)^2],$$

$$b^T = [0.5846\ ,\ 0.8277\ ,\ 0.4425\ ,\ 0.8266]$$

$$A = \begin{bmatrix} 0.1187 & 0.4147 & 0.8051 & 0.3876 & 0.3643 & 0.7031 \\ 0.4761 & 0.8606 & 0.4514 & 0.0311 & 0.5323 & 0.1964 \\ 0.6618 & 0.2715 & 0.3826 & 0.0302 & 0.7117 & 0.1784 \\ 0.9081 & 0.1459 & 0.7896 & 0.9440 & 0.8715 & 0.1265 \end{bmatrix}$$

Test Problem B.6:

$$f(x) = -0.5(x_1 x_4 - x_2 x_3 + x_2 x_6 - x_5 x_6 + x_5 x_4 - x_6 x_7),$$

$$b^T = [0.9879\ ,\ 0.6321\ ,\ 0.8082\ ,\ 0.6650]$$

$$A = \begin{bmatrix} 0.0832 & 0.3312 & 0.4580 & 0.7001 & 0.8287 & 0.9978 & 0.1876 \\ 0.3904 & 0.4277 & 0.2302 & 0.1373 & 0.4850 & 0.3495 & 0.8831 \\ 0.2393 & 0.8619 & 0.2734 & 0.8265 & 0.6598 & 0.4328 & 0.9315 \\ 0.4863 & 0.3787 & 0.6748 & 0.9301 & 0.4564 & 0.5893 & 0.8943 \end{bmatrix}$$

Test Problem B.7:

$$f(x) = e^{x_1 x_2 x_3 x_4 x_5} - 0.5(x_1^3 + x_2^3 + x_6^3 + 1)^2,$$

$$b^T = [0.9521 , 0.0309 , 0.8627 , 0.8343 , 0.6290]$$

$$A = \begin{bmatrix} 0.9869 & 0.0805 & 0.8373 & 0.1417 & 0.9988 & 0.6320 \\ 0.0139 & 0.0169 & 0.0182 & 0.4379 & 0.0295 & 0.5095 \\ 0.2497 & 0.6914 & 0.8961 & 0.3504 & 0.8225 & 0.2433 \\ 0.9691 & 0.6170 & 0.5921 & 0.4785 & 0.5994 & 0.5714 \\ 0.6197 & 0.6298 & 0.2372 & 0.5874 & 0.2560 & 0.9817 \end{bmatrix}$$

Test Problem B.8:

$$f(x) = (x_1-1)^2+(x_7-1)^2+10\sum_{k=1}^{6}(10-k)(x_k^2-x_{k+1})^2 b^T = [0.7840 , 0.4648 , 0.8864 , 0.8352 , 0.9839]$$

$$A = \begin{bmatrix} 0.8522 & 0.2376 & 0.3586 & 0.7260 & 0.8891 & 0.2771 & 0.1316 \\ 0.4673 & 0.8176 & 0.1173 & 0.5350 & 0.1426 & 0.0020 & 0.2892 \\ 0.9707 & 0.4058 & 0.7248 & 0.1826 & 0.6193 & 0.8108 & 0.9630 \\ 0.8412 & 0.4663 & 0.7011 & 0.1124 & 0.6848 & 0.9434 & 0.4656 \\ 0.0785 & 0.9515 & 0.9997 & 0.0028 & 0.4982 & 0.6384 & 0.3852 \end{bmatrix}$$

References

[1] Chaira T., Fuzzy mathematical morphology using triangular operators and its application to images, Journal of Intelligent and Fuzzy Systems, 28 (5) (2015), 2269-2277.

[2] Chang C. W., Shieh B. S., Linear optimization problem constrained by fuzzy max–min relation equations, Information Sciences 234 (2013) 71–79.

[3] Chen L.,Wang P. P., Fuzzy relation equations (i): the general and specialized solving algorithms, Soft Computing 6 (5) (2002) 428-435.

[4] Chen L.,Wang P. P., Fuzzy relation equations (ii): the branch-point-solutions and the categorized minimal solutions, Soft Computing 11 (2) (2007) 33-40.

[5] Chen J., Ye J. , Some Single-Valued Neutrosophic Dombi Weighted Aggregation Operators for Multiple Attribute Decision-Making, Symmetry 9 (6) (2017), 82-93.

[6] [4]. Dempe S., Ruziyeva A., On the calculation of a membership function for the solution of a fuzzy linear optimization problem, Fuzzy Sets and Systems 188 (2012) 58-67.

[7] Di Martino F., Loia V., Sessa S., Digital watermarking in coding/decoding processes with fuzzy relation equations, Soft Computing 10(2006) 238-243.

[8] Di Nola A.,Sessa S., Pedrycz W.,Sanchez E., Fuzzy relational equations and their applications in knowledge engineering, Dordrecht: Kluwer academic press, 1989.

[9] Dombi J., On a certain class of aggregative operators, Information Sciences 245 (2013) 313–328.

[10] Dubey D.,Chandra S., Mehra A., Fuzzy linear programming under interval uncertainty based on IFS representation, Fuzzy Sets and Systems 188 (2012) 68-87.

[11] Dubois D., Prade H., Fundamentals of Fuzzy Sets, Kluwer, Boston, 2000.

[12] Fan Y. R., Huang G. H., Yang A. L., Generalized fuzzy linear programming for decision making under uncertainty: Feasibility of fuzzy solutions and solving approach, Information Sciences 241 (2013) 12-27.

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

[14] Fernandez M. J., Gil P., Some specific types of fuzzy relation equations, Information Sciences 164 (2004) 189-195.

[15] Freson S., De Baets B., De Meyer H., Linear optimization with bipolar max–min constraints, Information Sciences 234 (2013) 3–15.

[16] Ghodousian A.,Khorram E., An algorithm for optimizing the linear function with fuzzy relation equation constraints regarding max-prod composition, Applied Mathematics and Computation 178 (2006) 502-509.

[17] Ghodousian A.,Khorram E., Fuzzy linear optimization in the presence of the fuzzy relation inequality constraints with max-min composition, Information Sciences 178 (2008) 501-519.

[18] Ghodousian A., Khorram E., Linear optimization with an arbitrary fuzzy relational inequality, Fuzzy Sets and Systems 206 (2012) 89-102.

[19] Ghodousian A., Khorram E., Solving a linear programming problem with the convex combination of the max-min and the max-average fuzzy relation equations, Applied Mathematics and computation 180 (2006) 411-418.

[20] Guo F. F., Pang L. P., Meng D.,Xia Z. Q., An algorithm for solving optimization problems with fuzzy relational inequality constraints, Information Sciences 252 (2013) 20-31.

[21] Guo F. F., Xia Z. Q., 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.

[22] Guu S. M.,Wu Y. K., Minimizing a linear objective function under a max-t-norm fuzzy relational equation constraint, Fuzzy Sets and Systems 161 (2010) 285-297.

[23] Guu S. M., Wu Y. K., Minimizing a linear objective function with fuzzy relation equation constraints, Fuzzy Optimization and Decision Making 12 (2002) 1568-4539.

[24] Guu S. M., Wu Y. K., Minimizing an linear objective function under a max-t-norm fuzzy relational equation constraint, Fuzzy Sets and Systems 161 (2010) 285-297.

[25] Han S. C.,Li H. X., Notes on pseudo-t-norms and implication operators on a complete Brouwerian lattice and pseudo-t-norms and implication operators: direct products and direct product decompositions, Fuzzy Sets and Systems 153(2005) 289-294.

[26] Hassanzadeh R., Khorram E., Mahdavi I.,Mahdavi-Amiri N., A genetic algorithm for optimization problems with fuzzy relation constraints using max-product composition, Applied Soft Computing 11 (2011) 551-560.

[27] Hock W.,Schittkowski K., Test examples for nonlinear programming codes Lecture Notes in Economics and Mathematical Systems, vol. 187, Springer, New York, 1981.

[28] Khorram E.,Shivanian E., Ghodousian A., Optimization of linear objective function subject to fuzzy relation inequalities constraints with max-average composition , Iranian Journal of Fuzzy Systems 4 (2) (2007) 15-29.

[29] Khorram E.,Ghodousian A., Linear objective function optimization with fuzzy relation equation constraints regarding max-av composition, Applied Mathematics and Computation 173 (2006) 872-886.

[30] Khorram E.,Ghodousian A.,Molai A. A., Solving linear optimization problems with max-star composition equation constraints, Applied Mathematic and Computation 178 (2006) 654-661.

[31] Klement E. P., Mesiar R.,Pap E., Triangular norms. Position paper I: Basic analytical and algebraic properties, Fuzzy Sets and Systems 143(2004) 5-26.

[32] Lee H. C., Guu S. M., On the optimal three-tier multimedia streaming services, Fuzzy Optimization and Decision Making 2(1) (2002) 31-39.

[33] Li P. K., Fang S. C., 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.

[34] Li P., Liu Y., Linear optimization with bipolar fuzzy relational equation constraints using lukasiewicz triangular norm, Soft Computing 18 (2014) 1399-1404.

[35] Li P.,Fang S. C., A survey on fuzzy relational equations, part i: classification and solvability, Fuzzy Optimization and Decision Making 8 (2009) 179-229.

[36] Lin J. L., On the relation between fuzzy max-archimedean t-norm relational equations and the covering problem, Fuzzy Sets and Systems 160 (2009) 2328-2344.

[37] Lin J. L., Wu Y. K., Guu S. M., On fuzzy relational equations and the covering problem, Information Sciences 181 (2011) 2951-2963.

[38] Liu C. C.,Lur Y. Y., Wu Y. K., Linear optimization of bipolar fuzzy relational equations with max-Łukasiewicz composition, Information Sciences 360 (2016) 149–162.

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

[40] Loia V., Sessa S., Fuzzy relation equations for coding/decoding processes of images and videos, Information Sciences 171(2005) 145-172.

[41] Lu J., Fang S. C., Solving nonlinear optimization problems with fuzzy relation equations constraints, Fuzzy Sets and Systems 119(2001) 1-20.

[42] Markovskii A. V., On the relation between equations with max-product composition and the covering problem, Fuzzy Sets and Systems 153 (2005) 261-273.

[43] Pedrycz W., Granular Computing: Analysis and Design of Intelligent Systems, CRC Press, Boca Raton, 2013.

[44] Pedrycz W., Fuzzy relational equations with generalized connectives and their applications, Fuzzy Sets and Systems 5(1983) 185-201.

[45] Pedrycz W., Vasilakos A. V., Modularization of fuzzy relational equations, Soft Computing 6(2002) 3-37.

[46] Peeva K., Resolution of fuzzy relational equations-methods, algorithm and software with applications, Information Sciences 234 (2013) 44-63.

[47] Perfilieva I., Fuzzy function as an approximate solution to a system of fuzzy relation equations, Fuzzy Sets and Systems 147(2004) 363-383.

[48] Perfilieva I., Novak V., System of fuzzy relation equations model of IF-THEN rules, Information Sciences 177 (**??**) (2007) 3218-3227.

[49] Perfilieva I., Finitary solvability conditions for systems of fuzzy relation equations, Information Sciences 234 (2013)29-43.

[50] Qu X. B., Wang X. P., Lei Man-hua. H., Conditions under which the solution sets of fuzzy relational equations over complete Brouwerian lattices form lattices, Fuzzy Sets and Systems 234 (2014) 34-45.

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

[52] Sanchez E., Equations de relations floves, These Biologie Humaine, Marseille, France (1972).

[53] Shieh B. S., Infinite fuzzy relation equations with continuous t-norms, Information Sciences 178 (2008) 1961-1967.

[54] Shieh B. S., Minimizing a linear objective function under a fuzzy max-t-norm relation equation constraint, Information Sciences 181 (2011) 832-841.

[55] Sun F., Conditions for the existence of the least solution and minimal solutions to fuzzy relation equations over complete Brouwerian lattices, Information Sciences 205 (2012) 86-92.

[56] Sun F., Wang X. P., Qu x. B., Minimal join decompositions and their applications to fuzzy relation equations over complete Brouwerian lattices, Information Sciences 224 (2013) 143-151.

[57] Wang S., Fang S. C.,. Nuttle H. L. W, Solution sets of interval-valued fuzzy relational equations, Fuzzy Optimization and Decision Making 2 (1)(2003) 41-60.

[58] Wang P. Z., Latticized linear programming and fuzzy relation inequalities, Journal of Mathematical Analysis and Applications 159(1991) 72-87.

[59] Wu Y. K., Guu S. M., A note on fuzzy relation programming problems with max-strict-t-norm composition, Fuzzy Optimization and Decision Making 3(2004) 271-278.

[60] Wu Y. K., Guu S. M., An efficient procedure for solving a fuzzy relation equation with max-Archimedean t-norm composition, IEEE Transactions on Fuzzy Systems 16 (2008) 73-84.

[61] Wu Y. K., Optimization of fuzzy relational equations with max-av composition, Information Sciences 177 (2007) 4216-4229.

[62] Wu Y. K.,Guu S. M., Minimizing a linear function under a fuzzy max-min relational equation constraints, Fuzzy Sets and Systems 150 (2005) 147-162.

[63] Wu Y. K., Guu S. M., Liu J. Y., 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.

[64] Xiong Q. Q., Wang X. P., Fuzzy relational equations on complete Brouwerian lattices, Information Sciences 193 (2012) 141-152.

[65] Xiong Q. Q., Wang X. P., Some properties of sup-min fuzzy relational equations on infinite domains, Fuzzy Sets and Systems 151(2005) 393-402.

[66] Yang X. P., Zhou X. G., Cao B. Y., Latticized linear programming subject to max-product fuzzy relation inequalities with application in wireless communication, Information Sciences 358–359 (2016) 44–55.

[67] Yang S. J., An algorithm for minimizing a linear objective function subject to the fuzzy relation inequalities with addition-min composition, Fuzzy Sets and Systems 255 (2014) 41-51.

[68] Yeh C. T., On the minimal solutions of max-min fuzzy relation equations, Fuzzy Sets and Systems 159 (2008) 23-39.