

Traffic Signal Control with Adaptive Fuzzy Coloured Petri Net Based on Learning Automata

S. Barzegar, M. Davoudpour, MR. Meybodi, A. Sadeghian, M. Tirandazian

Abstract – Increasing number of vehicles, as the natural consequence of population growth, has caused a significant bottle-neck in transportation network and consequently major delays at intersections. Hence, in this paper we study a hybrid adaptive model, based on combination of Coloured Petri Nets, Fuzzy Logic and Learning Automata to efficiently control traffic signals. We show that in comparison with the results found in the literature the vehicle delay time is drastically reduced using the proposed method.

Keywords: Adaptive Coloured Petri Nets; Fuzzy Logic; Learning Automata; Traffic Signal Control

I. INTRODUCTION

Coloured Petri Net (CPN) is a tool by which validation of discrete-event systems are studied and modeled. CPNs are used to analyze and obtain significant and useful information from the structure and dynamic performance of the modeled system. Coloured Petri Nets mainly focus on synchronization, concurrency and asynchronous events [1]. The graphic features of CPNs specify the applicability and visualization of the modeled system. Furthermore, synchronous and asynchronous events present their prioritized relations and structural adaptive effects. The main difference between CPNs and Petri Nets (PN) is that in CPNs the elements are separable but in PNs they are not. Coloured indicates the elements specific feature. The relation between CPNs and ordinary PNs is analogous to high level programming languages to an assembly code (Low level programming language). Theoretically, CPNs have precise computational power but practically since high level programming languages have better structural specifications, they have greater modelling power.

Siamak Barzegar is with the Department of Electronic and Computer Engineering, Islamic Azad University of Qazvin, Qazvin, Iran (email: Barzegar@qazviniau.ac.ir)

Maryam Davoudpour is with the Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada (email: mdavoud@scs.ryerson.ca)

Mohammad Reza Meybodi is with the Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran (email: mmeybodi@aut.ac.ir)

Alireza Sadeghian, Senior Member IEEE, is with the Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada (email: asadeghi@ryerson.ca)

Mehrdad Tirandazian is with the Department of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada (email: tirandaz@scs.ryerson.ca)

CPN's drawback is its non-adaptivity [2] therefore it is not possible to access the previous information available in CPNs. If there is more than one transition activated then each transition can be considered as the next shot. This Coloured Petri Net's characteristic indicates that since several events occur concurrently and event incidences are not similar, then when events occur they do not change by time, and this phenomenon is in contrast with the real and dynamic world. Simulation would be similar to execution of the main program. Our Purpose is to use the simulated model for analyzing the performance of the systems, as a result here the system problems and the weak points would be identified. However, classic CPN tools can do nothing to improve and solve problems and also it would not be possible to predict the next optimized situation.

In this Paper we present an Adaptive Fuzzy Coloured Petri Net based on Learning Automata. Using the information from previous states of the system and reactions of the dynamic environment, Adaptive Fuzzy Coloured Petri Net will predict the next optimized situation. This will update the current state of the system and will change and activate the probability of occurrence in time. The performance of the reaction on the systems in the dynamic environment will significantly help the Fuzzy Coloured Petri Nets to learn and get trained. In this paper we have used CPN tools for Fuzzy CPNs simulation.

Furthermore, we have studied the application of adaptive model mentioned in the previous paragraph to control traffic signals. The adaptive model is represented to optimize scheduling of traffic signals across intersections. Optimizing the methods used to control the traffic signals is significant, since it reduces air pollution, fuel consumption and improves time efficiency. The proposed algorithm uses the learning automata to adjust fuzzy functions, defined in input parameters of the problem.

Fuzzy logic was first used in traffic control systems by Pappis and Mamdani [3]. They simulated an isolated intersection composed of two one-way streets with no turns. Later, Niittymaki and Pursula [4] also simulated an isolated intersection where the proposed fuzzy logic controller led to a shorter vehicle delay and fewer stops. Niittymaki and Kikuchi [5] developed a fuzzy logic algorithm to control signals for pedestrians. Through simulation it was shown that their algorithm performed better than the conventional one. Chiu [6] used fuzzy reasoning to control multiple intersections with no turns. Fuzzy rules were used to adjust cycle time, phase split and offset parameters. Later Niittymaki [7] introduced a simple two-phase fuzzy signal controller. It was shown that the fuzzy

logic controller performed better than vehicle-actuated controller.

In our test model, we would study an intersection which has two main directions namely, North-South and East-West [8] with a traffic signal having 8 transitional phases as shown in Fig. 1:

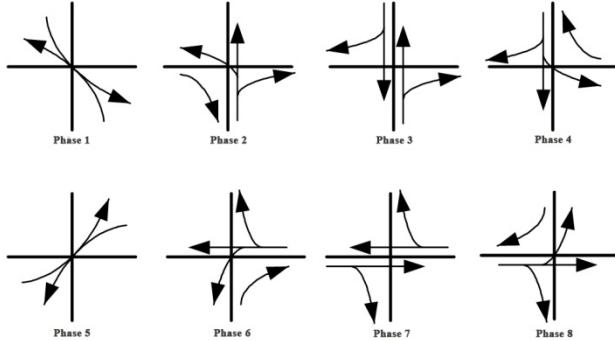


Figure 1. Eight- phase transition model.

Controlling traffic signals could be considered as an event discrete systems, which represent the level of synchronization and concurrency. Since the main advantages of PNs are observing synchronization and concurrency, thus the existing PNs are one of the best choices for simulation, analysis and evaluation of urban traffic Network. Therefore, we have used CPNs as a tool [1], [9].

The rest of the paper is structured as follows. In section II, we present learning automata as a basic learning strategy used in our proposed method. In section III, Fuzzy CPNs are briefly introduced. The proposed algorithm is introduced in Section IV. Section V explains the analysis of proposed algorithm and its comparison with the results of fuzzy algorithms used for scheduling traffic signals. And in the end we have concluding remarks.

II. LEARNING AUTOMATA

Learning automata is an abstract model which randomly selects one action out of its finite set of actions and evaluates it on a random environment, then again evaluates the same action and responds to the automata with a reinforcement signal. Based on this action, and received signal, the automata updates its internal state and selects its next action. Fig. 2 illustrates the relationship between an automata and its environment.

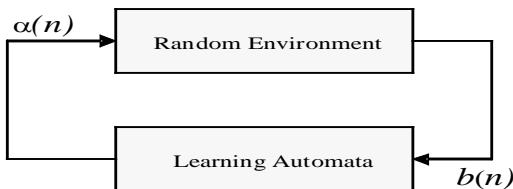


Figure 2. Relationship between learning automata and its environment

The environment can be defined by $E = \{a, b, c\}$ where $a = \{a_1, a_2, \dots, a_r\}$ represents a finite input set, $b = \{b_1, b_2, \dots, b_r\}$

represents the output set, and $c = \{c_1, c_2, \dots, c_r\}$ is a set of penalty probabilities, and each element c_i of c corresponds to one input of action a_i . An environment in which b can take only binary values 0 or 1 is called P-model environment. Also, by further generalization of the environment it is possible to have finite output sets with more than two elements that take values in the interval [0, 1]. Such an environment is called Q-model environment. Finally, when the output of the environment has continuous random variables, and assumes values in the interval [0, 1], then this environment is known as a S-model environment. Learning automata is classified into stochastic fixed-structure, and stochastic variable-structure. In the following, we only consider variable-structure automata.

A variable-structure automaton is defined by the quadruple $E = \{a, b, p, T\}$ in which $a = \{a_1, a_2, \dots, a_r\}$ is a set of actions (or outputs of the automaton). The output or action of an automaton is an instant of n denoted by $a(n)$, which is an element of the finite set $a = \{a_1, a_2, \dots, a_r\}$. $b = \{b_1, b_2, \dots, b_r\}$ represents the input set or response set, $p = \{p_1, p_2, \dots, p_r\}$ represents the action probability set, and finally $p(n+1) = T(a(n), b(n), p(n))$ represents the learning algorithm. The following shows, the operation of the automaton based on the action probability set p . The automaton randomly selects an action a_i , and performs it on the environment. After receiving the environment's reinforcement signal, the automaton updates its action probability set based on (1) for favorable responses, and (2) for unfavorable ones.

$$p_i(n+1) = p_i(n) + a.(1 - p_i(n)) \quad (1)$$

$$p_j(n+1) = p_j(n) - a.p_j(n) \quad \forall j \neq i$$

$$p_i(n+1) = (1 - b).p_i(n) \quad (2)$$

$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \quad \forall j \neq i$$

where a and b are reward and penalty parameters, respectively. If $a=b$, the automaton is called L_{RP} . If $b=0$ the automaton is called L_{RI} and if $0 < b < a < 1$, the automaton is called L_{RcP} . More information about learning automata can be found in [10].

III. FUZZY COLOURED PETRI NETS

Coloured Petri Nets were introduced by Kurt Jensen in 1987 as a developed model of Petri Nets. Coloured Petri Nets are appropriate tools for mathematical and graphical modeling. Coloured Petri Nets have numerous applications, and lots of research has taken place with respect to modeling, describing and analyzing systems, which have synchronized, asynchronized, distributed, parallel, non-deterministic or random natures. In fact, Petri Nets are models which could represent the performance and state of the system at the same time. There has been enormous research done in the following areas, (i) controlling and learning systems using coloured Petri Nets, (ii) optimizing Petri Net structures using genetic programming and (iii) learning and reasoning the ambiguous problems using fuzzy coloured Petri Nets. However, there are

no records for adapting coloured Petri Nets and using learning automata in Petri Net.

A formal definition of CPN is as follows [1]:

Definition 1: A Coloured PN (CPN) is a 6-tuple $CPN = (P, T, C, I^-, I^+, M_0)$ where:

1) $P = \{p_1, p_2, \dots, p_n\}$ denotes a finite and non-empty set of places,

2) $T = \{t_1, t_2, \dots, t_m\}$ denotes a finite and non-empty set of transitions, $P \cap T = \emptyset$,

3) C is a colour function that assigns a finite and non-empty set of colors to each place and a finite and non-empty set of modes to each transition.

4) I^- and I^+ denote the backward and forward incidence functions defined by $P \times T$, such that $I^-(p, t), I^+(p, t) \in [C(t) \rightarrow C(p)]_{MS}, \forall (p, t) \in P \times T$ ².

5) M_0 denotes a function defined on P , describing the initial marking such that $M_0(p) \in C(p)_{MS}$

A formal definition of a FCPN is as follows [11]:

Definition 2: A generalized non-hierarchical Fuzzy Coloured Petri Net is defined as 12-tuple $FCPN = (\Sigma, P, T, D, A, N, C, G, E, \beta, f, I)$ where

$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_l\}$ denotes a finite set of non-empty types, called colour sets, where $l \geq 0$,

and $P = \{P_C, P_F\}$ denotes a finite set of places,

$P_C = \{pc_1, pc_2, \dots, pc_m\}$ denotes a finite set of places that model the dynamic control behaviour of a system, and is called control places, where $m \geq 0$,

$P_F = \{pf_1, pf_2, \dots, pf_n\}$ denotes a finite set of places that model the fuzzy production rules, and is called fuzzy places, where $n \geq 0$, and $P_C \cap P_F = \emptyset$,

$T = \{T_C, T_F\}$ denotes a finite set of transitions,

$T_C = \{tc_1, tc_2, \dots, tc_i\}$ denotes a finite set of transitions that are connected to and from control places, and is called control transition, where $i \geq 0$,

$T_F = \{tf_1, tf_2, \dots, tf_j\}$ denotes a finite set of transitions that are connected to or from fuzzy places, and is called fuzzy transition, where $j \geq 0$, and $T_C \cap T_F = \emptyset$,

$D = \{d_1, d_2, \dots, d_h\}$ denotes a finite set of propositions, $|P_D| = |D|$,

$A = \{a_1, a_2, \dots, a_k\}$ denotes a finite set of arcs, $k \geq 0$, and $P \cap T = P \cap A = T \cap A = \emptyset$,

² The subscript MS denotes multisets. $C(p)_{MS}$ denotes the set of all finite multisets of $C(p)$.

$N : A \rightarrow P \times T \times P$ denotes a node function, and it maps each arc to a pair where the first element is the source node and the second element is the destination node, the two nodes have to be of different kinds,

In: an input function that maps each node x , to the set of nodes that are connected by an input arc($x \rightarrow x$);

Out: an output function that maps each node x , to the set of its nodes that are connected to x by output arc($x \rightarrow x$);

$C : (PUT) \rightarrow \Sigma_{ss}$ is a colour function, which maps each place and transition to a super-set of colour sets,

$G : T \rightarrow \text{expression}$ which denotes a guard function, $\forall t \in T : [\text{Type}(G(t)) = \text{Boolean} \wedge \text{Type}(\text{Var}(G(t))) \subseteq \Sigma]$, where $\text{Type}(\text{Vars})$ denotes the set of types $\{\text{Type}(v) | v \in \text{Vars}\}$, Vars denotes the set of variables, and $\text{Var}(G(t))$ denotes the set of variables used in $G(t)$,

$E : A \rightarrow \text{expression}$ which denotes an arc expression function,

$\forall a \in A : [\text{Type}(E(a)) = C(p(a))_{MS} \wedge \text{Type}(\text{Var}(E(a))) \subseteq \Sigma]$ where $p(a)$ is a place in $N(a)$, and MS stands for multi-set,

$\beta : PF \rightarrow D$ denotes a bijective mapping from fuzzy places to a proposition,

$f : T \rightarrow [0,1]$ denotes an association function, which assigns a certainty value to each colour used in each fuzzy transition,

I : denotes an initialization of double (δ, α),

$\delta : P \rightarrow \text{expression}$ which denotes an initialization function, $\forall p \in P : [\text{Type}(\delta(p)) = C(p)_{MS}]$,

α : denotes an association function which assigns a certainty value in the range $[0,1]$ to each token in the fuzzy places.

The structure of Fuzzy Coloured Petri Nets depends on the fuzzy production rules. The composite fuzzy production rules could be distinguished into following three rule-types respectively [12], [13].

Type1: Simple fuzzy production rule:

If d_j THEN d_k (CF=u)

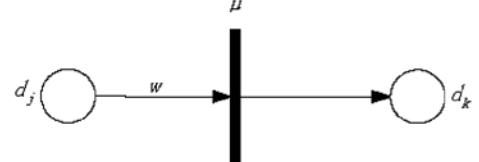


Figure 3. The FCPN denotation of fuzzy Coloured rule of Type 1

Type2: Compound joined fuzzy production rule:

If $d_1 \text{ AND } d_2 \text{ AND } \dots \text{ AND } d_n$ THEN d_k (CF = u)

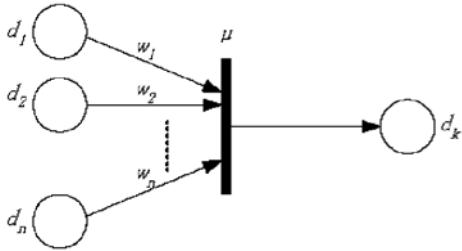


Figure 4. The FCPN denotation of fuzzy Coloured rule of Type 2

Type3: Compound disjoined fuzzy production rule:

If $d_1 \text{ OR } d_2 \text{ OR } \dots \text{ OR } d_n$ THEN $d_k (\text{CF} = u)$

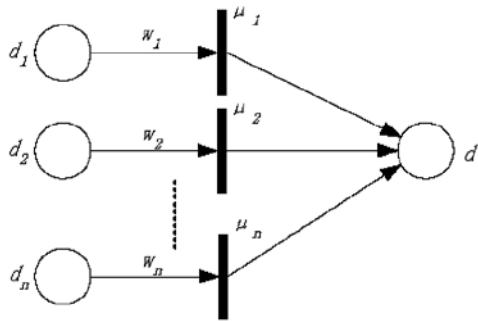


Figure 5. The FCPN denotation of fuzzy Coloured rule of Type 3

IV. THE PROPOSED ALGORITHMS

In this section we propose a combinative algorithm to control traffic signals across intersections. In this algorithm, fuzzy logic and learning automata are used for intelligent control of traffic signals. In the proposed algorithm learning automata is used to adjust membership functions of input and output parameters.

Before detailed description of the algorithm and the process of adjusting fuzzy membership functions used in learning automata, we would first explain the following topics: fuzzy logic, features for intelligent control of traffic signals, primitive selection of fuzzy membership functions and fuzzy rules, details of the proposed algorithm and modelling the proposed algorithm.

A. Fuzzy Logic System

Rule base fuzzy logic systems contain four components: rules, fuzzifier, inference engine and output processor (defuzzifier). In rule base there are collections of *If-Then* statements. After the information is gathered, the inputs should get fuzzified to become usable. The defuzzification is used for concurrency in the fuzzy system. Moreover, in the rule base the output variable of the *Then* part in each *If-Then* statement is a fuzzy amount and the output of different rules would not necessarily be the same. Thus, to drive to a result, fuzzy inference engine is designed, and with the defuzzification a crisp value of deciding variable would be calculated.

B. The Specifications to Intelligently Control Traffic Signals

The issue to control traffic signals could be divided into two parts: (1) To determine the priority of the phases, (2) To schedule the traffic signals. In this paper, we concentrate on the second part [14]-[16]. The following is the specifications considered for intelligent scheduling of traffic signal at an intersection: The average number of vehicles waiting to cross the intersection is denoted by (Q_C) (when light is green), The average number of vehicles waiting to cross the intersection is denoted by (Q_N) (when signal turns green in the next Phase), The average rate of vehicles crossing through the green signal is denoted by (AR_C, ER_C), the average rate of vehicles crossing through the green light in the next phase is denoted by (AR_N, ER_N), the state of traffic in the intersection against the traffic in the neighbourhood cross section is denoted by (SNe_C, SNe_N), predicting the state of traffic in a particular given time is denoted by Δt (within the next 10 minutes) on the basis of achieved information from the previous time (ES_C, ES_N). These features would be considered as inputs to control the traffic system. The final result expected from the overall system is to improve the traffic congestion by increasing the period between signal changes. Each feature is defined as fuzzy variable and each having a specific linguistic description. TABLE A shows the features and their corresponding linguistic descriptions. For the output specification which determines the increased duration of the green signal, value Zero, Few, Moderate and too many are considered in Fig.7.

TABLE A. FEATURES AND THEIR LINGUISTIC DESCRIPTIONS

Variables	linguistic descriptions			
Q_C, Q_N	Few	Moderate	Many	Too Many
AR_C, ER_C, AR_N	Short	Medium	High	
SNe_C, SNe_N	Worse	No Change	Better	
ES_C, ES_N	Worse	No Change	Better	

C. Primitive Selection of Fuzzy Membership Functions

For each linguistic description of a fuzzy variable, a membership function is considered. All the membership functions are equipped with a learning automata and a variable structure, which have the responsibility to regulate the fuzzy function parameters. Fuzzy membership functions are either triangular or trapezoidal in shape shown in Fig. 8. The beginning and the end parts of the membership functions are constant and pre-determined. The learning automata would regulate the center of membership functions, to achieve the best timing and controlling the traffic signal. In each automaton a number of actions (denoted by m) are defined. The probability to choose each action of the learning automata at the beginning of the learning process is determined by $\frac{1}{m}$.

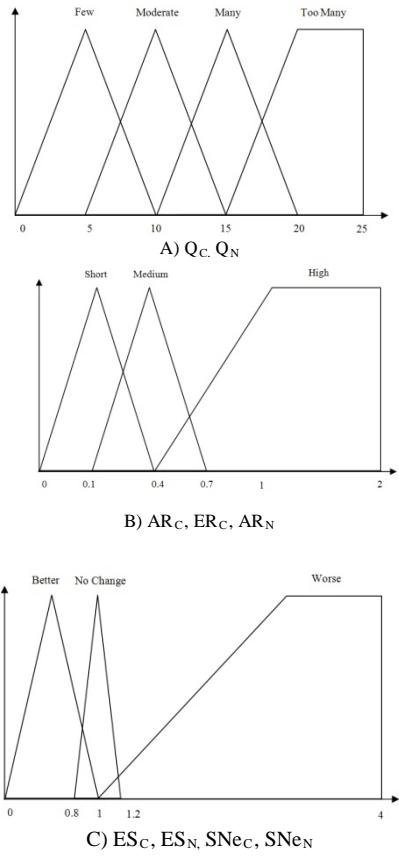


Figure 6. membership functions of input variables A) Q_C, Q_N - B) AR_C, ER_C, AR_N - C) ES_C, ES_N, SNe_C, SNe_N

The ratio of traffic in the neighbourhood intersections to the traffic in current intersection (SNe_C, SNe_N), and prediction of traffic in a specific period of time Δt (next 10 minutes) in selected phase, on the basis of gathered information from previous times (ES_C, ES_N) are calculated by the following formulas:

$$ES_C, ES_N \Rightarrow$$

$$\frac{\text{State of Average old Traffic in } t_C + \Delta t}{\text{State of Average Current Traffic}}$$

$$SNe_C, SNe_N \Rightarrow$$

$$\frac{\text{State of Traffic in Neighbor Intersections} + \text{Rest Time of Current Green Signal in Current Intersection} * b}{\text{State of Average Current Traffic} + \text{Rest Time of Current Green Signal in Neighbor Intersection} + a}$$

a is the required time for the vehicle to arrive from neighbourhood intersection to the detecting intersection and, its value is determined to be 7 seconds, and b is the probability of exit of vehicles from neighbourhood intersection to arrive to the mentioned intersection with value 0.8.

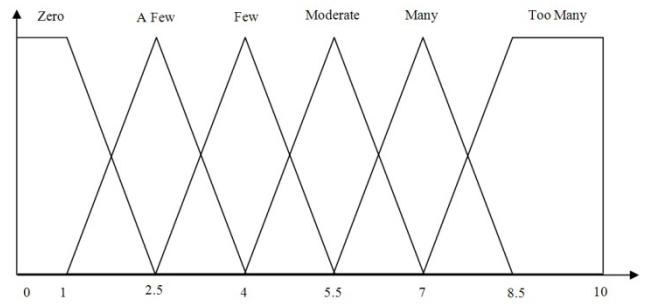


Figure 7. membership function in output variable

D. Fuzzy Rules

Some of the fuzzy production rules are as follows:

- 1) IF Q_C is Few AND Q_N is Too Many THEN EX is Zero
- 2) IF Q_C is Moderate AND Q_N is Few AND AR_C is Short AND AR_N is High AND ER_C is Medium THEN EX is Zero
- 3) IF Q_C is Few AND Q_N is Moderate AND AR_C is Short AND AR_N is short AND ER_C is Medium THEN EX is A Few
- 4) IF Q_C is Moderate AND Q_N is Too Many AND AR_C is Medium AND AR_N is Medium AND ER_C is High THEN EX is Few
- 5) IF Q_C is Moderate AND Q_N is Moderate THEN EX is Moderate
- 6) IF Q_C is Many AND Q_N is Many AND AR_C is Medium AND AR_N is short AND ER_C is Short AND SNe_C is Worse AND SNe_N is Worse AND ES_C is Better AND ES_N is No Change THEN EX is Many
- 7) IF Q_C is Many AND Q_N is Many AND AR_C is High AND AR_N is short AND ER_C is Medium AND SNe_C is Worse AND SNe_N is No Change AND ES_C is No Change AND ES_N is Worse THEN EX is Too Many
- 8) IF Q_C is Too Many THEN EX is Too Many

E. Details of the proposed Algorithm

To control a traffic signal in an eight phase intersection, we have used an algorithm, which is a combination of fuzzy logic methods and learning automata. The algorithm has two main parts: (i) To determine the green signal priority of phases and (ii) To increase the time of selected phase (extender). The minimum determined time for green light is 10 seconds, where the extender of the phase can increase that amount to 5 times more, which each time would be 0 to 10 seconds. As a result maximum time of a green signal would be determined as 60 seconds [17]. Each time the algorithm is performed one second before, the time is increased.

The performance of the algorithm is prioritized as follows:

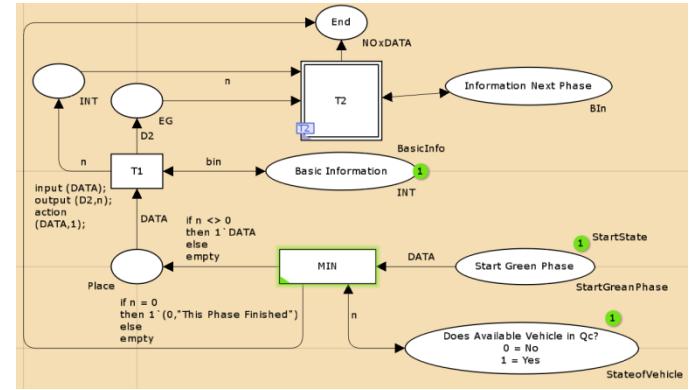
1. One phase out of eight phases is selected to make the traffic signal green.
2. The phase prioritizing section considers the priorities of the vehicles, which have emergency and then the average number of vehicles waiting to cross the intersection is calculated and then accordingly the signal switches to green. For example, in any phase if there exists a vehicle having an emergency state, then the traffic signal, in which

the vehicle is located, turns green and the previous phase having a green signal turns red. Accordingly, in this way the phase prioritizing section performs itself.

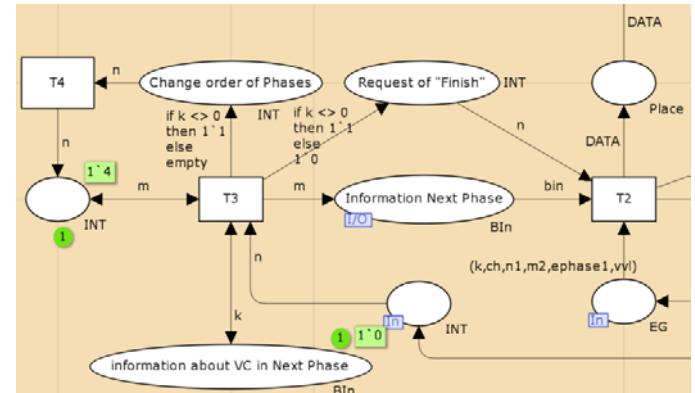
3. In a selected phase if there is no request for any vehicle to cross the intersection, then this phase is neglected and control goes to part 2.
 4. If there is a vehicle with an emergency state present, then the extender should increase the duration of green signal to enable the vehicle to exit the phase.
 5. Extender of green signal, in each learning automata would select one event considering the determined probability, therefore, as a result for all input parameters Q_N , Q_c , four membership functions and for all input parameters of E_{SC} , E_{SN} , SN_{cC} , SN_{eN} , A_{RC} , E_{RC} , A_{RN} , three membership functions would be created.
 6. The membership degree for each of the parameters AR_C , ER_C , AR_N , ES_C , ES_N , SNe_C , SNe_N , QC , QN , would be calculated considering the achieved information from the following: (i) The sensors, located at the intersections (ii) The traffic information related to prior periods, and (iii) Created membership functions in the previous part(5).
 7. Considering the amount of achieved membership for input parameters, and by activation of fuzzy rules, the output function would be determined.
 8. Considering the new traffic conditions in the intersections due to the increase of time in green signal, bonus or penalty is allotted to learning automata accordingly. Probability vectors of the learning automata of input parameters' membership functions are updated according to the following step:
 - If the Q_c Queue condition is worse than before, a penalty is assigned to the selected operation;

F. Modelling The Proposed Algorithm

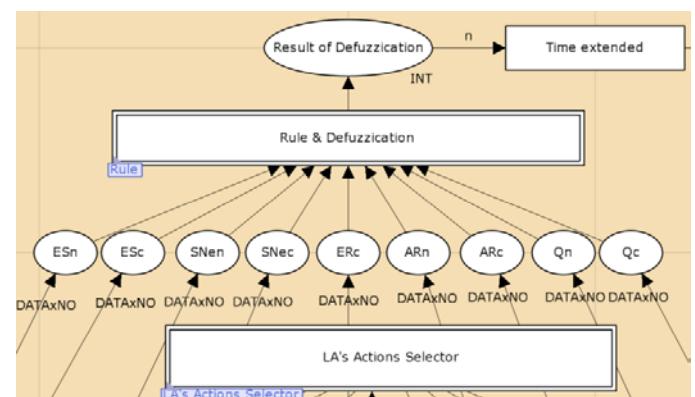
A model is created and shown in the following figure, which illustrates Coloured Petri Net tools for controlling the traffic [1], [18], [19].



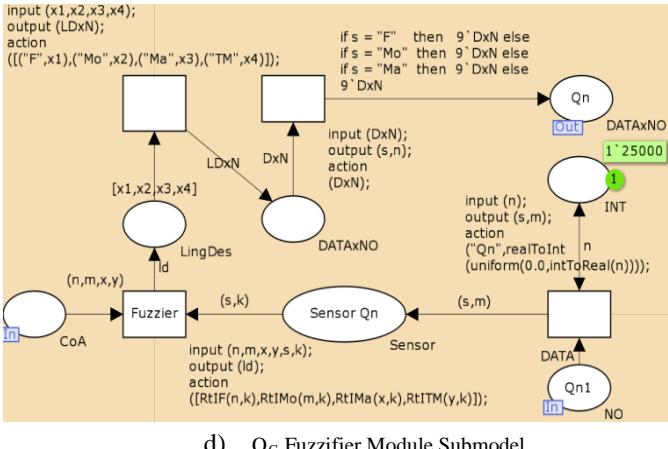
a) Main Model



b) Determining the phases' priorities Submodel



c) Rule & Defuzzification Submodel



d) Q_c Fuzzifier Module Submodel

Figure 8. a) Main Model b) Determining the phases' priorities Submodel, c) Rule and defuzzification submodel d) Q_c Fuzzifier Module Submodel

The created model consists of three main Modules:

- LA's Actions Selector Module
- Fuzzifier Module
- Rule and Defuzzification Module

LA's Actions Selector Module: For each membership function a S-L_{REP} learning automata is used with parameters $a=0.1$ and $b=0.05$. Each learning automata consists of 10 operations with initial probability, 0.1. In this module accordingly to the probability vector, one operation will be selected for each learning automata. If (a_i) operation is selected after (n) repetitions, and environment responses $b_i(n)$ to it, then the automata probability vectors are presented according to the following formula: If the queue gets worse, a penalty is allotted, $b_i(n) = 1$, otherwise a bonus is given.

Fuzzifier Module: Here, according to the selected operations, the amounts of input parameters are converted to a fuzzy module. Note that, for producing input parameters Q_c and Q_N poisson distribution and for remaining inputs uniform distribution has been used. Fig. 8 represents a sample coloured Petri Net fuzzifier module, of parameter Q_N .

Rules and Defuzzification Module: Rules of operations, and achieved results, are both converted to numerical values. And this shows the amount of increased time applied to the green signal (see Fig. 8) [20]-[22]).

V. SIMULATION RESULTS

Uniform delay formula is used to calculate the average delay of each vehicle [23]. Simulated model was performed 600 times by the colored Petri Net tool to achieve the required results. The average delay of the 8 transitional phases are calculated and shown in TABLE I:

TABLE I. AVERAGE DELAY PER VEHICLE DUE TO UNIFORM ARRIVALS

Phases	1	2	3	4	5	6	7	8
AV (Sec)	46	48	51	48	51	51	49	36

In [4] when the Traffic Volume in all directions reaches 1600, the mean delay in 2 transitional phases is approximately 27 (sec). The calculated mean delay of vehicles in 4 transitional phases is reported in [17] (TABLE II)

TABLE II. AVERAGE DELAY PER VEHICLE WITH ABNORMALITIES

Phases	1	2	3	4
AV (Sec)	32	19	20	20

VI. CONCLUSIONS

In this paper an adaptive fuzzy coloured Petri Net has been presented based on learning automata to efficiently control the traffic signals across intersections. The basis of the recommended algorithm was to combine the fuzzy logic and learning automata. Learning automata was used to regulate and adjust the membership functions in fuzzy system. The comparative model tries to predict the perfect status and represents the current status of the system, according to the information achieved from the prior states, combined with the reactions of the dynamic environment. The achieved delay average of (r) vehicles in our algorithms was compared to the other known algorithms for evaluation. The results achieved showed that the proposed algorithm has significantly a better performance in achieving the specified goal.

REFERENCES

- [1] K. Jensen , L. Michael Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," International Journal of Software Tools Technology Transfer, pp. 213-254, 2007.
- [2] A. Asar, M. Zhou, and R.J. Caudill, "Making Petri Nets Adaptive : A Critical Review," IEEE Networking, Sensing and Control Proceedings, pp. 644-649, 2005.
- [3] C.P. Pappis, and E.H. Mamdani, "A fuzzy logic controller for a traffic junction," IEEE Trans. Systems, Man, and CyberNetics, vol. 7, no.10, pp. 707-717, 1977.
- [4] J. Niittymaki And M. Pursula, "Signal control using fuzzy logic," Fuzzy Sets and Systems, vol. 116, no. 1, pp. 11-22, 2000.
- [5] J. Niittymaki And S. Kikuchi, "Application of fuzzy logic to the control of a pedestrian crossing signal," Transportation Research Record: Journal of the Transportation Research Board, vol. 1651, pp. 30-38, 1998.
- [6] S. Chiu, "Adaptive traffic signal control using fuzzy logic," in Proc. of the IEEE Intelligent Vehicles Symposium, pp. 98-107, 1992.
- [7] J. Niittymaki, "Installation and experiences of field testing a fuzzy signal controller," European Journal of Operational Research, pp. 273-281, 2001.
- [8] Y.S. Huang, T.H. Chung, and T. Lin, "Design and Analysis Urban Traffic Lights Using Timed Colour Petri Nets," in Proc. of IEEE International Conf. on Networking, Sensing and Control, pp. 248-253, 2006.

- [9] M. Davoudpour, and E.V. Rudakov, "Formal model of decomposition algorithm for functional block in discrete devices," *Vestnik Instruments Journal*, Bauman Moscow State Technical University, Moscow, Russia. vol. 1, pp. 90-98 2006.
- [10] K. S. Narendra, and M. A. L. Thathachar. "Learning Automata: An Introduction," Prentice Hall, 1989.
- [11] D. S. Yeung, J. N.K. Liu, S. C.K. Shiu, and G. S.K. Fung, "Fuzzy Coloured Petri Nets in Modelling Flexible Manufacturing Systems," ITESM, pp. 100-107, 1996.
- [12] Y. Ouchi and E. Tazaki, "Learning and Reasoning Method Using Fuzzy Coloured Petri Nets under Uncertainty," in Proc. of the IEEE International Conf. on Computational Cybernetics and Simulation, vol. 4, pp. 3867-3871, 1997.
- [13] C. Zhou and Z. Jiang, "Fault Diagnosis of TV Transmitters Based on Fuzzy Petri Nets," in Proc. of the IMACS Multiconference on Computational Engineering in Systems Applications (CESA), pp. 2003-2009, October 4-6, 2006.
- [14] A. Sadeghian, "Nonlinear Neuro-Fuzzy Prediction: Methodology, Design & Application," in Proc. of the 10th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'2001), vol. 2, pp. 1022-1026, Dec. 2001.
- [15] G. F. List, and M. Cetin, "Modeling Traffic Signal Control Using Petri Nets," *IEEE Trans. Intelligent Transportation Systems*, vol. 5, no. 3, pp. 177-187, 2004.
- [16] M. H. F. Zarandi, and S. Rezapour, "A Fuzzy Signal Controller for Isolated Intersections," *Journal of Uncertain Systems*, vol.3, no.3, pp. 174-182, 2009.
- [17] B. M. Nair, and J. Cai, "A fuzzy Logic Controller for Isolated Signalized Intersection with Traffic Abnormality Considered," in Proc. of IEEE Intelligent Vehicles Symposium, pp. 1229-1233, June 13-15, 2007.
- [18] L. Li, W.-H. Lin, and H. Liu, "Traffic Signal Priority/Preemption Control with Colored Petri Nets," in Proc. of the 8th International IEEE Conference on Intelligent Transportation Systems, pp. 694-699, 2005.
- [19] L.G. Zhang, Z.L. Li, and Y.Z. Chen, "Hybrid Petri Net Modeling of Traffic Flow and Signal Control," in Proc. of the Seventh International Conference on Machine Learning and CyberNetics, pp. 2304-2308, July 2008.
- [20] J. Lee, K. F. R. Liu, and W. Chiang, "A Fuzzy Petri Net-Based Expert System and Its Application to Damage Assessment of Bridges," *IEEE Trans. Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 29, no. 3, June 1999.
- [21] W. Pedrycza, H. Camargo, "Fuzzy timed Petri Nets," *Fuzzy Sets and Systems*, vol. 140, no. 2, pp. 301-330, 2003.
- [22] Y. Ting, W.B. Lu, C.H. Chen, and G.K. Wang, "A fuzzy reasoning design for fault detection and diagnosis of a computer-controlled system," *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 157-170, 2007.
- [23] D. B. Fambro and N. M. Rouphail, "Generalized Delay Model for Signalized Intersections and Arterial Streets," *Transportation Research Record*, pp. 112-121, 1997.