A Novel Method to Compute the Membership Value of the States of Fuzzy Automata

Doru Todinca*, Ioana Şora*, Daniel-Eugen Butoianu* and Radu-Emil Precup†
*Department of Computers and Information Technology
Politehnica University Timisoara, Romania
Email: {doru.todinca, ioana.sora}@cs.upt.ro
†Dept of Automation and Applied Informatics
Politehnica University Timisoara
Email: radu.precup@aut.upt.ro

***Abstract*—In this work we propose a novel method to compute the membership values of the next states of a fuzzy automaton, using an averaging function between the membership value of the input, and the membership value of the current state. We validate our method by simulation.**

***Index Terms*—fuzzy logic, fuzzy automata, membership value**

I. Introduction

Fuzzy automata (FA) are fuzzy logic extensions of finite state machines, or classic automata. Since they have been proposed at the end of 1960's [1], [2], fuzzy automata have been applied in different domains like: medicine [3], [4], intelligent decision support [5], intelligent hybrid control systems [6], air quality monitoring [7], industrial applications [8], pedestrians' intention prediction [9], [10], telecommunication systems [11], [12], decision making in fuzzy environments [13], the study of emotional behaviour [13], fire detection and monitoring [14], behaviour-based control structures [15], human-robot interaction [16], etc.

The list of applications' domains of FA might seem long, but, as mentioned in [17], fuzzy automata have far less practical applications than crisp automata. A related aspect is that, if we look in the literature, the papers that present mathematical approaches to FA (e.g. [18], [19], etc) are more numerous than the papers that present practical applications of FA.

Our aim is to try to find the causes of this relative lack of practical applications of fuzzy automata and to provide solutions in order to increase the applicability of FA. In this work we continue our investigation on the behaviour of fuzzy automata. The problem that we have identified as the main drawback of fuzzy automata is related to an undesired behaviour that characterizes many of these FA: depending on the input sequences applied, the membership value of some, or of all states of the FA decreases towards a small value, even towards zero, and cannot be increased after that. We called this non-controlling behaviour [20].

In [20], [21], [22] we developed a VHDL framework for modeling and simulation of fuzzy automata and we used this framework to study the behaviour of different types of fuzzy automata and of different operators used for computing the membership value of the next state of the FA. Also, we investigated by simulation some of the methods proposed in literature in order to avoid non-controlling behaviour: state normalization [23] and conservation of state [24] and found that conservation of state is not very efficient, while state normalization is computationally intensive.

In this paper, which extends the work from [25], we propose a general solution for the problem of non-controlling behaviour. Our solution refers to the computation of the membership value of the next state of a fuzzy automaton, as an average between the membership value of the input of the FA and the membership value of the current state of the FA, taking into account also the value of the transition from the present state to the next state for the given input.

The paper is organized as follows: next section briefly describes fuzzy automata, section III presents in detail our method, while section IV shows the simulation results. The paper ends with a section of conclusions.

II. Fuzzy automata

Fuzzy automata are based on fuzzy sets. Fuzzy sets have been defined by L.A. Zadeh as extensions of classic, or crisp sets, as follows [26], [27]: given an universe of discourse X (a crisp set), a fuzzy set $\tilde{A} \subset X$ is defined as

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$$

where $\mu_{\tilde{A}} : X \to [0, 1]$ is called membership function.

Fuzzy sets can be continuous or discrete and each element of a fuzzy set has a membership value (mv). Zadeh [26] defined the intersection an union between fuzzy sets as minimum and respectively maximum between their membership functions. The minimum and maximum operators have been extended to classes of operators called t-norms (for intersection) and s-norms, or t-conorms, for union. More details about t-norms and s-norms can be found e.g. in [27], chapter 3, section 3.2.

We consider here only discrete fuzzy automata, which are fuzzy extensions of classic discrete automata. The inputs and the states of a fuzzy automaton are fuzzy sets in the universes of discourse U and X, respectively. We do not discuss the outputs of the FA here.

When input u_j is applied to a fuzzy automaton at time t and if there is a transition from the current state x_k to next

978-1-5386-4640-3/18/$31.00 ©2018 IEEE

state x_i, the membership value $\mu_x(x_i)$ of the next state x_i will be computed based on the membership value $\mu_u(u_j)$ of the intput u_j, the membership value $\mu_x(x_k)$ of the current state, and on the transiton's weight $\mu_x(x_i|x_k, u_j)$.

According to [24], the membership function of the next state will be:

$$\mu_x(x_i) = \sum_{x_k} \sum_{u_j} [\mu_x(x_t) \times \mu_u(u_j) \times \mu_x(x_i|x_k, u_j)]$$

In [24] $\times$ represents algebraic product and $\sum$ represents algebraic sum, but other pairs of t- and s-norms can be used.

III. Our method

A. The membership assignment function

When computing the membership value $\mu_x(x_i)$ of the next state, most FA use the minimum operator between the mv's $\mu_x(x_k)$, $\mu_u(u_j)$ and $\mu_x(x_i|x_k, u_j)$, but this means that, if either the input u_j, or the state x_k has a small mv, then the next state will have also a small value. In a fuzzy automaton it is possible that the next state x_i will be reached by more than a single transition, form other present states and/or for other inputs. However, the application of the minimum operator in the computation of the mv of the next state means that, if all the present states have small mv's, then the mv of the next state cannot be increased by a high value of mv of an input, which is, in our opinion, a counter-intuitive behaviour and the main cause for the uncontrolling behaviour of most FA.

Different researchers have applied other operators instead of minimum, e.g., algebraic product in [24], or other t-norms. The problem with the t-norms is that they behave in a similar way like minimum, and moreover, they will not produce a result bigger than the result obtained by using minimum, since minimum is the biggest t-norm.

A remarkable contribution to this problem was presented by Doostfatemeh and Kremer, in [25]. They propose to use for computing the mv of the next state, a function denoted F_1 and called *membership assignment functions*, defined as: $F_1 : [0, 1] \times [0, 1] \to [0, 1]$. The function F_1 satisfies two axioms (properties):

1) $0 \le F_1(x, y) \le 1$
2) $F_1(0, 0) = 0$, $F_1(1, 1) = 1$

It means that F_1 should not necessary be a t-norm, but it can be, for example, an averaging function, like arithmetic mean or geometric mean [25], which is, in our opinion, a very important result.

But Doostfatemeh and Kremer assumed that the inputs of the FA are crisp, which means that their membership values will be 1 for an active input, and they applied the function F_1 only between the membership value of the current state and the transition weight, hence the mv of the input does not influence the computation of the mv of the next state.

In our previous work [20], [21], [22], we applied F_1 considering also the mv of the input, such that we computed the mv of the next state as $F_1(F_1(\mu_u(u_j), \mu_x(x_k)), \mu_x(x_i|x_k, u_j))$, but we used only t-norms for the function F_1.

In this work we propose a new interpretation of the way we compute the mv of the next state of a fuzzy automaton. We consider that, if the mv $\mu_x(x_k)$ of the present state x_k is small, but the mv, $\mu_u(u_j)$ of the input u_j if big, then the mv of the next state should be bigger than the mv of the current state, i.e., $\mu_x(x_i) \ge \mu_x(x_k)$. This can be easily achieved if we compute the mv of the next state as an average between the mv of the input and the mv of the current state, but cannot be achieved by using the $\min$ operator (or another t-norm) between $\mu_u(u_j)$ and $\mu_x(x_k)$.

We didn't discuss yet the weight of the transition between the present state x_k and next state x_i. We assume that many fuzzy automata will be fuzzy extensions of crisp automata, i.e., they will have the same transitions like the crisp automata, but the inputs will be fuzzy sets. This means that, if a transition is active, its weight will be 1. If we apply an averaging function again for the computation of the mv of the next state, then we will unnecessarily increase the mv of the next state. For example, if we use arithmetic mean for F_1, then the mv of the next state will be the average between 1 (the weight of the transition) and the average of $\mu_x(x_k)$ and $\mu_u(u_j)$, obtaining a value which is always ≥ 0.5. We consider that the minimum function is better suited here because, if the weight of the transitions will be crisp (1 for the existing transition and 0 when there is no transition), then the mv of the next state will be the average between the mv of the input and the mv of the current state. If the transitions are not crisp, but fuzzy, (which means, between 0 and 1), then the minimum will limit the mv of the next state.

The formula that we propose for computing the mv $\mu_x(x_i)$ of the next state x_i for a transition from state x_k determined by the input u_j is:

$$\mu_x(x_i) = \min(F_1(\mu_u(u_j), \mu_x(x_k)), \mu_x(x_i|x_k, u_j)) \quad (1)$$

where F_1 is an averaging function like arithmetic mean, geometric mean, or weighted average $w\_avg(x, y, w) = w \cdot x + (1 - w) \cdot y$, where $w \in [0, 1]$ is the weight. By using the weight w we can increase or decrease the influence of $\mu_u(u_j)$ in the value of $\mu_x(x_i)$.

B. Multi-membership resolution

In a FA it is possible to be simultaneous transitions to the same state x_i, from different states and/or due to different inputs. This can happen because the inputs are fuzzy sets, which means that several inputs can be active at the same time, but in different degrees (i.e., having different membership values). Also, it it possible to have several states of the fuzzy automaton active (in different degrees) at the same time t.

If there are several paths that conduct to the same next state x_i, each path providing a membership value for $\mu_x(x_i)$ (value computed according to the formula (1)), how to compute a single mv for x_i? What happens if some paths give a small value and some path a big value?

Doostfatemeh and Kremer [25] called this problem *multi-membership resolution*, and they defined a multi-membership resolution function, denoted F_2, for which they gave a set of three axioms. Again, there are many functions that satisfy these axioms, among which the s-norms (including maximum), arithmetic mean, etc.

If we use an averaging function for F_2, like arithmetic mean, then the paths that provide a small value will "weaken" the strongest path, i.e., the path that gives the highest value for $\mu_x(x_i)$, which we consider undesirable. The most intuitive choice would be to use the maximum function for F_2, which is, the value of the mv of the next state will be determined by the strongest path that leads to that state. In this work we adopt this approach.

A possible drawback of the maximum function used for F_2 is that, it is no difference if there is only one path towards a state, or many paths, since only the strongest path counts. If we use other s-norms instead of maximum, they will give a final result bigger than the maximum, since the maximum is the smallest s-norm. We believe that our results from [20], [21], [22], where some FA using other pairs of s- and t-norms than $\max$ and $\min$ obtained better results than the max-min FA, can be explained mostly because the s-norm used instead of maximum gave bigger values for the mv of the next state than in case of using maximum.

In this work we will use only maximum for F_2, because we believe that it is easier and more intuitive to control the mv of the next state from mv of the input, than by combining the different paths that conduct to the same next state. Depending on the applications, other functions may be used for F_2, if necessary.

The formula that we propose for computing the mv $\mu_x(x_i)$ of the next state x_i, considering all possible transitions is:

$$\mu_x(x_i) = \max_{x_k}[\max_{u_j}[\min(F_1(\mu_u(u_j), \mu_x(x_k)), \mu_x(x_i|x_k, u_j))]] \quad (2)$$

C. The averaging function used for F_1

The problem encountered by the weighted average function $w \cdot x + (1 - w)$ presented in subsection III-A, is given by the fact that, if $\mu_u(u_j) = 0$, or $\mu_x(x_k) = 0$, but not both of them are zero, then the resulted value of the weighted average will not be zero. This is a problem, since we use the value 0 for $\mu_u(u_j)$ or $\mu_x(x_k)$ to show that the input u_j, or respectively the state x_k, is not active at that moment. Hence, we have to modify the formula for the weighted average function as follows:

$$w\_avg(x, y, w) = \begin{cases} 0, \text{ if } x = 0 \text{ or } y = 0 \\ w \cdot x + (1 - w) \cdot y, \text{ if } x, y > 0 \end{cases}$$

,where $w \in [0, 1]$.

For $F_1(\mu_u(u_j), \mu_x(x_k), w)$ we obtain:

$$F_1 = \begin{cases} 0, \text{ if } \mu_u(u_j) = 0 \text{ or } \mu_x(x_k) = 0 \\ w \cdot \mu_u(u_j) + (1 - w) \cdot \mu_x(x_k), \text{ otherwise} \end{cases}$$

,where $w \in [0, 1]$.

Another possibility would be to use for F_1 the geometric mean $\sqrt{x \cdot y}$, resulting $F_1 = \sqrt{\mu_u(u_j) \cdot \mu_x(x_k)}$.

In this work we implemented both formulas and we will show the simulation results obtained.

IV. Results

A. The first set of simulations

We have simulated a fuzzy automaton, whose state table is given in table I. The state table gives the next state of the FA for all combinations of present state and current input. The system has four states: INIT, S (small), M (medium), and L (large), and three inputs: small number (SN), medium number (MN) and large number (LN). From the INIT state, the system goes to state S if we apply a small number at the inputs, to state M for a medium number and to state L for a large number. From state S it goes to state M if a MN is applied, to INIT if a LN is applied, and it remains in state S if a SN is applied. The other transitions can be easily understood from table I.

TABLE I
The state table of the first fuzzy automaton

Present state	**Inputs**		
	SN	***MN***	***LN***
INIT	S	M	L
S	S	M	INIT
M	S	M	L
L	INIT	M	L

Inputs SN, MN and LN are fuzzy sets, presented in figure 1. We can see that a number situated in the interval $[0, 100]$ belongs to at least one of these fuzzy sets, but, in most cases, it will belong to two of the mentioned fuzzy sets since the sets are strongly overlapping. In order to increase the readability, we scaled the interval of the membership values from $[0, 1]$ to $[0, 100]$ and we did the same in the next figures, for all fuzzy sets. In figure 1 and in the following figures we represented on the abscissa the simulation time in simulation cycles (clock cycles), and on the ordinate the membership functions (values) of the fuzzy sets, scaled to the interval $[0, 100]$

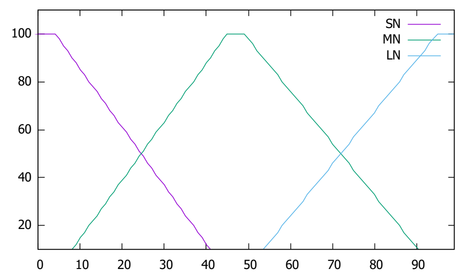

Fig. 1. The inputs SN, MN and LN.

In order to test the functionality of the fuzzy automaton we used our VHDL framework, and we applied a test sequence to the FA with the same structure, but with different operators. We used w_avg operator with weights w increasing from 0 to 1 in steps of 0.1, each operator being applied to a fuzzy automaton. Another FA uses the geometric mean. For comparison we used also the following pairs of t- and s-norms in other fuzzy automaton: algebraic, drastic, bounded and nilpotent product and sum, and $\min$ and $\max$.

We incremented a number between 0 and 300 modulo 100 in steps of 1, such that it activated the fuzzy input sets SN, MN and LN in different degrees, and applied this sequence to all fuzzy automata.

In the following figures we present the evolution of states for different fuzzy automata. Figure 2 shows the evolution of the states of the FA which uses the w_avg function with $w = 0.5$. We see that all states have membership values from 0 to 100 and, even after the membership value of a fuzzy set reaches the value 0, it can be increased again. Figure 3 presents a detail of fig 2, only for the first 100 simulation steps, and only for the states S, M and L (without INIT).

Figure 4 shows the evolution of the states of the FA that uses the geometric mean, and it is quite similar with figure 2, in the sense that the mv of all states can be increased again after they reach the value 0.

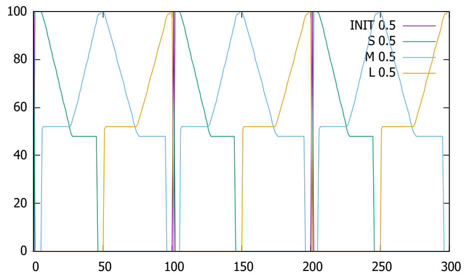

Fig. 2. The states of the automaton using w_avg with $w = 0.5$

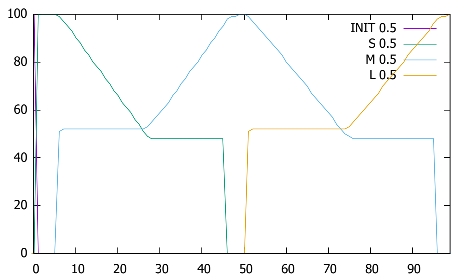

Fig. 3. States S, M and L of the automaton using w_avg with $w = 0.5$, detail

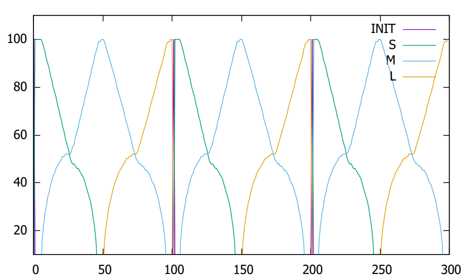

Fig. 4. The states of the automaton using geometric mean

For comparison, we represent the states of the max-min automaton in figure 5. We can see that the mv's of the states decrease after some time, and then they can be increased again, but not above 0.5 (above 50 in fig 5). The nilpotent automaton behaves in a similar way like the max-min automaton. A much worse behaviopur is exhibited by the other automata. We exemplify it for the algebraic automaton, shown in figure 6. As we can see, the mv's of **all** states of this FA decrease to zero and cannot be increased after that. This is a totally undesired behaviour. It is worth mentioning that all automata using pairs of s- and t-norms use also the conserving transition method (proposed in [24]) for computing the mv of the next state. This method implies the use also of negated inputs in order to avoid the decrease of thr mv of their states (see [24] for more details).

If the conserving transition method is not applied, all FA using s- ans t-norms will have the mv of their states zero after a number of simulation steps, except the max-min automaton. But even for max-min automaton, the mv's of the states will be smaller than in the case when conserving transition is applied.

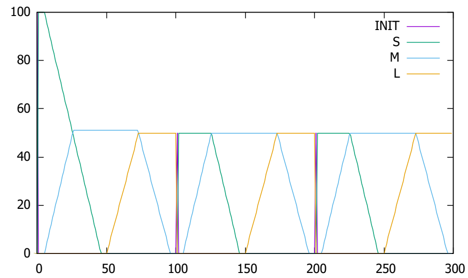

Fig. 5. The states of the automaton using max and min

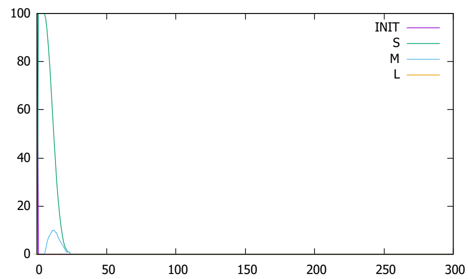

Fig. 6. The states of the automaton using algebraic product and sum

Figure 7 allows a comparison between the shape of the states (for clarity, only state M is represented) of different automata using the w_avg function with the weights $w = 0.1$, $w = 0.5$ and $w = 0.9$, and the automaton using the geometric mean. We can see that the boundaries of the states are very similar, only the shape of the state M is different.

B. The second set of simulations

This second set of simulations uses the same automaton like in section IV-A, but the shapes of the input fuzzy sets are different. We can see the input fuzzy sets in figure 8 and we notice that their overlap is smaller than in the first set of simulations.

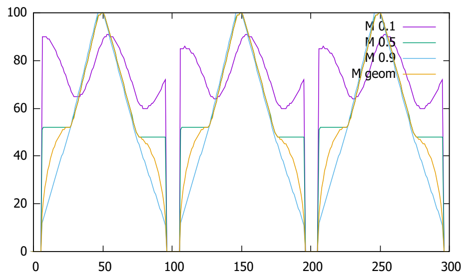

Fig. 7. Comparison of the M states of automata with w_avg and weights 0.1, 0.5, 0.9, and automaton with geometric mean

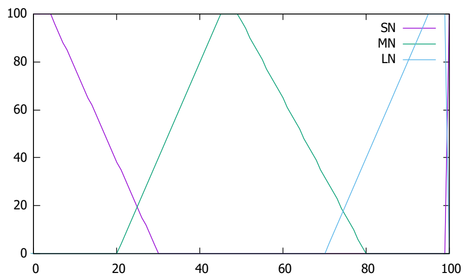

Fig. 8. The inputs SN, MN and LN in the second case.

Figures 9, 10 and 11 present the evolution of the states for fuzzy automata which use w_avg and $w = 0.5$ (in fig 9), geometric mean in fig 10, and $\max$ and $\min$ in fig 11. We can see that the mv of the states take all values in the interval $[0, 100]$ for the FA using w_avg and geometric mean, but, for the max-min automaton, their values do not increase above 20, where for the first simulations, they do not increase above 50. Hence, the shape of the inputs has a big influence on maximum values that can be attended by the mv's of the states of max-min automaton, but their influence is very small for the maximum values that can be attended by the mv's of the automata which use our method.

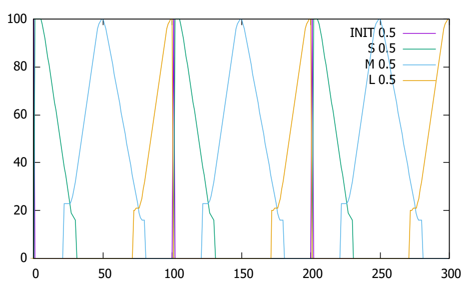

Fig. 9. The states of the automaton using w_avg with $w = 0.5$, second case

C. Other simulations

We have simulated a second fuzzy automaton, also with four states, which emulates the states of a patient. Its state table is given in table II. The FA has the states VB (very bad), B (bad), good (G) and very good (VG), and the system goes to a better

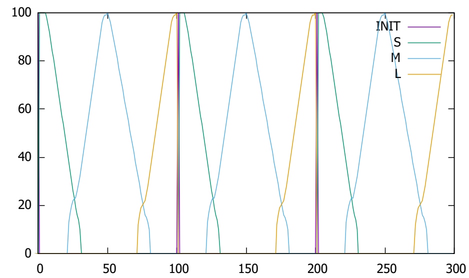

Fig. 10. The states of the automaton using geometric mean, second case

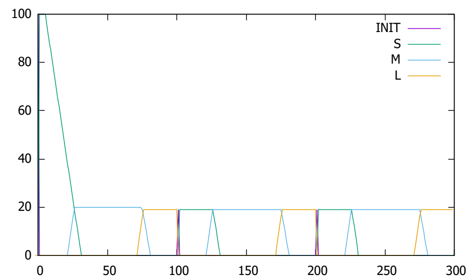

Fig. 11. The states of the automaton using $\max$ and $\min$, second case

or to a worse state based on a monitored (input) parameter. If the input parameter has the value W (worse), then the FA goes to a worse state. On the contrary, if the input parameter has the value I (improve), then the state of the FA goes to a better one. If the parameter has the value S (stable), the state remains unchanged. The results obtained are very similar to the results obtained for the first system (i.e, the mv of the states can be increased after it had a small value), and we do not present them here due to space limitations.

TABLE II
THE STATE TABLE OF THE SECOND FUZZY AUTOMATON

Present state	Inputs		
	W	***S***	***B***
VB	VB	VB	B
B	VB	B	G
G	B	G	VG
VG	G	VG	VG

A more challenging test was to apply averaging function to the FA described by Reyneri in [28]. This FA describes a circular trajectory. Its implementation is detailed in [21]. The challenge consists of the fact that the FA from [28] was designed for max-min automata. We did not change anything in the system (e.g., the transition matrix, the membership functions of the inputs, etc), except the method to compute the mv of the next state.

When we used w_avg with $w = 0.5$ the trajectory obtained was not the expected one. Looking into more details at the state evolution, we observed that in the case of w_avg the mv of the next state increased too fast, compared to the system from [21]. Hence, we decided to decrease the weight of the mv of the input, and after that, the FA worked well for $w \leq 0.3$. Figure 12 presents the trajectory obtained by the FA using our method and $w = 0.3$ compared with the trajectory obtained by the max-min automaton.

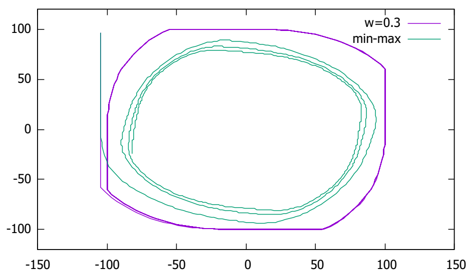

Fig. 12. Circular trajectory with automaton using w_avg with $w = 0.3$. Comparison with original max-min automaton.

V. Conclusions

In this work we proposed a new method to compute the membership value of the next state of a fuzzy automaton. We believe that our method solves the problem of noncontrolling behaviour of fuzzy automata, increasing their practical applicability. We explained the motivation behind the proposed formula (2) for computing the mv of the next state.

Our method permits a much better control of the mv of the next state using the mv of the inputs, compared to other existing methods from literature. In this way we can avoid the situation when the mv of the states of a fuzzy automaton reach small values and cannot be increased after that. We validated our method by simulation.

In the future we want to test our method using more examples of fuzzy automata, including those reported for signatures [29], [30], [31].

References

[1] E. S. Santos, "Maximin automata," *Information and Control*, vol. 13, no. 4, pp. 363–377, 1968.

[2] E. S. Santos and W. G. Wee, "General formulation of sequential machines," *Information and control*, vol. 12, no. 1, pp. 5–10, 1968.

[3] F. Steimann and K.-P. Adlassnig, "Clinical monitoring with fuzzy automata," *Fuzzy Sets and Systems*, vol. 61, pp. 37–42, 1994.

[4] K.-P. Adlassnig, "Fuzzy methods in clinical research and patient care," *Hepatology*, vol. 9, p. 10, 2016.

[5] J. L. Grantner, S. T. Fuller, and J. Dombi, "Fuzzy automaton model with adaptive inference mechanism for intelligent decision support systems," in *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2377–2384.

[6] J. Grantner and G. Fodor, "Fuzzy automata for intelligent hybrid control systems," in *Proceedings of the 2002 IEEE International Conference FUZZ-IEEE'02*, vol. 2, May 12-17 2002, pp. 1027–1032.

[7] J. Boubeta-Puig, A. Camacho, L. Llana, and M. Núñez, "A formal framework to specify and test systems with fuzzy-time information," in *International Work-Conference on Artificial Neural Networks*. Springer, 2017, pp. 403–414.

[8] J. Adamy, "Breakout prediction for continuous casting by fuzzy mealy automata," in *Proc. 3rd European Congress on Intelligent Techniques and Soft Computing EUFIT*, vol. 95, 1995, pp. 754–759.

[9] J.-Y. Kwak, E.-J. Lee, B. Ko, and M. Jeong, "Pedestrians intention prediction based on fuzzy finite automata and spatial-temporal features," *Electronic Imaging*, vol. 2016, no. 3, pp. 1–6, 2016.

[10] J.-Y. Kwak, B. C. Ko, and J.-Y. Nam, "Pedestrian intention prediction based on dynamic fuzzy automata for vehicle driving at nighttime," *Infrared Physics & Technology*, vol. 81, pp. 41–51, 2017.

[11] C.-L. Chen and P.-C. Hsiao, "Supporting QoS in wireless MAC by fuzzy logic control," in *Proceedings of IEEE Wireless Communications and Networking Conference WCNC'05*, vol. 2, New Orleans, USA, March 2005, pp. 1242–1247.

[12] A. T. Jahromi, P. Karimaghaee, S. G. Haghighi, M. Doostfatemeh, and M. J. Er, "Intelligent selective packet discarding using general fuzzy automata," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 3714–3720.

[13] L. He, W. Kinsner, and N. Sepehri, "Decision-making in fuzzy environments using ontological control with fuzzy automata," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*. IEEE, 2010, pp. 542–549.

[14] B. C. Ko, S. J. Ham, and J. Y. Nam, "Modeling and formalization of fuzzy finite automata for detection of irregular fire flames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 12, pp. 1903–1912, 2011.

[15] S. Kovács and L. T. Kóczy, "Application of interpolation-based fuzzy logic reasoning in behaviour-based control structures," in *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 1543–1548.

[16] S. Kovács, D. Vincze, M. Gácsi, Á. Miklósi, and P. Korondi, "Fuzzy automaton based human-robot interaction," in *Applied Machine Intelligence and Informatics (SAMI), 2010 IEEE 8th International Symposium on*. IEEE, 2010, pp. 165–169.

[17] J. Virant and N. Zimic, "Fuzzy automata with fuzzy relief," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 1, pp. 69–74, february 1995.

[18] R. Verma and S. Tiwari, "Distinguishability and completeness of crisp deterministic fuzzy automata," *Iranian Journal of Fuzzy Systems*, vol. 14, no. 5, pp. 19–30, 2017.

[19] H. Pan, Y. Li, Y. Cao, and P. Li, "Nondeterministic fuzzy automata with membership values in complete residuated lattices," *International Journal of Approximate Reasoning*, vol. 82, pp. 22–38, 2017.

[20] D. Todinca and D. Butoianu, "VHDL framework for modeling fuzzy automata," in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on*. IEEE, 2012, pp. 171–178.

[21] D.-E. Butoianu and D. Todinca, "Performance evaluation of fuzzy automata using vhdl simulation," in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on*. IEEE, 2014, pp. 196–203.

[22] ——, "The efficiency of minimum operator for fuzzy automata: A case study," in *Applied Computational Intelligence and Informatics (SACI), 2015 IEEE 10th Jubilee International Symposium on*. IEEE, 2015, pp. 203–208.

[23] T. Watanabe, M. Matsumoto, and M. Enokida, "Synthesis of synchronous fuzzy sequential circuits," in *Proceedings of Third IFSA Word Congress*, 1989, pp. 288–291.

[24] Y. Mori, K. Otsuka, and M. Mukaidono, "Properties of fuzzy sequential circuit using fuzzy transition matrix and their design method," in *Fuzzy System 1995, Proceedings of 1995 IEEE International Conference on*, vol. 4, Yokohama, Mar. 20–24, 1995, pp. 2133–2138.

[25] M. Doostfatemeh and S. Kremer, "New directions in fuzzy automata," *International Journal of Approximate Reasoning*, vol. 38, pp. 175–214, 2005.

[26] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[27] H.-J. Zimmermann, *Fuzzy Set Theory – and Its Applications, Second, Revised Edition*. Kluwer Academic Publishers, 1991.

[28] L. M. Reyneri, "An introduction to fuzzy state automata," in *Biological and Artificial Computation: From Neuroscience to Technology*. Springer, 1997, pp. 273–283.

[29] L. Kóczy, T. Vámos, and G. Biró, "Fuzzy signatures," *Proceedings of Eurofuse-SIC*, vol. 99, pp. 25–28, 1999.

[30] Á. Ballagi, L. T. Kóczy, and T. Gedeon, "Robot cooperation without explicit communication by fuzzy signatures and decision trees." in *IFSA/EUSFLAT Conf.*, 2009, pp. 1468–1473.

[31] C. Pozna, N. Minculete, R.-E. Precup, L. T. Kóczy, and Á. Ballagi, "Signatures: Definitions, operators and applications to fuzzy modelling," *Fuzzy Sets and Systems*, vol. 201, pp. 86–104, 2012.