Iranian Journal of Fuzzy Systems

Volume 23, Number 5, (2026), pp. 147-156

Original paper

Vector simulations and minimal quotient automata in vector general fuzzy automata

M. Shamsizadeh [1], Kh. Abolpour [2] and H. Bordbar [3]

[1] *Department of Mathematics and Statistics, Faculty of Energy and Data Sciences, Behbahan Khatam Alanbia University of Technology, Behbahan, Iran*

[2] *Department of Mathematics, Shi.C., Islamic Azad University, Shiraz, Iran*

[3] *Centre for Information Technologies and Applied Mathematics, University of Nova Gorica, Nova Gorica, Slovenia.*

shamsizadeh.m@bkatu.ac.ir, babolpor_kh@yahoo.com, hashem.bordbar@ung.si

Abstract

In this paper, we introduce the concept of vector simulation for vector general fuzzy automata (VGFAs). An algorithm is developed to decide vector simulation between two VGFAs in polynomial time. We prove that the union of vector simulations is again a vector simulation. For any VGFA, we construct its quotient automaton and show that the identity relation is the only vector simulation on the quotient, which implies its minimality.

Keywords: Vector space, vector general fuzzy automaton, vector simulation, minimal automaton.

1 Introduction

The theory of finite automata is a cornerstone of theoretical computer science, providing a rigorous framework for modeling computation and system behavior. Over the decades, extensive research has enriched the algebraic and structural properties of automata, deepening both their theoretical foundations and practical applications [1, 4, 5]. The extension of classical automata into the fuzzy domain, known as fuzzy finite automata (FFAs), was pioneered by Wee [21] and Santos [15, 16] in the late 1960s. Subsequent developments in fuzzy language theory by Lee and Zadeh [10], and Thomason and Marinos [18], established a conceptual basis for analyzing FFA behavior, paving the way for incorporating uncertainty and imprecision into computational models.

Algebraic formalization of FFAs was further advanced by Malik and Mordeson [12, 13], who developed a robust framework for algebraic fuzzy finite automata. Wechler [20] extended the field by introducing abstract set values, enabling generalized fuzzy automata capable of handling imprecise information. These developments significantly broadened the applicability of fuzzy automata in real-world contexts.

FFAs have been applied across diverse domains, including adaptive learning systems [11], pattern recognition [3, 13], database theory [3, 13], and discrete event systems [11, 14]. They also underpin fuzzy neural networks widely used in machine learning, control, and optimization problems [19, 22]. Doostfatemeh and Kremer [2] introduced general fuzzy automata (GFAs), addressing limitations in membership assignment and providing a flexible framework for systems with multiple memberships. Practical applications of GFAs include intelligent selective packet discarding in communication networks [6] and state minimization and dynamic reduction under semantically similar environments, with applications to air pollution modeling in smart cities [7, 8]. Complementary algebraic perspectives have been provided by La Guardia et al. [9], who studied semi-vector spaces and semi-algebras with direct applications in fuzzy automata, enriching the mathematical foundations of the field.

Despite these advances, vector general fuzzy automata (VGFA) remain relatively underexplored. While GFAs provide a flexible modeling framework, a systematic approach for analyzing VGFA behavior and minimizing their

Corresponding Author: M. Shamsizadeh

Received: August 2025; Revised: June 2026; Accepted: August 2026.

https://doi.org/10.22111/ijfs.2026.53052.9378

structure is lacking. In particular, no established notion of *vector simulation* exists to capture behavioral equivalence between VGFAs and support minimization rigorously. Motivated by this gap, the present study introduces the concept of vector simulation for VGFA, develops an efficient algorithm to determine such simulations, and establishes their closure properties. By constructing the quotient VGFA and proving that the identity relation is the only vector simulation on it, we show that the quotient VGFA is minimal. This work not only provides a new theoretical framework for analyzing VGFA but also extends minimization techniques in fuzzy automata to the vector domain while demonstrating computational feasibility.

The remainder of this paper is organized as follows. Section 2 reviews fundamental concepts related to general fuzzy automata. Section 3 introduces vector simulation for VGFAs, presents an algorithm for its computation, and analyzes its time complexity. We then establish closure properties of vector simulations, define the quotient VGFA, and prove that the identity relation is the only vector simulation on the quotient. Finally, we demonstrate the minimality of the quotient VGFA, highlighting the efficiency and optimization inherent in this approach.

2 Preliminaries

In this section, we present some notions that are important in the following section.

Definition 2.1. [17] *Let F be a field and $n \in N_0$ (the set of non-negative integers). Define F_n as the vector space of n-dimensional column vectors over F. A vector general fuzzy automaton (VGFA) is an automaton denoted by*

$$\tilde{F}_v = (Q, X, \tilde{R}, Z, \tilde{\delta}, \tilde{\omega}, E_1, E_2, E_3, E_4),$$

with the following properties:

(i) *Let $k, m, r \in N_0$ such that:*

1. $Q = F_k$ *is a nonempty finite set of states,*
2. $X = F_m$ *is a finite set of input symbols,*
3. $\tilde{R} \subseteq P(\tilde{Q})$ *is the set of L-fuzzy start symbols,*
4. $Z = F_r$ *is a finite set of output symbols.*

(ii) *There exist matrices A (of size $k \times k$), B (of size $k \times m$), and C (of size $r \times k$), all over F, such that:*

1. $\tilde{\delta} : (Q \times [0, 1]) \times X \times Q \to [0, 1]$ *is the augmented transition function, where*
$$\delta(u, a, Au + Ba) \in \Delta,$$
2. $\tilde{\omega} : (Q \times [0, 1]) \times Z \to [0, 1]$ *is the augmented output function,*
3. $E_1 : [0, 1] \times [0, 1] \to [0, 1]$ *is called the membership assignment function. It combines the current membership degree of a state with the membership degree of a transition to produce the membership degree of the next state.*
4. $E_2 : [0, 1] \times [0, 1] \to [0, 1]$ *is called the membership assignment output function. It operates similarly to E_1, but at the output stage: it combines the membership degree of the current state with the degree of the output transition to produce the membership degree of the output symbol.*
5. $E_3 : [0, 1]^* \to [0, 1]$ *is called the multi-membership resolution function. When multiple transitions lead to the same next state (possibly via different paths or inputs), each may contribute a different membership degree.*
6. $E_4 : [0, 1]^* \to [0, 1]$ *is called the multi-membership resolution output function. It serves the same role as E_3, but at the output level: when multiple membership degrees are assigned to the same output symbol, E_4 resolves them into a single value.*

Definition 2.2. [17] *Let*

$$\tilde{F}_v = (Q, X, \tilde{R}, Z, \tilde{\delta}, \tilde{\omega}, E_1, E_2, E_3, E_4),$$

be a VGFA. We define the max-min vector general fuzzy automaton

$$\tilde{F}_v^* = (Q, X, \tilde{R}, Z, \tilde{\delta}^*, \tilde{\omega}, E_1, E_2, E_3, E_4),$$

where $\tilde{\delta}^* : Q \times X^* \times Q \to [0,1] \times [0,1]$ *satisfies the following conditions:*

$$\tilde{\delta}^*((u_1, \mu^{t_i}(u_1)), \Lambda, u_2) = \begin{cases} 1 & \text{if } u_1 = u_2, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For every $i \geq 0$,

$$\tilde{\delta}^*((u_1, \mu^{t_i}(u_1)), a_{i+1}, u_2) = \tilde{\delta}((u_1, \mu^{t_i}(u_1)), a_{i+1}, u_2),$$

and recursively:

$$\tilde{\delta}^*((u_1, \mu^{t_0}(u_1)), a_1 a_2 \dots a_{n-1}, u_n) = \vee \Big\{ \tilde{\delta}((u_1, \mu^{t_0}(u_1)), a_1, u_2) \wedge \tilde{\delta}((u_2, \mu^{t_1}(u_2)), a_2, u_3) \wedge \dots \\ \wedge \tilde{\delta}((u_{n-1}, \mu^{t_{n-2}}(u_{n-1})), a_{n-1}, u_n) \;\Big|\; u_2 \in Q_{act}(t_1), \dots, u_{n-1} \in Q_{act}(t_{n-1}) \Big\}, \tag{2}$$

where $a_i \in X$ *for all* $1 \leq i \leq n-1$.

Since the vector GFA operates in discrete time, we also use the notation:

$$u_{t+1} = Au_t + Ba_t, \tag{3}$$

$$\omega_{u_t} = Cu_t, \tag{4}$$

where $u_t \in Q_{act}(t)$, $a_t \in X$, *and* $\omega_{q_t} \in Z$ *for* $t \in N_0$.

In the rest of paper, vector general fuzzy automaton(VGFA) denote automaton by 13-tuple machine

$$(F, Q, X, A, B, C, \tilde{R}, Z, \tilde{\delta}, \tilde{\omega}, E_1, E_2, E_3, E_4),$$

and $\tilde{M}_{vi}$ denotes VGFA $\tilde{M}_{vi} = (F^i, Q_i, X, A_i, B_i, C_i, \tilde{R}_i = (u_i, 1), Z_i, \tilde{\delta}_i, \tilde{\omega}_i, E_1, E_2, E_3, E_4)$ over field F^i.

3 Vector simulation of vector general fuzzy automata

We begin by introducing the concept of vector simulation for two given VGFA. Next, we prove that the union of any collection of vector simulations also forms a valid vector simulation. Utilizing this union, we construct the largest vector simulation and demonstrate that the corresponding quotient VGFA derived from it is the minimal VGFA.

Definition 3.1. *Let* $\tilde{M}_{v1}$ *and* $\tilde{M}_{v2}$ *be two vector general fuzzy automata (VGFA), and let* $\Vdash$ *be a binary relation on* $Q_1 \times Q_2$. *For any subset* $K \subseteq Q_2$, *define:*

$$Z_{\Vdash}(K) = \{\nu \in Q_1 \mid \exists \nu' \in K, \, \nu \Vdash \nu'\}.$$

Similarly, for any subset $H \subseteq Q_1$, *define:*

$$Z_{\Vdash}(H) = \{\nu' \in Q_2 \mid \exists \nu \in H, \, \nu \Vdash \nu'\}.$$

The relation $\Vdash$ can be extended to subsets of Q_1 and Q_2. For $H \subseteq Q_1$ and $K \subseteq Q_2$, the extension is defined as:

$$\begin{aligned} H \Vdash K &\iff H \subseteq Z_{\Vdash}(K) \text{ and } K \subseteq Z_{\Vdash}(H) \\ &\iff \big(\forall \nu \in H, \, \exists \nu' \in K, \, \nu \Vdash \nu'\big) \text{ and } \big(\forall \nu' \in K, \, \exists \nu \in H, \, \nu \Vdash \nu'\big). \end{aligned}$$

Definition 3.2. *Let* $\tilde{M}_{v1}$ *and* $\tilde{M}_{v2}$ *be two vector general fuzzy automata (VGFAs). The relation* $\Vdash$ *is called a vector simulation (VS) if the following conditions hold:*

1. $u_0^1 \Vdash u_0^2$,
2. $u_1 \Vdash u_2$ *implies that* $A_1 u_1 + B_1 a \Vdash A_2 u_2 + B_2 a$,
3. $u_1 \Vdash u_2$ *implies that* $C_1 u_1 = C_2 u_2$.

Algorithm 1 An algorithm for computing vector simulation of two given vector general fuzzy automata

1: **Input:** Two VGFAs $\tilde{M}_{v_i} = (F^i, Q_i, X, A_i, B_i, C_i, \tilde{R}_i, Z_i, \tilde{\delta}_i, \tilde{\omega}_i, F_1, F_2, F_3, F_4)$, where $Q_1 = \{u_1, u_2, \ldots, u_n\}$, $Q_2 = \{v_1, v_2, \ldots, v_m\}$, $\tilde{R}_1 = (u_1, 1)$, $\tilde{R}_2 = (v_1, 1)$, and $X = \{a_1, a_2, \ldots, a_r\}$.
2: Set $i = 1$.
3: For $k = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, m$, set $u_k \Vdash_i v_l$ if and only if $C_1 u_k = C_2 v_l$.
4: Set $i = i + 1$.
5: Set $j = 1$.
6: For $k = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, m$, set $u_k \Vdash_i v_l$ if and only if $u_k \Vdash_{i-1} v_l$ and $A_1 u_k + B_1 a_j \Vdash_{i-1} A_2 v_l + B_2 a_j$.
7: Increment j by 1.
8: If $j \leq r$, go to Step 6. Otherwise, proceed to the next step.
9: If $\Vdash_i = \Vdash_{i-1}$, proceed to the next step. Otherwise, go to 4.
10: If $u_1 \Vdash_i v_1$, proceed to the next step. Otherwise, go to 12.
11: **Output:** $\Vdash_i$.
12: **Output:** fail.

The algorithm described here computes a vector simulation between two specified VGFAs, denoted as $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$. If no simulation exists between $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$, the algorithm terminates and indicates failure.

The time complexity for Step 3 is $O(mn)$. The sequence from Steps 5 to 8 forms a loop, which will iterate no more than $O(mnr)$ times. Similarly, the series of Steps 4 to 9 also constitutes a loop, with a maximum repetition of $O(mn)$ times. The time complexity for Step 10 does not exceed $O(mn)$. Consequently, the overall time complexity is $O(m^2n^2r)$.

Note 1. *Let $\tilde{M}_{v_k}, k = 1, 2, 3$ be three VGFAs. If $\Vdash_1$ is a vector simulation (VS) between $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$, and $\Vdash_2$ is a VS between $\tilde{M}_{v2}$ and $\tilde{M}_{v3}$, then their composition*

$$\Vdash = \Vdash_1 \circ \Vdash_2 = \{(\nu_1, \nu_2) \mid \exists \nu \text{ such that } \nu_1 \Vdash_1 \nu \text{ and } \nu \Vdash_2 \nu_2\},$$

is a VS between $\tilde{M}_{v1}$ and $\tilde{M}_{v3}$.

Theorem 3.3. *Let $\tilde{M}_{v_i}, i = 1, 2$, be two VGFAs, and let $\{\Vdash_i \mid i \in I\}$ be an arbitrary nonempty set of vector simulations (VSs) between $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$. Let $\Vdash = \bigcup_{i \in I} \Vdash_i$. Then, $\Vdash$ is a VS.*

Proof. Let $\Vdash = \cup_{i \in I} \Vdash_i$. Then $u \Vdash u'$ if and only if $u \Vdash_i u'$, for some $i \in I$. Since I is nonempty, we have $u_0^1 \Vdash u_0^2$. Now, let $u \Vdash u'$. Then there exists $i \in I$ such that $u \Vdash_i u'$. So, for every $a \in X$, $A_1 u + B_1 a \Vdash_i A_2 u' + B_2 a$, for some $i \in I$. Then $A_1 u + B_1 a \Vdash A_2 u' + B_2 a$, for every $a \in X$. Also, if $u \Vdash u'$, then $u \Vdash_i u'$, for some $i \in I$. So, $C_1 u \Vdash_i C_2 u'$, for some $i \in I$. Therefore, $C_1 u \Vdash C_2 u'$. Hence, the claim holds. □

Theorem 3.4. *Let $\tilde{M}_{v_i}, i = 1, 2$, be two VGFAs, and let $\Vdash$ be a vector simulation (VS) between $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$. If $u \Vdash u'$, then for every $x \in X^*, x = a_1 a_2 \ldots a_n$, we have*

$$A_1^{|x|} u + \sum_{i=0}^{|x|-1} A_1^{|x|-i-1} B_1 a_i \Vdash A_2^{|x|} u' + \sum_{i=0}^{|x|-1} A_2^{|x|-i-1} B_2 a_i.$$

Proof. We prove the claim by induction on $|x| = n$. At first, let $u \Vdash u'$ and $n = 0$. Then $x = \Lambda$. So, $A_1^0 u + \Sigma_{i=0}^{-1} A_1^{|x|-i-1} B_1 a_i = u \Vdash u' = A_2^0 u' + \Sigma_{i=0}^{-1} A_2^{0-i-1} B_2 a_i$. Now, let the claim holds, for every $y \in X^*$ such that $|y| = n - 1$ and $n \geq 2$. Let $x = y a_{n-1}, y = a_0 a_1 \ldots a_{n-2} \in X^*$, $a_{n-1} \in X$ and $|x| = n$. Then by the induction hypothesis $A_1^{n-1} u + \Sigma_{i=0}^{n-2} A_1^{n-i-2} B_1 a_i \Vdash A_2^{n-1} u' + \Sigma_{i=0}^{n-2} A_2^{n-i-2} B_2 a_i$. Now, by Definition 3.2,

$$A_1(A_1^{n-1} u + \Sigma_{i=0}^{n-2} A_1^{n-i-2} B_1 a_i) + B_1 a_{n-1} \Vdash A_2(A_2^{n-1} u' + \Sigma_{i=0}^{n-2} A_2^{n-i-2} B_2 a_i) + B_2 a_{n-1}.$$

So,

$$A_1^n u + \Sigma_{i=0}^{n-2} A_1^{n-i-1} B_1 a_i + B_1 a_{n-1} \Vdash A_2^n u' + \Sigma_{i=0}^{n-2} A_2^{n-i-1} B_2 a_i + B_2 a_{n-1}.$$

Then

$$A_1^n u + \Sigma_{i=0}^{n-1} A_1^{n-i-1} B_1 a_i \Vdash A_2^n u' + \Sigma_{i=0}^{n-1} A_2^{n-i-1} B_2 a_i.$$

Hence, the claim holds. □

Definition 3.5. *Let $\tilde{M}_v = (F, Q, X, A, B, C, \tilde{R} = (u_0, 1), Z, \tilde{\delta}, \tilde{\omega}, F_1, F_2, F_3, F_4)$ be a VGFA. The behavior of $\tilde{M}_v$ is the map $\beta : X^* \to Z$ defined by*

$$\beta(x) = Cu', \quad \text{where} \quad u' = A^{|x|}u_0 + \sum_{i=0}^{|x|-1} A^{|x|-i-1}Ba_i, \quad x = a_0a_1 \dots a_{|x|-1}.$$

Example 3.6. *Before presenting the computations, we briefly describe the role of each matrix in both automata. In $\tilde{F}_{v1}$, the state-transition matrix A_1 is the 3×3 anti-diagonal permutation matrix over F_2, which reverses the coordinate ordering of each state vector. The input matrix B_1 is the all-ones column vector, so reading input $b = [1]$ adds $\mathbf{1} = (1, 1, 1)^T \pmod 2$ to the current state, effectively flipping all bits, while reading input $a = [0]$ leaves the state unaffected. The output matrix C_1 extracts the third coordinate of the current state, so the output is determined solely by the last entry of each state vector. Together, these matrices encode a linear automaton over F_2^3 whose transition function is $\tilde{\delta}_1(v, x) = A_1v + B_1x$ and whose output function is $\tilde{\omega}_1(v) = C_1v$.*

Let $\tilde{F}_{v1} = (F, Q_1, X, A_1, B_1, C_1, \tilde{R}_1, Z, \tilde{\delta}_1, \tilde{\omega}_1, E_1, E_2, E_3, E_4)$ be a VGFA defined over field $F = Q_2$ of integers modulo 2 such that $A_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $C_1 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, $Q_1 = \{v_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, v_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_4 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, v_5 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, v_6 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\}$, $X = \{a = [0], b = [1]\}$, $\tilde{R}_1 = (\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, 1)$, $Z = \{\begin{bmatrix} 0 \end{bmatrix}, \begin{bmatrix} 1 \end{bmatrix}\}$.

Similarly, $\tilde{F}_{v2}$ operates over F_2^2: the matrix A_2 is the 2×2 swap permutation, B_2 is the all-ones column vector in F_2^2 (so input $b = [1]$ flips both bits of A_2u), and C_2 extracts the second coordinate as output.

Let $\tilde{F}_{v2} = (F, Q_2, X, A_2, B_2, C_2, \tilde{R}_2, Z, \tilde{\delta}_2, \tilde{\omega}_2, E_1, E_2, E_3, E_4)$ be a VGFA defined over field $F = Q_2$ of integers modulo 2 such that $A_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $B_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $C_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}$, $Q_2 = \{u_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, u_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, u_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, u_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\}$, $\tilde{R}_2 = (\begin{bmatrix} 0 \\ 0 \end{bmatrix}, 1)$.

The following computations determine the transition function $\tilde{\delta}_1(v_i, x) = A_1v_i + B_1x$ for each state $v_i \in Q_1$ and each input $x \in X$. All arithmetic is performed modulo 2. Note that input $a = [0]$ contributes nothing, so the next state is simply A_1v_i, whereas input $b = [1]$ adds $B_1 = (1, 1, 1)^T$, flipping every bit of A_1v_i.

So, we have

$$A_1v_1 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = v_1,$$

$$A_1v_1 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = v_6,$$

$$A_1v_2 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = v_3,$$

$$A_1v_2 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = v_5,$$

$$A_1v_3 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = v_2,$$

$$A_1v_3 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = v_4,$$

$$A_1v_4 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = v_5,$$

$$A_1v_4 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = v_3,$$

$$A_1v_5 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = v_4,$$

$$A_1v_5 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = v_2,$$

$$A_1v_6 + B_1a = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = v_6,$$

$$A_1v_6 + B_1b = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = v_1.$$

The output values are computed via $\tilde{\omega}_1(v_i) = C_1v_i$. Since $C_1 = [0\ 0\ 1]$ extracts the third coordinate, the output is 1 precisely for those states whose third entry is 1, namely v_2, v_5, v_6, and 0 otherwise.

Also, we have $C_1v_1 = 0, C_1v_2 = 1, C_1v_3 = 0, C_1v_4 = 0, C_1v_5 = 1, C_1v_6 = 1$.

Analogously, the transitions of $\tilde{F}_{v2}$ are computed via $\tilde{\delta}_2(u_i, x) = A_2u_i + B_2x$ over F_2^2. Here A_2 swaps the two coordinates, and $B_2 = (1,1)^T$ again flips both bits when input $b = [1]$ is applied.

Similarly,

$$A_2u_1 + B_2a = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad A_2u_1 + B_2b = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$A_2u_2 + B_2a = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad A_2u_2 + B_2b = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$A_2u_3 + B_2a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad A_2u_3 + B_2b = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$A_2u_4 + B_2a = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad A_2u_4 + B_2b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Since $C_2 = [0\ 1]$ extracts the second coordinate, the output is 1 for states with second entry equal to 1, namely u_2 and u_3, and 0 for u_1 and u_4, and $C_2u_1 = 0, C_2u_2 = 1, C_2u_3 = 1, C_2u_4 = 0$.

Having computed the transition and output functions for both automata, we now identify a vector space bisimulation (VS) between $\tilde{F}_{v1}$ and $\tilde{F}_{v2}$. The relation $\Vdash$ pairs states that agree on output values and whose transitions are compatible under the relation: states with output 0 in $\tilde{F}_{v2}$ (namely u_1, u_4) are paired with output-0 states in $\tilde{F}_{v1}$ (namely v_1, v_3, v_4), and states with output 1 (namely u_2, u_3) are paired with output-1 states (namely v_2, v_5, v_6), subject to transition compatibility.

Then $\Vdash = \{(u_1, v_1), (u_2, v_6), (u_3, v_2), (u_4, v_3), (u_3, v_5), (u_4, v_4)\}$ is a VS between $\tilde{M}_{v1}$ and $\tilde{M}_{v2}$.

Example 3.7. *This example illustrates the computation of the output function β of the VGFA $\tilde{M}_{v2}$ from Example 3.6 for two input words: the single symbol a and the two-symbol word ab. Recall that the output for an input word is computed by first evolving the initial state through the transition function $\tilde{\delta}_2$ and then applying the output matrix C_2 to the resulting state. Specifically, for a single input $x \in X$, the next state is $u' = A_2u_0 + B_2x$, and the output is $\beta(x) = C_2u'$, where $u_0 = (0,0)^T$ is the initial state given by $\tilde{R}_2$.*

Let VGFA $\tilde{M}_{v2}$ in Example 3.6. Then we have

For the input word $a = [0]$, since $B_2 a = \mathbf{0}$, the input contributes nothing and the next state is simply $A_2 u_0$. Since $u_0 = (0,0)^T$ is the zero vector, we get $u' = \mathbf{0}$, and the output $C_2 u'$ extracts the second coordinate, which is 0.

$\beta(a) = Cu'$, where

$$u' = Au_1 + Ba = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

So, $\beta(a) = C \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$.

For the two-symbol word ab, the computation involves two successive transitions. Starting from the initial state $u_0 = (0,0)^T$, the automaton first reads $a = [0]$ and then $b = [1]$. The resulting state after reading ab is given by:

$$u' = A_2^2 u_0 + A_2 B_2 a + B_2 b,$$

where the first term propagates the initial state through two steps, the second term accounts for the contribution of the first input a after one further transition, and the third term is the direct contribution of the second input b. Since $u_0 = \mathbf{0}$ and $a = [0]$, the first two terms vanish, and only $B_2 b = (1,1)^T$ survives, giving $u' = (1,1)^T = u_2$. The output $C_2 u'$ then extracts the second coordinate, which is 1.

Also, we have $\beta(ab) = Cu'$, where

$$u' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Then $\beta(ab) = \begin{bmatrix} 1 \end{bmatrix}$.

Theorem 3.8. *Let $\tilde{M}_{v_i}, i = 1, 2$, be two VGFAs, and let $\Vdash$ be a vector simulation (VS) between them. Then, $\tilde{M}_{v_1}$ and $\tilde{M}_{v_2}$ have the same behavior.*

Proof. Let $x \in X^*$, $|x| = n, x = a_0 a_1 ... a_{n-1}$ and $\beta_1(x) = C_1 u$, where $u = A_1^n u_0^1 + \Sigma_{i=0}^{n-1} A_1^{n-i-1} B_1 a_i$. By Definition 3.2, $u_0^1 \Vdash u_0^2$ and by Theorem 3.4,

$$u = A_1^n u_0^1 + \Sigma_{i=0}^{n-1} A_1^{n-i-1} B_1 a_i \Vdash A_2^n u_0^2 + \Sigma_{i=0}^{n-1} A_2^{n-i-1} B_2 a_i.$$

By Definition 3.2, $C_1 u = C_2 (A_2^n u_0^2 + \Sigma_{i=0}^{n-1} A_2^{n-i-1} B_2 a_i)$. So, $\beta_1(x) = \beta_2(x)$. □

Suppose that $|\tilde{M}_v|$ denotes the cardinality of states of VGFA $\tilde{M}_v$.

Definition 3.9. *Let $\tilde{M}_v$ be a VGFA. $\tilde{M}_v$ is called the minimal VGFA if $|\tilde{M}_v| \leq |\tilde{M}'_v|$, for every VGFA $\tilde{M}'_v$.*

A VS between a VGFA and itself is called a VS on VGFA.

Theorem 3.10. *Let $\mathcal{M}$ be the set of all vector simulations a (VSs) on the VGFA $\tilde{M}_v$. Then, the union of all VSs in $\mathcal{M}$ is an equivalence relation on Q as well as a VS on $\tilde{M}_v$.*

Proof. By Theorem 3.3, the union of all VSs in $\mathcal{M}$ is a VS. Since the identity relation is in $\mathcal{M}$, the union of all VSs is reflexive. It is clear that $\Vdash$ is a symmetric relation between Q_1 and Q_2. Now, by induction on i in the algorithm, we show that $\Vdash_i$ is transitive. Let $u_1 \Vdash_1 u_2$ and $u_2 \Vdash_1 u_3$. Then $C_1 u_1 = C_2 u_2$ and $C_2 u_2 = C_3 u_3$. So, $C_1 u_1 = C_3 u_3$ and therefore, $u_1 \Vdash_1 u_3$. Now, suppose that if $u_1 \Vdash_{i-1} u_2$ and $u_2 \Vdash_{i-1} u_3$, then $u_1 \Vdash_{i-1} u_3$. Let $u_1 \Vdash_i u_2$ and $u_2 \Vdash_i u_3$. Then $u_1 \Vdash_{i-1} u_2$ and $u_2 \Vdash_{i-1} u_3$ and for every $a \in X$, $A_1 u_1 + B_1 a \Vdash_{i-1} A_2 u_2 + B_2 a$ and $A_2 u_2 + B_2 a \Vdash_{i-1} A_3 u_3 + B_3 a$. So, by induction hypothesis $u_1 \Vdash_{i-1} u_3$ and for every $a \in X$, $A_1 u_1 + B_1 a \Vdash_{i-1} A_2 u_3 + B_2 a$. Therefore, $u_1 \Vdash_i u_3$. Then the relation $\Vdash$ is a transitive relation between Q_1 and Q_3. So, it is an equivalence relation on Q. □

Note 2. *Let $\Vvdash$ be the union of all VSs on $\tilde{M}_v$ and $D \subseteq Q$. Define*

1. $[K] = \{H \big| K \Vvdash H\}$,
2. $\vDash = \{(K, [K]) \big| K \in Q\}$,
3. $D' = \{[K] \big| K \in D\}$.

Theorem 3.11. *Let $D, E \subseteq Q$. Then*

1. *$D \Vvdash E$ if and only if $D' = E'$.*

2. $D \vDash D'$.

Proof. 1. Let $D \Vdash\!\vdash E$ and $[K] \in D'$. Then $K \in D$. So, there exists $T \in E$ such that $K \Vdash\!\vdash T$. Then $[K] = [T] \in E'$. Therefore, $D' \subseteq E'$. Similarly, $E' \subseteq D'$.

Now, let $D' = E'$ and suppose that $K \in D$. Then $[K] \in D' = E'$. So, $[K] \in B'$ and $K \in B$.

2. Clearly it holds.

□

Let $\Vdash\!\vdash$, be the union of all VSs on VGFA $\tilde{M}_v = (F, Q, X, A, B, C, \tilde{R} = (u_0, 1), Z, \tilde{\delta}, \tilde{\omega}, E_1, E_2, E_3, E_4)$. Define $\tilde{M}'_v$ to be the quotient VGFA of $\tilde{M}_v$ as follows:

- $Q' = \{[K] \mid K \in Q\}$,
- $R' = [u_0]$,
- $\mu^{t_0}([u_0]) = 1$,
- $\delta' : Q' \times X \times Q' \to [0, 1]$ defined by

$$\delta'([K], a, [H]) = \vee\{\delta(K', a, H') \mid K \Vdash\!\vdash K', H \Vdash\!\vdash H'\},$$

- $\omega'([K], z) = \omega(K, z)$.

The rest of the cases are identical to those in $\tilde{M}_v$. Clearly, δ' and ω' are well-defined.

Theorem 3.12. *Let $\tilde{M}_v$ be a VGFA and $\tilde{M}'_v$ be the quotient VGFA of $\tilde{M}_v$. Then $\Vdash$ is a VS between $\tilde{M}_v$ and $\tilde{M}'_v$.*

Proof. Clearly, $u_0 \vDash [u_0]$. Now, let $u_1 \vDash [u_2]$, then $u_1 \Vdash\!\vdash u_2 \vDash [u_2]$. So, $A_1 u_1 + B_1 a \Vdash\!\vdash A_1 u_2 + B_1 a \vDash A_1[u_2] + B_1 a$, for every $a \in X$. Also, let $u_1 \vDash [u_2]$. Then $u_1 \Vdash\!\vdash u_2 \vDash [u_2]$. So, $Cu_1 = Cu_2 = C[u_2]$. Therefore, $\vDash$ is a VS between $\tilde{M}_v$ and $\tilde{M}'_v$. □

Theorem 3.13. *Let $\tilde{M}'_v$ be the quotient of VGFA $\tilde{M}_v$. Then $\beta(\tilde{M}_v) = \beta(\tilde{M}'_v)$.*

Proof. By Theorem 3.12, there is a VS between two VGFAs $\tilde{M}'_v$ and $\tilde{M}_v$. By Theorem 3.8, they have the same behavior. □

Theorem 3.14. *The identity relation is the only VS on the quotient VGFA $\tilde{M}'_v$.*

Proof. Let $\Vdash$ be a VS on $\tilde{M}'_v$ and $[\nu] \Vdash [\nu']$, but $[\nu] \neq [\nu']$. Let $\vDash \circ \Vdash \circ (\vDash)^{-1}$. Then $\nu \vDash [\nu] \Vdash [\nu'] \vDash \nu'$. Therefore, $\nu \Vdash\!\vdash \nu'$, which is a contradiction. So, the claim holds. □

Definition 3.15. *Let $\tilde{M}_v$ be a VGFA. A state $\nu \in Q$ is called accessible if there exists an input sequence $x \in X^*$ such that*

$$\tilde{\delta}^*((u_0, \mu^{t_0}(u_0)), x, u) > 0.$$

We say that $\tilde{M}_v$ is an *accessible VGFA* if and only if every state in Q is accessible.

Theorem 3.16. *Let $\tilde{M}_v$ be an accessible VGFA, and let $\Vdash$ be the greatest VS on Q. Then the quotient VGFA $\tilde{M}'_v$ is a minimal VGFA.*

Proof. Let $\Vdash\!\vdash$ be a VS between $\tilde{M}'_v$ and $\tilde{M}'_{v2}$. Let each state $\tilde{M}'_{v2}$ have a relationship to at least one $\tilde{M}'_v$ state and every $\tilde{M}'_v$ state is related to at most one state of $\tilde{M}_{v2}$. In contrast to Lemma 3.14, the composition $\Vdash\!\vdash$ with its inverse would not be the identity of $tildeM'_v$. Consequently, $\Vdash\!\vdash$ provides a one-to-one correspondence between the states of $\tilde{M}'_v$, and $\tilde{M}'_{v2}$. □

4 Conclusions

In this paper, we introduced the concept of a vector simulation (VS) for vector general fuzzy automata (VGFA). We demonstrated that the union of any family of VSs is itself a VS, highlighting the inherent structure and coherence in these relations. Additionally, we explored the construction of the quotient VGFA associated with a given VGFA. We established that, by utilizing the union of VSs, the quotient VGFA is indeed a minimal VGFA.

Our results provide a deeper understanding of the interplay between vector simulations and VGFA, particularly in terms of minimizing the structure while preserving essential behaviors. Looking ahead, we aim to explore potential connections between σ-algebras and VGFA, which could lead to novel insights into the algebraic properties of these systems. Furthermore, we plan to investigate the relationship between linear spaces, VGFA, and VSs, which may provide a more comprehensive framework for studying these automata in the context of vector spaces and their applications in various mathematical and computational fields.

To illustrate the practical relevance of the theoretical framework developed in this paper, we note that the minimization of VGFA via vector simulation has direct implications in coding theory and cryptography. Specifically, the quotient VGFA construction can be used to reduce the state complexity of linear automata that model error-detecting and error-correcting codes over finite fields, while preserving their input-output behavior. Similarly, in pseudo-random sequence generation, minimal VGFA representations lead to more efficient implementations of linear feedback shift registers over F_2. These connections suggest that the results of this paper are not purely theoretical, and we intend to develop these applications in concrete case studies in future work.

Several promising directions for future research emerge from this study. One avenue is to investigate the extension of vector simulation to **probabilistic or weighted VGFAs**, providing a unified framework for modeling complex and uncertain systems. Another is to explore the connections between vector simulation and algebraic or topological structures, such as **semi-vector spaces** and **σ-algebras**, which may offer new theoretical foundations for VGFA minimization and behavioral analysis. Finally, practical applications of vector simulation–based minimization in areas such as **linear code generation**, **error detection and correction**, and **pseudo-random sequence generation** could provide valuable contributions to **cryptography and coding theory**.

Acknowledgement

The authors contributed equally to this work. All authors have read and agreed to the published version of this paper.
The third author acknowledges the financial support of the Slovenian Research and Innovation Agency (research core funding No. P2-0103).
The authors declare no conflicts of interest.

References

[1] Z. Bavel, J. W. Thomas, *On the decomposability of monadic algebras and automata*, in: Proceedings of the 8th Annual Symposium on Switching and Automata Theory, (1967), 322-335. https://doi.org/10.1109/FOCS.1967.24

[2] M. Doostfatemeh, S. C. Kremer, *New directions in fuzzy automata*, International Journal of Approximate Reasoning, **38**(2) (2005), 175-214. https://doi.org/10.1016/j.ijar.2004.08.001

[3] D. Dubois, H. Prade, *Fuzzy sets and systems: Theory and applications*, Mathematics in Science and Engineering, **144**, Academic Press, New York, NY, USA, 1980.

[4] W. M. L. Holcombe, *Algebraic automata theory*, Cambridge University Press, Cambridge, 1982.

[5] M. Ito, *Algebraic structures of automata*, Theoretical Computer Science, **429** (2012), 164-168. https://doi.org/10.1016/j.tcs.2011.12.035

[6] A. T. Jahromi, P. Karimaghaee, S. G. Haghighi, et al., *Intelligent selective packet discarding using general fuzzy automata*, in: 2008 IEEE International Conference on Systems, Man and Cybernetics, (2008), 3714-3720. https://doi.org/10.1109/ICSMC.2008.4811877

[7] R. Kaur, A. Tripathi, *State minimization of general finite fuzzy automata*, International Journal of Mathematical, Engineering and Management Sciences, **6** (2021), 1709. https://doi.org/10.33889/IJMEMS.2021.6.6.101

[8] R. Kaur, A. Tripathi, *Dynamic state general fuzzy automata under semantically similar environment: An application-based study on air pollution for smart cities 4.0*, New Mathematics and Natural Computation, (2025), 1-24. https://doi.org/10.1142/S1793005727500189

[9] E. K. La Guardia, et al., *On semi-vector spaces and semi-algebras with applications in fuzzy automata*, Axioms, **13**(5) (2024), 308. https://doi.org/10.3390/axioms13050308

[10] E. T. Lee, L. A. Zadeh, *Note on fuzzy languages*, Information Sciences, **1**(4) (1969), 421-434. https://doi.org/10.1016/0020-0255(69)90025-5

[11] F. Lin, H. Ying, *Modeling and control of fuzzy discrete event systems*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **32**(4) (2002), 408-415. https://doi.org/10.1109/TSMCB.2002.1018761

[12] D. S. Malik, J. N. Mordeson, *Fuzzy discrete structures*, Studies in Fuzziness and Soft Computing, **58**, Physica-Verlag, New York, NY, USA, 2000. https://doi.org/10.1007/978-3-7908-1838-3

[13] J. N. Mordeson, D. S. Malik, *Fuzzy automata and languages: Theory and applications*, Chapman and Hall/CRC, Boca Raton, FL, USA, 2002. https://doi.org/10.1201/9781420035643

[14] D. W. Qiu, *Supervisory control of fuzzy discrete event systems: A formal approach*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, **35**(1) (2005), 72-88. https://doi.org/10.1109/TSMCB.2004.840457

[15] E. S. Santos, *Maximin automata*, Information and Control, **13**(4) (1968), 363-377. https://doi.org/10.1016/S0019-9958(68)90864-4

[16] E. S. Santos, *On reduction of max-min machines*, Journal of Mathematical Analysis and Applications, **40**(1) (1972), 60-78. https://doi.org/10.1016/0022-247X(72)90031-5

[17] M. Shamsizadeh, E. Movahednia, M. De la Sen, *Isomorphism between two vector general fuzzy automata*, Informatica, **34**(3) (2023), 617-633. https://doi.org/10.15388/23-INFOR514

[18] M. G. Thomason, P. N. Marinos, *Deterministic acceptors of regular fuzzy languages*, IEEE Transactions on Systems, Man, and Cybernetics, **SMC-4**(2) (1974), 228-230. https://doi.org/10.1109/TSMC.1974.5409123

[19] R. J. Wai, Y. W. Lin, *Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic Petri recurrent fuzzy neural network*, IEEE Transactions on Fuzzy Systems, **21**(4) (2013), 688-701. https://doi.org/10.1109/TFUZZ.2012.2227974

[20] W. Wechler, *The concept of fuzziness in automata and language theory*, Akademie-Verlag, Berlin, Germany, 1978. https://doi.org/10.1515/9783112733356

[21] W. G. Wee, *On generalizations of adaptive algorithm and application of the fuzzy sets concept to pattern classification*, Ph.D. Dissertation, Purdue University, West Lafayette, IN, USA, 1967.

[22] G. D. Wu, P. H. Huang, *A vectorization-optimization-method-based type-2 fuzzy neural network for noisy data classification*, IEEE Transactions on Fuzzy Systems, **21**(1) (2013), 1-15. https://doi.org/10.1109/TFUZZ.2012.2197754