Journal of Algorithms and Computation

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

Solving a non-convex non-linear optimization problem constrained by fuzzy relational equations and Sugeno-Weber family of t-norms

A. Ghodousian*[1], A. Ahmadi†[2] and A. Dehghani‡[3]

[1, 2, 3]Faculty of Engineering Science, College of Engineering, University of Tehran, P.O.Box 11365-4563, Tehran, Iran.

ABSTRACT

Sugeno-Weber family of t-norms and t-conorms is one of the most applied one in various fuzzy modelling problems. This family of t-norms and t-conorms was suggested by Weber for modeling intersection and union of fuzzy sets. Also, the t-conorms were suggested as addition rules by Sugeno for so-called λ–fuzzy measures. In this paper, we study a nonlinear optimization problem where the feasible region is formed as a system of fuzzy relational equations (FRE) defined by the Sugeno-Weber t-norm. We firstly investigate the resolution of the feasible region when it is defined with max-Sugeno-Weber composition and present some necessary and sufficient conditions for determining the feasibility of the problem. Also, two procedures are presented for simplifying the problem. Since the feasible solutions set of FREs

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

ARTICLE INFO

Article history:
Received 10, March 2017
Received in revised form 08, November 2017
Accepted 30 November 2017
Available online 01, December 2017

AMS subject Classification: 05C78.

*Corresponding author: A. Ghodousian, Email: a.ghodousian@ut.ac.ir
†ahmadi91@ut.ac.ir
‡ali.dehghani@ut.ac.ir

ABSTRACT Continued
is non-convex and the finding of all minimal solutions is an NP-hard problem, conventional nonlinear programming methods may not be directly employed. For these reasons, a genetic algorithm is presented, which preserves the feasibility of new generated solutions. The proposed GA does not need to initially find the minimal solutions. Also, it does not need to check the feasibility after generating the new solutions. Additionally, we propose a method to generate feasible max-Sugeno-Weber 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.

1 Introduction

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

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

whrere $I = \{1, 2, ..., m\}$, J $=\{1, 2, ..., n\}$, $A = (a_{ij})_{m\times n}$, $0 \leqslant a_{ij} \leqslant 1$ $(\forall i \in I \ and \ \forall j \in J)$, is a fuzzy matrix , $b = (b_i)_{m\times 1}$, $0 \leqslant b_i \leqslant 1$ $(\forall i \in I\ )$,is an m –dimensional fuzzy vector, and "φ" is the max-Sugeno-Weber composition, that is, $\varphi(x, y)$=$T^{\lambda}_{SW}(x, y)$=max $\{\frac{x+y-1+\lambda xy}{1+\lambda}\ , 0\}$ in which $\lambda > -1$.
If a_i is the i'th row of matrix A , then problem (1) can be expressed as follows:

$$\min f(x)$$
$$\varphi(a_i, x) = b_i\,,\ i \in I$$
$$x \in [0,1]^n$$

where the constraints mean:

$$\varphi(a_i, x) = \max_{j\in J}\{\varphi(a_{ij}, x_j)\} = \max_{j\in J}\{T^{\lambda}_{SW}(a_{ij}, x_j)\}$$

$$= \max_{j\in J}\{max\{\frac{a_{ij} + x_j - 1 + \lambda a_{ij}x_j}{1+\lambda}\ , 0\}\} = b_i\ \ , \forall i \in I$$

The members of the family $\{T^{\lambda}_{SW}\}$ are increasing functions of the parameter λ. It can be easily shown that Sugeno-Weber t-norm $T_{SW}^{\lambda}(x,y)$ converges to the product fuzzy intersection xy as λ goes to infinity and converges to Drastic product t-norm as λ approaches -1 [8]. Also, it is interesting to note that $T_{SW}^{0}(x,y) = max\{x+y-1,0\}$, that is, the Sugeno-Weber t-norm is converted to Lukasiewicz t-norm if $\lambda = 0$.

The problem to determine an unknown fuzzy relation R on universe of discourses $U \times V$ such that $A\varphi R = B$, where A and B are given fuzzy sets on U and V , respectively, and φ is an composite operation of fuzzy relations, is called the problem of fuzzy relational equations (FRE). Since Sanchez [51] proposed the resolution of FRE defined by max-min composition, different fuzzy relational equations were generalized in many theoretical aspects and utilized in many applied problems such as fuzzy control, discrete dynamic systems, prediction of fuzzy systems, fuzzy decision making, fuzzy pattern recognition, fuzzy clustering, image compression and reconstruction, fuzzy information retrieval, and so on [5,11,18,22,37,41,42,45,48,57,59,65]. 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 FRE. The authors in [42] used particular FRE for the compression/decompression of color images in the RGB and YUV spaces.

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 [1,9,13,14,15,16,17,18,19,20,21,25,26,27,30,35,54,61,66]. 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 $min\{c^T x : A \varphi x^{\circ} b, x \in [0, 1]^n\}$, where φ is max-min composition and "$\circ$" 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 $z(x) = \max\limits_{i=1}^{n}\{min\{c_i, x_i\}\}$ 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 $z(x) = x_1 \vee x_2 \vee ... \vee x_n$ 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, the most important difficulties related to FRE or FRI problems can be categorized as follows:

1. 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.

2. A feasible region formed as FRE or FRI [15] is often a non-convex set.

3. 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 Sugeno-Weber 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 (Sugeno-Weber 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-Sugeno-Weber 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. A comparative study is presented in section 4 and, finally in section 5 the experimental results are demonstrated.

2 Some basic properties of max-Sugeno-Weber FREs

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 $S_{T_{SW}^{\lambda}}(a_i, b_i)$ denote the feasible solutions set of i 'th equation, that is, $S_{T_{SW}^{\lambda}}(a_i, b_i)=\{x \in [0,1]^n : \max_{j=1}^{n}\{T_{SW}^{\lambda}(a_{ij}, x_j)\} = b_i\}$. Also, let $S_{T_{SW}^{\lambda}}(A, b)$ denote the feasible solutions set of problem (1). Based on the foregoing

notations, it is clear that $S_{T_{SW}^{\lambda}}(A, b) = \bigcap_{i\in I} S_{T_{SW}^{\lambda}}(a_i, b_i)$.

Definition 1. For each $i \in I$, we define $J_i = \{j \in J : a_{ij} \geq b_i\}$.

According to definition 1, we have the following lemmas.

Lemma 1. Let $i \in I$. If $j \notin J_i$, then $T_{SW}^{\lambda}(a_{ij}, x_j) < b_i$, $\forall x_j \in [0,1]$.

Proof. From the monotonicity and identity law of t-norms, we have

$T_{SW}^{\lambda}(a_{ij}, x_j) \leq T_{SW}^{\lambda}(a_{ij}, 1) = a_{ij}$, $\forall x_j \in [0,1]$. Now, the result follows from the assumption

(i.e., $j \notin J_i$) and definition 1.

Lemma 2. Let $i \in I$ and $j \in J_i$.

(a) If $x_j > \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}}$, then $T_{SW}^{\lambda}(a_{ij}, x_j) > b_i$.

(b) If $x_j = \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}}$, then $T_{SW}^{\lambda}(a_{ij}, x_j) = b_i$.

(b) If $x_j < \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}}$ and $b_i \neq 0$, then $T_{SW}^{\lambda}(a_{ij}, x_j) < b_i$.

(d) If $x_j \leq \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}}$ and $b_i = 0$, then $T_{SW}^{\lambda}(a_{ij}, x_j) = b_i$.

Proof. The proof is easily obtained from the definition of Sugeno-Weber t-norm and definition 1.

Lemma 3 below gives a necessary and sufficient condition for the feasibility of sets $S_{T_{SW}^{\lambda}}(a_i, b_i), \forall i \in I$.

Lemma 3. For a fixed $i \in I, S_{T_{SW}^{\lambda}}(a_i, b_i) \neq \varnothing$ if and only if $J_i \neq \varnothing$

Proof. Suppose that $S_{T_{SW}^{\lambda}}(a_i, b_i) \neq \varnothing$. So, there exists $x \in [0,1]^n$ such that $\max_{j=1}^{n}\{T_{SW}^{\lambda}(a_{ij}, x_j)\} = b_i$. Therefore, we must have $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) = b_i$ for some $j_0 \in J$. Now, lemma 1 implies $j_0 \in J_i$ that means $J_i \neq \varnothing$. Conversely, suppose that $J_i \neq \varnothing$ and let $j_0 \in J_i$.

We define $\dot{x} = [\dot{x}_1, \dot{x}_2, ..., \dot{x}_n] \in [0,1]^n$ where

$$\dot{x}_j = \begin{cases} \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}} & j = j_0 \\ 0 & j \neq j_0 \end{cases} , \forall j \in J$$

By this definition, we have $T_{SW}^{\lambda}(a_{ij_0}, \dot{x}_{j_0}) = b_i$ and $T_{SW}^{\lambda}(a_{ij}, \dot{x}_j) = 0 \leq b_i$ for each $j \in J - \{j_0\}$. Therefore,

$$\underset{j=1}{\overset{n}{max}}\{T_{SW}^{\lambda}(a_{ij},\dot{x}_j)\} = max\{T_{SW}^{\lambda}(a_{ij_0},\dot{x}_{j_0}), \underset{\substack{j\in J\\ j\neq j_0}}{max}\{T_{SW}^{\lambda}(a_{ij},\dot{x}_j)\}\}$$

$$= T_{SW}^{\lambda}(a_{ij_0},\dot{x}_{j_0}) = b_i$$

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

Definition 2. Suppose that $i \in I$ and $S_{T_{SW}^{\lambda}}(a_i, b_i) \neq \varnothing$ (hence, $J_i \neq \varnothing$ 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:

$$(\hat{x_i})_k = \begin{cases} \frac{(1+\lambda)b_i+(1-a_{ik})}{1+\lambda a_{ik}} & k \in J_i \\ 1 & k \notin J_i \end{cases}, \forall k \in J$$

Also, for each $j \in J_i$, we define $\breve{x}_i(j) = [\breve{x}_i(j)_1, \breve{x}_i(j)_2, ..., \breve{x}_i(j)_n] \in [0,1]^n$ such that

$$\breve{x}_i(j)_k = \begin{cases} \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}} & b_i \neq 0 \quad \text{and} \quad k = j \\ 0 & \text{otherwise} \end{cases}, \forall k \in J$$

The following theorem characterizes the feasible region of the i 'th relational equation $(i \in I)$.

Theorem 1. Let $i \in I$. If $S_{T_{SW}^{\lambda}}(a_i, b_i) \neq \varnothing$, then $S_{T_{SW}^{\lambda}}(a_i, b_i) = \bigcup_{j\in J_i} [\breve{x}_i(j), \hat{x}_i]$.

Proof. Firstly, we show that $\bigcup_{j\in J_i} [\breve{x}_i(j), \hat{x}_i] \subseteq S_{T_{SW}^{\lambda}}(a_i, b_i)$.
Then, we prove that $x \notin \bigcup_{j\in J_i} [\breve{x}_i(j), \hat{x}_i]$ implies $x \notin S_{T_{SW}^{\lambda}}(a_i, b_i)$. The second statement is equivalent to $S_{T_{SW}^{\lambda}}(a_i, b_i) \subseteq \bigcup_{j\in J_i} [\breve{x}_i(j), \hat{x}_i]$, and then the result follows. Let, $\dot{x} \in \bigcup_{j\in J_i} [\breve{x}_i(j), \hat{x}_i]$. Thus, there exists some $j_0 \in J_i$ such that $\dot{x} \in [\breve{x}_i(j_0), \hat{x}_i]$ (i.e. , $\breve{x}_i(j_0) \leq \dot{x} \leq \hat{x}_i$). In the first case,

suppose that $b_i \neq 0$.

So, definition 2 implies $\dot{x}_{j_0} = \frac{(1+\lambda)b_i+(1-a_{ij_0})}{1+\lambda a_{ij_0}}$, $\dot{x}_j = [0, \frac{(1+\lambda)b_i+(1-a_{ij})}{1+\lambda a_{ij}}]$, $\forall j \in J_i - \{j_0\}$, and

$\dot{x}_j \in [0,1]$, $\forall j \notin J_i$.

Therefore $T^{\lambda}_{SW}(a_{ij}, \dot{x}_j) < b_i$, $\forall j \notin J_i$ (resulted from lemma 1), and then

$\underset{j \notin J_i}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} < b_i$. Also, $T^{\lambda}_{SW}(a_{ij}, \dot{x}_j) \le b_i$, $\forall j \in J_i - \{j_0\}$ (resulted from lemma 2, parts (b) and (c)), which implies

$\underset{j \in J_i - \{j_0\}}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} \le b_i$. Additionally, $T^{\lambda}_{SW}(a_{ij_0}, \dot{x}_{j_0}) = b_i$ from lemma 2 (part (b)).

Hence, we have

$$\overset{n}{\underset{j=1}{max}}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} = max\{T^{\lambda}_{SW}(a_{ij_0}, \dot{x}_{j_0}), \underset{j \in J_i - \{j_0\}}{max} T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)$$

$$, \underset{j \notin J}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\}\} = b_i$$

Otherwise, suppose that $b_i = 0$. In this case, definition 2 implies $\dot{x}_j = [0, \frac{(1+\lambda)b_i + (1-a_{ij})}{1+\lambda a_{ij}}]$, $\forall j \in J_i$, and $\dot{x}_j \in [0,1]$, $\forall j \notin J_i$. By similar arguments we have $\underset{j \notin J_i}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} < b_i$
Also , $\underset{j \in J_i}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} = b_i$ (resulted from lemma 2, part (d)). Therefore,

$$\overset{n}{\underset{j=1}{max}}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} = max\{\underset{j \in J_i}{max} T^{\lambda}_{SW}(a_{ij}, \dot{x}_j), \underset{j \notin J_i}{max}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\}\} = b_i$$

.

Thus, for each case $\dot{x} \in S_{T^{\lambda}_{SW}}(a_i, b_i)$ that implies $\bigcup_{j \in J_i} [\breve{x}_i(j), \hat{x}_i] \subseteq S_{T^{\lambda}_{SW}}(a_i, b_i)$.
Conversely, assume that $\dot{x} \notin \bigcup_{j \in J_i} [\breve{x}_i(j), \hat{x}_i]$. Hence, either $\dot{x}$ is not less than $\hat{x}_i$ (i.e., $\dot{x} \not\le \hat{x}_i$) or $\dot{x}$ is not greater than $\breve{x}_i(j)$, $\forall j \in J_i$ (i.e., $\dot{x} \not\ge \breve{x}_i(j)$, $\forall j \in J_i$). If $\dot{x} \not\le \hat{x}_i$, there must exists some

$k \in J$ such that $\dot{x}_k > (\hat{x}_i)_k$. Therefore from definition 2 we must have $\dot{x}_k > 1$, for $k \notin J_i$, and $\dot{x}_k > \frac{(1+\lambda)b_i + (1-a_{ik})}{1+\lambda a_{ik}}$, for $k \in J_i$. In the former case, the infeasibility of $\dot{x}$ is obvious. In the latter case, lemma 2 (part (a)) implies $T^{\lambda}_{SW}(a_{ik}, \dot{x}_k) > b_i$. Therefore, $\overset{n}{\underset{j=1}{max}}\{T^{\lambda}_{SW}(a_{ij}, \dot{x}_j)\} > b_i$ that means $\dot{x} \notin S_{T^{\lambda}_{SW}}(a_i, b_i)$. Otherwise, suppose that $\dot{x} \not\ge \breve{x}_i(j)$, $\forall j \in J_i$. Since each solution $\breve{x}_i(j)$ ($j \in J_i$) has at most one positive component $\breve{x}_i(j)_j$ (from definition 2), we conclude $\dot{x}_j < \breve{x}_i(j)_j$ ($\forall j \in J_i$) . So, for each $j \in J_i$ we have $\dot{x}_j < 0$, if $b_i = 0$, and $\dot{x}_j < \frac{(1+\lambda)b_i + (1-a_{ij})}{1+\lambda a_{ij}}$, if $b_i \neq 0$. In the former case, the result trivially follows. In the latter case, lemma 2 (part (c)) implies $T^{\lambda}_{SW}(a_{ij}, \dot{x}_j) < b_i$ ($\forall j \in J_i$)

.Therefore , $\underset{j\in J_i}{max}\{T^{\lambda}_{SW}(a_{ij},\dot{x}_j)\} < b_i$, and then we have

$$\overset{n}{\underset{j=1}{max}}\{T^{\lambda}_{SW}(a_{ij},\dot{x}_j)\} = max\{\underset{j\in J_i}{max}\ T^{\lambda}_{SW}(a_{ij},\dot{x}_j), \underset{j\notin J_i}{max}\{T^{\lambda}_{SW}(a_{ij},\dot{x}_j)\}\} < b_i$$

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

From theorem 1, $\hat{x}_i$ is the unique maximum solution and $\breve{x}_i(j)$'s $(j \in J_i)$ are the minimal solutions of $S_{T^{\lambda}_{SW}}(a_i,b_i)$.

Definition 3. Let $\hat{x}_i (i \in I)$ be the maximum solution of $S_{T^{\lambda}_{SW}}(a_i,b_i)$. We define $\overline{X} = \underset{i\in I}{min}\{\hat{x}_i\}$.

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
$\underline{X}(e)_j = \underset{i\in I}{max}\{\breve{x}_i(e(i))_j\} = \underset{i\in I}{max}\{\breve{x}_i(j_i)_j\}$, $\forall j \in J$.

Theorem 2 below completely determines the feasible solutions set of problem (1).

Theorem 2. $S_{T^{\lambda}_{SW}}(A,b) = \bigcup_{e\in E}[\underline{X}(e), \overline{X}]$.
Proof. Since $S_{T^{\lambda}_{SW}}(A,b) = \bigcap_{i\in I} S_{T^{\lambda}_{SW}}(a_i,b_i)$, from theorem 1 we have

$$S_{T^{\lambda}_{SW}}(A,b) = \bigcap_{i\in I}\bigcup_{j\in J_i}[\breve{x}_i(j),\hat{x}_i] = \bigcap_{i\in I}\bigcup_{e\in E}[\breve{x}_i(e(i)),\hat{x}_i] = \bigcup_{e\in E}\bigcap_{i\in I}[\breve{x}_i(e(i)),\hat{x}_i] =$$

$$\bigcup_{e\in E}[\underset{i\in I}{max}\{\breve{x}_i(e(i))\}, \underset{i\in I}{min}\{\hat{x}_i\}] = \bigcup_{e\in E}[\underline{X}(e),\overline{X}]$$

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 $S_{T^{\lambda}_{SW}}(A,b)$. Moreover, we have the following corollary that is directly resulted from theorem 2.

Corollary 1(first necessary and sufficient condition).

$S_{T_{SW}^{\lambda}}(A,b) \neq \varnothing$ if and only if $\overline{X} \in S_{T_{SW}^{\lambda}}(A,b)$.

The following example illustrates the above-mentioned definitions.

Example 1. Consider the problem below with Sugeno-Weber t-norm

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

where $\varphi(x,y) = T_{SW}^{1}(x,y) = max\{\frac{x+y-1+xy}{2}, 0\}$ $(i.e., \lambda = 1)$.

By definition 1, we have

$J_1 = \{1,4\}, J_2 = \{1,5\}, J_3 = \{2,5,6\}, J_4 = \{1,4,5\}$ 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:
$\hat{x}_1 = [0.7895, 1, 1, 1, 1, 1]$,
$\hat{x}_2 = [1, 1, 1, 1, 1, 1]$,
$\hat{x}_3 = [1, 0.7778, 1, 1, 1, 0.6842]$,
$\hat{x}_4 = [0.8947, 1, 1, 1, 1, 1]$,
$\hat{x}_5 = [1, 1, 0.8182, 0.6667, 1, 0.1765]$,

$\breve{x}_1(1) = [0.7895, 0, 0, 0, 0, 0]$,
$\breve{x}_1(4) = [0, 0, 0, 1, 0, 0]$,
$\breve{x}_2(1) = [1, 0, 0, 0, 0, 0]$,
$\breve{x}_2(5) = [0, 0, 0, 0, 1, 0]$,
$\breve{x}_3(2) = [0, 0.7778, 0, 0, 0, 0]$,
$\breve{x}_3(5) = [0, 0, 0, 0, 1, 0]$,
$\breve{x}_3(6) = [0, 0, 0, 0, 0, 0.6842]$,
$\breve{x}_4(1) = [0.8947, 0, 0, 0, 0, 0]$,
$\breve{x}_4(4) = [0, 0, 0, 1, 0, 0]$,
$\breve{x}_4(5) = [0, 0, 0, 0, 1, 0]$,
$\breve{x}_5(j) = [0, 0, 0, 0, 0, 0]$, $j \in \{1, 2, 3, 4, 5, 6\}$

Therefore, by theorem 1 we have

$S_{T_{SW}^{\lambda}}(a_1, b_1) = [\breve{x}_1(1), \hat{x}_1] \cup [\breve{x}_1(4), \hat{x}_1]$,
$S_{T_{SW}^{\lambda}}(a_2, b_2) = [\breve{x}_2(1), \hat{x}_2] \cup [\breve{x}_2(5), \hat{x}_2]$,
$S_{T_{SW}^{\lambda}}(a_3, b_3) = [\breve{x}_3(2), \hat{x}_3] \cup [\breve{x}_3(5), \hat{x}_3] \cup [\breve{x}_3(6), \hat{x}_3]$,

$S_{T_{SW}^{\lambda}}(a_4, b_4) = [\check{x}_4(1), \hat{x}_4] \cup [\check{x}_4(4), \hat{x}_4] \cup [\check{x}_4(5), \hat{x}_4]$,
and $S_{T_{SW}^{\lambda}}(a_5, b_5) = [0_{1\times 6}, \hat{x}_5]$ where $0_{1\times 6}$ is a zero vector.

From definition 3, $\overline{X} = [0.7895, 0.7778, 0.8182, 0.6667, 1, 0.1765]$. It is easy to verify that $\overline{X} \in S_{T_{SW}^{\lambda}}(A, b)$. Therefore, the above problem is feasible by corollary 1. Finally, the cardinality of set E is equal to 36 (definition 4). So, we have 36 solutions $\underline{X}(e)$ associated to 36 vectors e . For example, for $e = [1, 5, 5, 5, 1]$, we obtain
$\underline{X}(e) = max\{\check{x}_1(1), \check{x}_2(5), \check{x}_3(5), \check{x}_4(5), \check{x}_5(1)\}$
from definition 5 that means
$\underline{X}(e) = [0.7895, 0, 0, 0, 1, 0]$.

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 Sugeno-Weber 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 $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) < b_i$, $\forall x \in S_{T_{SW}^{\lambda}}(A, b)$. In this case, it is obvious that $\max_{j=1}^{n}\{T_{SW}^{\lambda}(a_{ij}, x_j)\} = b_i$
is equivalent to $\max_{\substack{j=1 \\ j\neq j_0}}^{n}\{T_{SW}^{\lambda}(a_{ij}, x_j)\} = b_i$, 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 $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) < b_i$, $\forall x \in S_{T_{SW}^{\lambda}}(A, b)$, 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 $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) < b_i$, $\forall x \in S_{T_{SW}^{\lambda}}(A, b)$. But, from lemma 1 we have $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) < b_i$, $\forall x_{j_0} \in [0, 1]$. Thus , $T_{SW}^{\lambda}(a_{ij_0}, x_{j_0}) < b_i$, $\forall x \in S_{T_{SW}^{\lambda}}(A, b)$.

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 $j_0 \in J_{i_2}$ for some $i_2 \in I (i_1 \neq i_2)$ and

$$\frac{(1+\lambda)b_{i_2}+(1-a_{i_2j_0})}{1+\lambda a_{i_2j_0}}<\frac{(1+\lambda)b_{i_1}+(1-a_{i_1j_0})}{1+\lambda a_{i_1j_0}}$$

, then “resetting $a_{i_1j_0}$ to zero” is an equivalence operation .

Proof. Similar to the proof of lemma 4, we show that $T^{\lambda}_{SW}(a_{i_1j_0},x_{j_0})<b_i$, $\forall x\in S_{T^{\lambda}_{SW}}(A,b)$. Consider an arbitrary feasible solution $x\in S_{T^{\lambda}_{SW}}(A,b)$. Since $x\in S_{T^{\lambda}_{SW}}(A,b)$, it turns out that $T^{\lambda}_{SW}(a_{i_1j_0},x_{j_0})>b_{i_1}$ never holds. So, assume that $T^{\lambda}_{SW}(a_{i_1j_0},x_{j_0})=b_{i_1}$, that is , $max\{\frac{a_{i_1j_0}+x_{j_0}-1+\lambda a_{i_1j_0}x_{j_0}}{1+\lambda},0\}=b_{i_1}$. Since $b_{i_1}\neq 0$, we conclude that $\frac{a_{i_1j_0}+x_{j_0}-1+\lambda a_{i_1j_0}x_{j_0}}{1+\lambda}=b_{i_1}$, or equivalently $x_{j_0}=\frac{(1+\lambda)b_{i_1}+(1-a_{i_1j_0})}{1+\lambda a_{i_1j_0}}$. Now, from

$$\frac{(1+\lambda)b_{i_2}+(1-a_{i_2j_0})}{1+\lambda a_{i_2j_0}}<\frac{(1+\lambda)b_{i_1}+(1-a_{i_1j_0})}{1+\lambda a_{i_1j_0}}$$

, we obtain $x_{j_0}>\frac{(1+\lambda)b_{i_2}+(1-a_{i_2j_0})}{1+\lambda a_{i_2j_0}}$ Therefore, from lemma 2 (part (a)), we have $T^{\lambda}_{SW}(a_{i_2j_0},x_{j_0})>b_{i_2}$ that contradicts $x\in S_{T^{\lambda}_{SW}}(A,b)$.

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: $a_{12},a_{13},a_{15},a_{16};a_{22},a_{23},a_{24},a_{26};a_{31},a_{33},a_{34};a_{42},a_{43},a_{46}$; in all of these cases, $a_{ij}<b_i$, that is, $j\notin J_i$. Moreover, from the second simplification (lemma 5), we can change the values of components $a_{14},a_{21},a_{36},a_{41}$ and a_{44} to zeros with no effect on the solutions set of the problem. For example, since $a_{36}>b_3$ (i.e. , $6\in J_3$) , $b_3\neq 0$, $a_{56}>b_5$ (i.e. , $6\in J_5$) and

$$0.1250=\frac{(1+\lambda)b_5+(1-a_{56})}{1+\lambda a_{56}}<\frac{(1+\lambda)b_3+(1-a_{36})}{1+\lambda a_{36}}=0.4643$$

“resetting a_{36} to zero” is an equivalence operation.

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 $\tilde{J}_i=\{j\in J:\tilde{a}_{ij}\geq b_i\}$ $(i\in I$) 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).

$S_{T_{SW}^{\lambda}}(A,b) \neq \varnothing$ if and only if $\tilde{J}_i \neq \varnothing$, $\forall i \in I$.

Proof. Since $S_{T_{SW}^{\lambda}}(A,b) = S_{T_{SW}^{\lambda}}(\tilde{A},b)$ from lemmas 4 and 5, it is sufficient to show that $S_{T_{SW}^{\lambda}}(\tilde{A},b) \neq \varnothing$ if and only if $\tilde{J}_i \neq \varnothing$, $\forall i \in I$.
Let $S_{T_{SW}^{\lambda}}(\tilde{A},b) \neq \varnothing$. Therefore, $S_{T_{SW}^{\lambda}}(\tilde{a}_i,b_i) \neq \varnothing$, $\forall i \in I$, where $\tilde{a}_i$ denotes i'th row of matrix $\tilde{A}$. Now, lemma 3 implies $\tilde{J}_i \neq \varnothing$, $\forall i \in I$. Conversely, suppose that $\tilde{J}_i \neq \varnothing$, $\forall i \in I$. Again by using lemma 3 we have $\tilde{J}_i \neq \varnothing$, $\forall i \in I$ .By contradiction, suppose that $S_{T_{SW}^{\lambda}}(\tilde{A},b) \neq \varnothing$.
Therefore, $\overline{X} \notin S_{T_{SW}^{\lambda}}(\tilde{A},b)$ from corollary 1, and then there exists $i_0 \in I$ such that $\overline{X} \notin S_{T_{SW}^{\lambda}}(\tilde{a}_{i_0},b_{i_0})$. Since $\underset{j\notin\tilde{J}_i}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} < b_{i_0}$ (from lemma 1), we must have either $\underset{j\in\tilde{J}_i}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} > b_{i_0}$ or $\underset{j\in\tilde{J}_i}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} < b_{i_0}$. 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

$\underset{j\in\tilde{J}_{i_0}}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} \leq \underset{j\in\tilde{J}_{i_0}}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},(\hat{x}_{i_0})_j)\} = b_{i_0}$, and then the former case (i.e., $\underset{j\in\tilde{J}_i}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} > b_{i_0}$) never holds. Therefore, $\underset{j\in\tilde{J}_i}{max}\{T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j)\} < b_{i_0}$ that implies $b_{i_0} \neq 0$ and $T_{SW}^{\lambda}(\tilde{a}_{i_0j},\overline{X}_j) < b_{i_0}$, $\forall j \in \tilde{J}_{i_0}$. Hence, by lemma 2, we must have $\overline{X}_j < \frac{(1+\lambda)b_{i_0}+(1-\tilde{a}_{i_0j})}{1+\lambda\tilde{a}_{i_0j}}$, $\forall j \in \tilde{J}_{i_0}$. On the other hand,
$(1+\lambda)b_{i_0} + (1-\tilde{a}_{i_0j}) \leq 1+\lambda\tilde{a}_{i_0j}$, $\forall j \in \tilde{J}_{i_0}$. 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 $j \in \tilde{J}_{i_j}$ and

$\overline{X}_j = (\hat{x}_{i_j})_j = \frac{(1+\lambda)b_{i_j}+(1-\tilde{a}_{i_jj})}{1+\lambda\tilde{a}_{i_jj}}$.
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 $j \in \tilde{J}_{i_j}$ and

$\frac{(1+\lambda)b_{i_j}+(1-\tilde{a}_{i_jj})}{1+\lambda\tilde{a}_{i_jj}} < \frac{(1+\lambda)b_{i_0}+(1-\tilde{a}_{i_0j})}{1+\lambda\tilde{a}_{i_0j}}$
(because , $\frac{(1+\lambda)b_{i_j}+(1-\tilde{a}_{i_jj})}{1+\lambda\tilde{a}_{i_jj}} = \overline{X}_j < \frac{(1+\lambda)b_{i_0}+(1-\tilde{a}_{i_0j})}{1+\lambda\tilde{a}_{i_0j}}$).
But in these cases, we must have $\tilde{a}_{i_0j}$ ($\forall j \in \tilde{J}_{i_0}$) from the second simplification process. Therefore, $\tilde{a}_{i_0j} < b_{i_0} \neq 0$ ($\forall j \in \tilde{J}_{i_0}$) that is a contradiction.

Remark 1. Since $S_{T_{SW}^{\lambda}}(A,b) = S_{T_{SW}^{\lambda}}(\tilde{A},b)$ (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 for solving problem (1)

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-Sugeno-Weber 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) $x = [x_1, x_2, \ldots, x_n]$ 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 $F \subseteq S_{T^{\lambda}_{SW}}(A, b) and\ F$ is convex. Then, the initial population is generated by randomly selecting individuals from set F .

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

$$(\breve{x}_i)_k = \begin{cases} \frac{(1+\lambda)b_i+(1-a_{ik})}{1+\lambda a_{ik}} & b_i \neq 0 \quad \text{and} \quad k \in \tilde{J}_i \\ 0 & \text{otherwise} \end{cases} ,\forall k \in J$$

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

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

Thus, $\underline{X}(e) \leq \underline{X}, \quad \forall e \in E$.

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

Proof. From theorem 2, we have $S_{T_{SW}^{\lambda}}(\tilde{A}$,b$)=S_{T_{SW}^{\lambda}}(A,b) = \bigcup_{e\in E}[\underline{X}(e), \overline{X}]$.

To prove the lemma, we show that $\underline{X}(e) \leq \underline{X} \leq \overline{X}$, $\forall e \in E$ .Then, we can conclude $[\underline{X}, \overline{X}] \subseteq [\underline{X}(e), \overline{X}], \quad \forall e \in E,$ that implies both $F \subseteq S_{T_{SW}^{\lambda}}(A,b)$ and the convexity of F . But from remark 2, $\underline{X}(e) \leq \underline{X}, \quad \forall e \in E$ Therefore, it is sufficient to prove $\underline{X} \leq \overline{X}$. 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 $\underline{X}_{j_0} = (\breve{x}_{i_1})_{j_0} = \frac{(1+\lambda)b_{i_1}+(1-\tilde{a}_{i_1j_0})}{1+\lambda\tilde{a}_{i_1j_0}}$,

$$\overline{X}_{j_0} = (\hat{x}_{i_2})_{j_0} = \frac{(1+\lambda)b_{i_2}+(1-\tilde{a}_{i_2j_0})}{1+\lambda\tilde{a}_{i_2j_0}}$$

and $\overline{X}_{j0} < \underline{X}_{j_0}$ (i.e,$\frac{(1+\lambda)b_{i_2}+(1-\tilde{a}_{i_2j_0})}{1+\lambda\tilde{a}_{i_2j_0}} < \frac{(1+\lambda)b_{i_1}+(1-\tilde{a}_{i_1j_0})}{1+\lambda\tilde{a}_{i_1j_0}}$). But these cases occur only when $b_{i_1} \neq 0$ and $j_0 \in \tilde{J}_{i_1} \cap \tilde{J}_{i_2}$. These facts together with

$$\frac{(1+\lambda)b_{i_2}+(1-\tilde{a}_{i_2j_0})}{1+\lambda\tilde{a}_{i_2j_0}} < \frac{(1+\lambda)b_{i_1}+(1-\tilde{a}_{i_1j_0})}{1+\lambda\tilde{a}_{i_1j_0}}$$

imply $\tilde{a}_{i_1j_0} = 0$ from the second simplification process. Therefore, $\tilde{a}_{i_1j_0} < b_{i_1}$ 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
$\overline{X} = [0.7895, 0.7778, 0.8182, 0.6667, 1, 0.1765]$. Also, according to example 2, the simplified matrix $\tilde{A}$ is

$$\tilde{A} = \begin{bmatrix} 0.9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0.8 & 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0.1 & 0.2 & 0 & 0.7 \end{bmatrix}$$

From definition 7, we have
$\breve{x}_1 = [0.7895, 0, 0, 0, 0, 0]$,
$\breve{x}_2 = [0, 0, 0, 0, 1, 0]$,
$\breve{x}_3 = [0, 0.7778, 0, 0, 1, 0]$,
$\breve{x}_4 = [0, 0, 0, 0, 1, 0]$,
$\breve{x}_5 = [0, 0, 0, 0, 0, 0]$.
, and then $\overline{X} = \max_{i=1}^{5}\{\breve{x}_i\} = [0.7895, 0.7778, 0, 0, 1, 0]$ .Therefore, set F=$[\underline{X}, \overline{X}]$ is obtained as a collection of intervals:

$$F = [\underline{X}, \overline{X}] = [0.7895, 0.7778, [0, 0.8182], [0, 0.6667], 1, [0, 0.1765]]$$

By generating random numbers in the corresponding intervals, we acquire one initial individual : x=[0.7895,0.7778,0.74,0.666,1,0.07] .

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 $\overline{X} \notin S_{T^{\lambda}_{SW}}(A, b)$,then stop ; the problem is infeasible (corollary1).

3.For $i = 1, 2, \dots, S_{pop}$

Generate a random n-dimensional 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:

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

which essentially defines the weight to be a value of the Gaussian function with argument r , mean 1 , and standard deviation $q\ S_{pop}$, 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:

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

A more detailed analysis of the influence of the parameter q is presented in section 4 .

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 $S_{T^{\lambda}_{SW}}(A,b) \neq \varnothing$ So, from theorem 3 we have $\tilde{J}_i \neq \varnothing$,$\forall i \in I$, Where $\tilde{J}_i = \{j \in J : \tilde{a}_{ij} \geq b_i\}$, $\forall i \in I$ (see definition1 and remark 1).

Definition 8. Let $I^+ = \{i \in I : b_i \neq 0\}$. So, we define D=$\{j \in J :$ if $\exists i \in I^+$ such that $j \in \tilde{J}_i \Rightarrow |\tilde{J}_i| > 1\}$, where $|\tilde{J}_i|$ denotes the cardinality of set $\tilde{J}_i$.

For a given individual x=$[x_1, x_2, \ldots, x_n]$, 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 $x' = [x'_1, x'_2, \ldots, x'_n]$ 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 $|\tilde{J}_{i_0}| = 1$. 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

$$\max_{k=1}^{n}\{T_{SW}^{\lambda}(\tilde{a}_{i_0k},x_k)\}=\max\{\max_{j\in\tilde{J}_{i_0}}\{T_{SW}^{\lambda}(\tilde{a}_{i_0k},x_k)\},\max_{j\notin\tilde{J}_{i_0}}\{T_{SW}^{\lambda}(\tilde{a}_{i_0k},x_k)\}\}$$

$$=\max\{T_{SW}^{\lambda}(\tilde{a}_{i_0j_0},x_{j_0}),\max_{\substack{k=1\\k\neq j_0}}^{n}\{T_{SW}^{\lambda}(\tilde{a}_{i_0k},x_k)\}\}$$

$$=\max\{T_{SW}^{\lambda}(\tilde{a}_{i_0j_0},0),\max_{\substack{k=1\\k\neq j_0}}^{n}\{T_{SW}^{\lambda}(\tilde{a}_{i_0k},x_k)\}\}<b_{i_0}$$

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 $\dot{x}=[\dot{x}_1,\ldots,\dot{x}_n]$.
2. While $D\neq\varnothing$
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 $x\in S_{T_{SW}^{\lambda}}(A,b)$, 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 $D\neq\varnothing$, 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 $S_{T_{SW}^{\lambda}}(A,b)$. 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 algorithm 2) and one parent $pop(k)$(for some $k=1,2,\ldots,S_{pop}$).

2.Generate a randim number $\lambda_1\in[0,1]$.Set $x_{new1}=\lambda_1x'+(1-\lambda_1)\overline{X}$.

3.Let $\lambda_2 = \underset{\substack{j=1 \\ j\neq k}}{\overset{S_{pop}}{min}} \|pop(k) - pop(j)\|$ 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 $x' \in S_{T^{\lambda}_{SW}}(A, b)$, 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-Sugeno-Weber fuzzy relational equations:

Algorithm 4 (construction of feasible Max-Sugeno-Weber FRE).

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

By the following theorem, it is proved that algorithm 4 always generates random feasible max-Sugeno-Weber fuzzy relational equations.

Theorem 4. The solutions set $S_{T_{SW}^{\lambda}}(A,b)$ of FRE (with Sugeno-Weber 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,$J_i \neq \varnothing$, $\forall i \in I$. 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 $i \in I$. Hence, $b_i \neq 0$ and there must exists some $k \in I(k \neq i)$ such that $j_i \in J_k$ and $\frac{(1+\lambda)b_k+(1-a_{kj_i})}{1+\lambda a_{kj_i}} < \frac{(1+\lambda)b_i+(1-a_{ij_i})}{1+\lambda a_{ij_i}}$. But in this case, we must have $a_{kj_i} > \frac{(1+\lambda)b_k+(1-L)}{1+\lambda L}$ in which $L = \frac{(1+\lambda)b_i+(1-a_{ij_i})}{1+\lambda a_{ij_i}}$. This contradicts step 4.

4 Comparative study

As mentioned, GAs emulate the natural evolution by simulating mutation, crossover and selection operators. In this section, to see how the current GA is situated comparing the other GAs designed for FRE problems, we compare theoretically our algorithm with the GAs presented in [23] and [38] . In addition, an experimental comparison is given in the next section.

As the selection strategy, Lu and Fang [38] used the normalized geometric ranking method in which the probability of the r 'th individual being selected is defined by $P_r = q'(1-q)^{r-1}$, where is the probability of selecting the best individual, r is the rank of the individual, $q' = q/(1-(1-q)^{S_{pop}})$ and S_{pop} is the population size. In a similar way, authors in [23] used the normalized arithmetic ranking method. In contrast, we use Gaussian function as the selection strategy, which makes the search more diversified. Following equations (2) and (3), for given parameters q and S_{pop} , the probability $P_{qS_{pop}}$ of choosing one of the qS_{pop} highest ranking individuals is $P_{qS_{pop}} \approx 0.68$ (and $P_{2qS_{pop}} \approx 0.95$). This is due to the characteristic of the normal distribution: around 0.68% of the individuals fall inside the interval $(-\sigma,\sigma)$ around the mean and respectively 0.95% in the interval $(-2\sigma,\sigma)$. For example, for $q = 0.1$ and $S_{pop} = 50$ (as used in experiments in section 5), one of the 5 highest ranking individuals will be used with probability 0.68% , and one of the 10 highest ranking individuals with probability 0.95% .

In [38], the proposed mutation operator decreases one variable of vector x to a random number between $[0, x_j)$ each time (the same mutation operator has been used in [23]). In this mutation operator, a decreasing variable often followed by increasing several other variables to guarantee the feasibility of a new solution. However, in the current GA, the feasibility of the new solution x' is simultaneously obtained by decreasing a proper variable to zero. Therefore, we have no need to revise the new solution to make it feasible. Moreover, since the proposed mutation operator decreases the selected variables to zeros, the new individuals are more likely to have greater distances from the maximum solution $\overline{X}$, especially x' may be even a minimal solution (see remark 4). This strategy increases the ability of the algorithm to expand the search space for finding new individuals.

Finally, authors in both [23] and [38] used the same "three-point" crossover operator. The three-point crossover is defined by three points (two parents x_1 ,x_2 , and the maximum solution $\overline{X}$) 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 $\overline{X}$. 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 $\overline{X}$). 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 contrast, the current crossover operator uses only one parent each time. Offspring x_{new1} is obtained as a random point on the line segment between x' and $\overline{X}$. But, offspring x_{new2} lies close to its parent. This difference between x_{new1} and x_{new2} provides a suitable tradeoff between exploration and exploitation. Also, as is stated in remark 6, the new solutions x_{new1} and x_{new2} are always feasible.

5 Experimental results

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 $\theta = 0.5$,$\zeta = 0.01$,$\lambda = 0.995$ 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 $\lambda = 3$ in ${T_{SW}}^{\lambda}$. Also, the maximum number of iterations is equal to 100 for all the methods.

5.1 Performance of the max-Sugeno-Weber GA

To verify the solutions found by the max-Sugeno-Weber GA, the optimal solutions of the test problems are also needed. Since $S_{T_{SW}^{\lambda}}(A,b)$ 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 Sugeno-Weber 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 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-Sugeno-Weber GA and the above procedure. According to Table 2, the optimal solutions computed by the max-Sugeno-Weber GA and the optimal solutions obtained by the above procedure match very well. Tables 1 and 2, demonstrate the attractive ability of the max-Sugeno-Weber GA to detect the optimal solutions of problem (1). Also, the good convergence rate of the max-Sugeno-Weber GA could be concluded from Table 1 and figures 1-8.

Table 1: Results of applying the max-Sugeno-Weber GA to the eight test problems.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	66.96834	66.96832	66.97396
A.2	-0.534223	-0.534223	-0.533495
A.3	-1.231383	-1.231383	-1.231263
A.4	5.088561	5.088561	5.088632
A.5	76.41313	76.41313	76.41313
A.6	0.216664	0.216664	0.216683
A.7	-1.811296	-1.811296	-1.810535
A.8	78.644803	78.644797	78.649339

Table 2: Comparison of the solutions found by Max-Sugeno-Weber GA and the optimal values of the test problems.

Test problems	Solutions of max-Sugeno-Weber GA	Optimal values
A.1	66.9683	66.96832
A.2	-0.534223	-0.534224
A.3	-1.231383	-1.231385
A.4	5.088561	5.088561
A.5	76.41313	76.4115
A.6	0.216664	0.216664
A.7	-1.811296	-1.811296
A.8	78.644797	78.644797

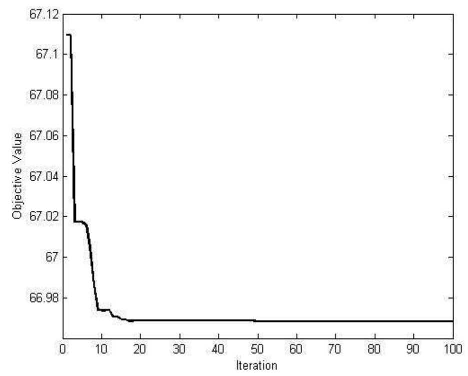

Figure 1: The performance of the max-Sugeno-Weber GA Sugeno- on test problem 1.

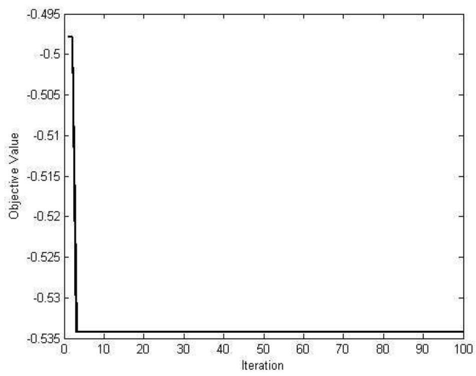

Figure 2: The performance of the max-Weber GA on test problem 2.

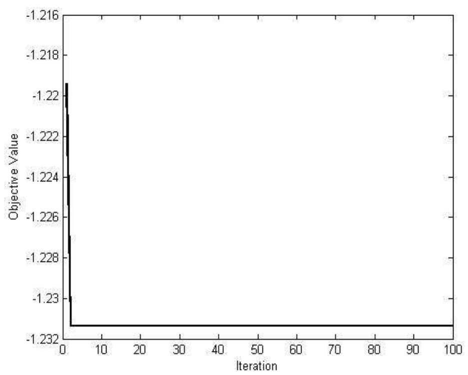

Figure 3: The performance of the max-Sugeno-Weber GA Sugeno- on test problem 3.

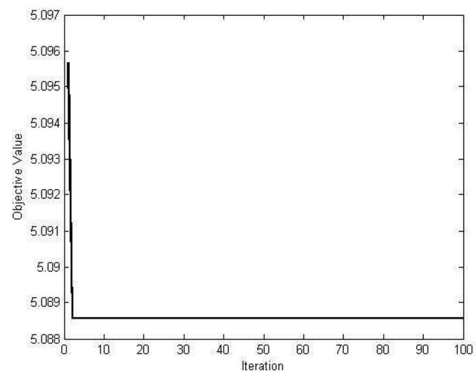

Figure 4: The performance of the max-Weber GA on test problem 4.

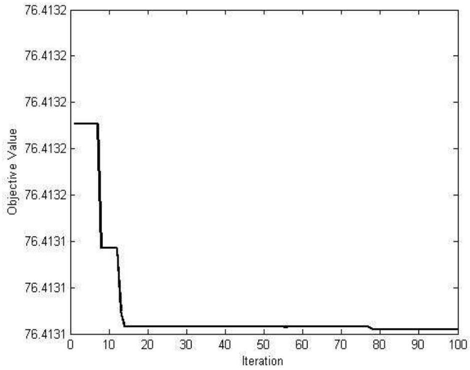

Figure 5: The performance of the max-Sugeno-Weber GA Sugeno- on test problem 5.

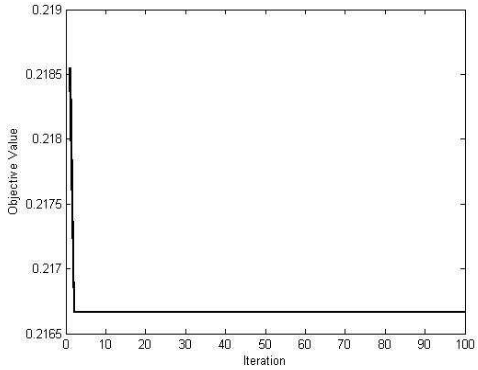

Figure 6: The performance of the max-Weber GA on test problem 6.

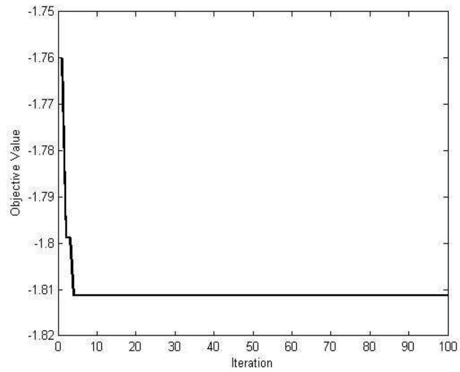

Figure 7: The performance of the max-Sugeno-Weber GA Sugeno- on test problem 7.

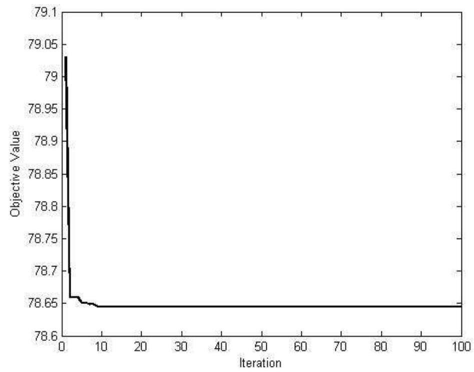

Figure 8: The performance of the max-Weber GA on test problem 8.

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

Test problems		Lu and Fang	Our algorithm
B.1	Average best-so-far	8.4297014	8.4296796*
	Median best-so-far	8.4296755	8.4296755
	Average mean fitness	8.4308865	8.4298745*
B.2	Average best-so-far	-1.3888	-1.3888
	Median best-so-far	-1.3888	-1.3888
	Average mean fitness	-1.3877	-1.3886*
B.3	Average best-so-far	0	0
	Median best-so-far	0	0
	Average mean fitness	7.1462e-07	0*
B.4	Average best-so-far	5.0909	5.0909
	Median best-so-far	5.0909	5.0909
	Average mean fitness	5.0910	5.0908*
B.5	Average best-so-far	71.1011	71.0969*
	Median best-so-far	71.1011	71.0968*
	Average mean fitness	71.1327	71.1216*
B.6	Average best-so-far	-0.3291	-0.4175*
	Median best-so-far	-0.3291	-0.4175*
	Average mean fitness	-0.3287	-0.4162*
B.7	Average best-so-far	-0.6737	-0.6737
	Median best-so-far	-0.6737	-0.6737
	Average mean fitness	-0.6736	-0.6737*
B.8	Average best-so-far	93.9796	93.9796
	Median best-so-far	93.9796	93.9796
	Average mean fitness	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		Max-product GA	Our GA
B.1	Average best-so-far	13.61745044	13.61740502*
	Median best-so-far	13.61740371	13.61740260*
	Average mean fitness	13.61785924	13.61781613*
B.2	Average best-so-far	-1.5557	-1.5557
	Median best-so-far	-1.5557	-1.5557
	Average mean fitness	-1.5524	-1.5557*
B.3	Average best-so-far	0	0
	Median best-so-far	0	0
	Average mean fitness	1.5441e-05	0*
B.4	Average best-so-far	5.8816	5.8816
	Median best-so-far	5.8816	5.8816
	Average mean fitness	5.8823	5.8816*
B.5	Average best-so-far	45.0650	45.0315*
	Median best-so-far	45.0650	45.0314*
	Average mean fitness	45.1499	45.0460*
B.6	Average best-so-far	-0.3671	-0.4622*
	Median best-so-far	-0.3671	-0.4622*
	Average mean fitness	-0.3668	-0.4614*
B.7	Average best-so-far	-2.470232	-2.470232
	Median best-so-far	-2.470232	-2.470232
	Average mean fitness	-2.470175	-2.470213*
B.8	Average best-so-far	38.0195	38.0150*
	Median best-so-far	38.0195	38.0150*
	Average mean fitness	38.0292	38.0171*

6 Conclusion

In this paper, we studied the resolution of FREs defined by the Sugeno-Weber family of t-norms and introduced a nonlinear problem with the max-Sugeno-Weber fuzzy relational equations. In order to determine the feasibility of the problem, two necessary and sufficient conditions were derived. Also, we presented two simplification approaches depending on the Sugeno-Weber t-norm to simplify the problem. A genetic algorithm was proposed for solving the nonlinear optimization problems constrained by the max-Sugeno-Weber FRE. Moreover, we presented a method for generating feasible max-Sugeno-Weber FREs. These feasible FREs were utilized 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. Moreover, 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 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.

7 Acknowledgment

We are very grateful to the anonymous referees and the editor in chief for their comments and suggestions, which were very helpful in improving the paper.

Appendix A

Test Problem A.1:

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

$$b^T = \begin{bmatrix} 0.8983 & 0.6010 & 0.5193 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.1288 & 0.2334 & 0.8095 & 0.9629 \\ 0.8298 & 0.7879 & 0.4601 & 0.4487 \\ 0.8394 & 0.2564 & 0.7787 & 0.3144 \end{bmatrix}$$

Test Problem A.2:

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

$$b^T = \begin{bmatrix} 0.7786 & 0.2373 & 0.3468 & 0.3616 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.1311 & 0.7461 & 0.4680 & 0.5989 & 0.8454 \\ 0.0854 & 0.8333 & 0.3092 & 0.3097 & 0.1570 \\ 0.5236 & 0.4872 & 0.2388 & 0.2346 & 0.2496 \\ 0.1359 & 1.0607 & 0.5913 & 0.4695 & 0.3187 \end{bmatrix}$$

Test Problem A.3:

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

$b^T = \begin{bmatrix} 0.4424 & 0.8200 & 0.3454 & 0.9695 \end{bmatrix}$

$$A = \begin{bmatrix} 0.5187 & 0.9324 & 0.0428 & 0.8363 & 0.1606 & 0.0713 \\ 0.4106 & 0.3265 & 0.6434 & 0.2242 & 0.8498 & 0.6213 \\ 0.2936 & 0.8862 & 0.5514 & 0.9045 & 0.0719 & 0.3287 \\ 0.0001 & 0.0511 & 1.1752 & 0.8503 & 0.2803 & 0.9732 \end{bmatrix}$$

Test Problem A.4:

$f(x) = x_1 + 2x_2 + 4x_5 + e^{x_1x_4 - x_6},$

$b^T = \begin{bmatrix} 0.3489 & 0.1305 & 0.5058 & 0.5365 & 0.3639 \end{bmatrix}$

$$A = \begin{bmatrix} 0.2948 & 0.8676 & 1.4230 & 0.0843 & 0.1903 & 0.0870 \\ 0.9011 & 0.3266 & 0.9803 & 0.2140 & 0.0547 & 0.0546 \\ 0.5668 & 0.9460 & 1.9417 & 0.1000 & 0.5289 & 0.1163 \\ 0.7226 & 0.1070 & 1.4073 & 0.8179 & 0.1933 & 0.3163 \\ 0.0822 & 0.5846 & 1.0050 & 0.5441 & 0.0961 & 0.3857 \end{bmatrix}$$

Test Problem A.5:

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

$b^T = \begin{bmatrix} 0.5781 & 0.1572 & 0.1360 & 0.7476 & 0.5925 \end{bmatrix}$

$$A = \begin{bmatrix} 0.0421 & 0.7560 & 0.3016 & 0.1842 & 1.0425 & 0.3630 & 0.1994 \\ 0.1184 & 0.0399 & 0.5884 & 0.0745 & 0.7625 & 0.2001 & 0.1282 \\ 0.1039 & 0.1807 & 0.4807 & 0.7209 & 0.4503 & 0.1642 & 0.5829 \\ 0.8212 & 0.2989 & 0.2370 & 1.3179 & 0.3754 & 0.3298 & 0.6610 \\ 0.2137 & 0.0123 & 0.4783 & 1.7088 & 0.6534 & 0.7482 & 0.0643 \end{bmatrix}$$

Test Problem A.6:

$f(x) = -0.5(x_1x_4 - x_2x_3 + x_2x_6$
$- x_5x_6 + x_5x_4 - x_6x_7),$

$$b^T = \begin{bmatrix} 0.9716 & 0.8143 & 0.4050 & 0.1165 & 0.5972 & 0.4759 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.3286 & 0.3507 & 0.4794 & 0.0395 & 0.7495 & 0.9996 & 0.0173 \\ 0.8101 & 0.4893 & 1.1555 & 1.2660 & 0.2951 & 0.1741 & 0.8708 \\ 0.4898 & 0.3535 & 0.3386 & 0.7532 & 0.9118 & 0.2006 & 0.2162 \\ 0.6246 & 0.0600 & 0.0228 & 0.1334 & 0.6390 & 0.0472 & 0.1037 \\ 0.3862 & 0.3641 & 0.9550 & 0.4229 & 0.1560 & 0.1882 & 0.0257 \\ 1.3701 & 0.5373 & 0.2954 & 0.7454 & 0.0579 & 0.3314 & 0.0006 \end{bmatrix}$$

Test Problem A.7:

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

$$b^T = \begin{bmatrix} 0.3364 & 0.4433 & 0.7286 & 0.5127 & 0.9257 & 0.9494 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.1615 & 0.4527 & 0.5684 & 0.1564 & 0.3201 & 0.7808 & 0.4241 & 0.2079 \\ 0.0237 & 0.5623 & 0.3919 & 0.2685 & 0.1053 & 0.1429 & 0.5663 & 0.1432 \\ 0.4408 & 0.3131 & 0.6791 & 0.4859 & 0.9134 & 0.2937 & 0.0327 & 0.2587 \\ 0.0018 & 0.7938 & 0.3462 & 0.1977 & 0.3626 & 0.5397 & 0.2654 & 0.5528 \\ 0.9349 & 0.2952 & 0.4759 & 0.6082 & 0.7751 & 0.2702 & 0.7719 & 0.5722 \\ 0.0349 & 0.8649 & 1.2400 & 0.9739 & 0.1920 & 0.5756 & 1.0667 & 0.5092 \end{bmatrix}$$

Test Problem A.8:

$f(x) = (x_1 - 1)^2 + (x_7 - 1)^2$
$+ 10 \sum_{k=1}^{7} (10 - k)(x_k^2 - x_{k+1})^2,$

$$b^T = \begin{bmatrix} 0.9318 & 0.6864 & 0.2702 & 0.0977 & 0.6027 & 0.7675 & 0.0973 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.9507 & 0.1824 & 0.6716 & 0.3393 & 0.4980 & 0.0681 & 0.3028 & 0.8800 \\ 0.4767 & 0.3615 & 0.2162 & 0.6696 & 0.9345 & 0.3157 & 0.7011 & 0.7735 \\ 0.2093 & 0.0504 & 0.2526 & 0.5754 & 0.0571 & 0.4407 & 0.3550 & 0.2951 \\ 0.0847 & 0.1121 & 0.0232 & 0.2573 & 0.0123 & 0.1752 & 0.1558 & 0.1141 \\ 0.5341 & 0.3756 & 0.6194 & 0.7273 & 0.3675 & 0.9322 & 0.5919 & 0.6010 \\ 0.4135 & 0.7710 & 0.7931 & 0.7238 & 0.1424 & 1.1018 & 0.1037 & 0.7139 \\ 0.0202 & 0.0266 & 0.0358 & 0.9970 & 0.2024 & 0.1484 & 0.1082 & 0.0085 \end{bmatrix}$$

Appendix B

Test Problem B.1:

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

$$b^T = \begin{bmatrix} 0.2077 & 0.4709 & 0.8443 \end{bmatrix}$$

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

Test Problem B.2:

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

$$b^T = \begin{bmatrix} 0.4228 & 0.9427 & 0.9831 \end{bmatrix}$$

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

Test Problem B.3:

$$f(x) = x_1x_2x_3x_4x_5,$$

$$b^T = \begin{bmatrix} 0.6714 & 0.5201 & 0.1500 \end{bmatrix}$$

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

Test Problem B.4:

$$f(x) = x_1 + 2x_2 + 4x_5 + e^{x_1x_4},$$

$$b^T = \begin{bmatrix} 0.6855 & 0.5306 & 0.5975 & 0.2992 \end{bmatrix}$$

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

Test Problem B.5:

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

$b^T = \begin{bmatrix} 0.5846 & 0.8277 & 0.4425 & 0.8266 \end{bmatrix}$

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

Test Problem B.6:

$f(x) = -0.5(x_1x_4 - x_2x_3 + x_2x_6$
$- x_5x_6 + x_5x_4 - x_6x_7)$,

$b^T = \begin{bmatrix} 0.9879 & 0.6321 & 0.8082 & 0.6650 \end{bmatrix}$

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

Test Problem B.7:

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

$b^T = \begin{bmatrix} 0.9521 & 0.0309 & 0.8627 & 0.8343 & 0.6290 \end{bmatrix}$

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

Test Problem B.8:

$f(x) = (x_1 - 1)^2 + (x_7 - 1)^2$
$+ 10 \sum_{k=1}^{6} (10 - k)(x_k^2 - x_{k+1})^2,$

$$b^T = \begin{bmatrix} 0.7840 & 0.4648 & 0.8864 & 0.8352 & 0.9839 \end{bmatrix}$$

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

References

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

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

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

[4] Dempe,S., 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.

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

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

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

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

[9] Fan, Y. R., 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.

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

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

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

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

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

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

[16] Ghodousian, A., E. Khorram, 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.

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

[18] Guo, F. F, 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.

[19] Guu, S. M., 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.

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

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

[22] Han, S. C., H. X. Li, 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.

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

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

[25] Khorram, E., E. Shivanian, A. Ghodousian, 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.

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

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

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

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

[30] Li, P. K., 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.

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

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

[33] 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.

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

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

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

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

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

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

[40] Nobuhara, H., K. Hirota, W. Pedrycz, Relational image compression: optimizations through the design of fuzzy coders and YUV colors space, Soft Computing 9 (2005) 471-479.

[41] Nobuhara, H., 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.

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

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

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

[45] Perfilieva, I., V. Novak, System of fuzzy relation equations model of IF-THEN rules, Information Sciences 177 (16) (2007) 3218-3227.

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

[47] Qu, X. B., X. P. Wang, Man-hua. 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.

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

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

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

[51] 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.

[52] Sun, F., 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.

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

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

[55] Wu, Y. K., 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.

[56] Wu, Y. K., 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.

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

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

[59] Wu, Y. K., 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.

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

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

[62] Yang, X. P., 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.

[63] 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.

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