Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

An optimal general type-2 fuzzy controller for Urban Traffic Network

Mohammad Hassan Khooban [a,*], Navid Vafamand [b], Alireza Liaghat [b], Tomislav Dragicevic [a]

[a] Department of Energy Technology, Aalborg University, Aalborg DK-9220, Denmark
[b] School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran

ARTICLE INFO

Article history:
Received 12 January 2016
Received in revised form
8 September 2016
Accepted 10 October 2016
This paper was recommended for publication by Oscar Camacho

Keywords:
Traffic Signal Control
Optimal General Type-2 Fuzzy Controller (OGT2FC)
Modified Backtracking Search Algorithm (MBSA)

ABSTRACT

Urban traffic network model is illustrated by state-charts and object-diagram. However, they have limitations to show the behavioral perspective of the Traffic Information flow. Consequently, a state space model is used to calculate the half-value waiting time of vehicles. In this study, a combination of the general type-2 fuzzy logic sets and the Modified Backtracking Search Algorithm (MBSA) techniques are used in order to control the traffic signal scheduling and phase succession so as to guarantee a smooth flow of traffic with the least wait times and average queue length. The parameters of input and output membership functions are optimized simultaneously by the novel heuristic algorithm MBSA. A comparison is made between the achieved results with those of optimal and conventional type-1 fuzzy logic controllers.

1. Introduction

One of the main difficulties in big modern cities is the overpopulations of automobiles. Traffic signal control has a crucial role in the transport safety and the smoothness of traffic flow. Particularly, in order to prevent automobiles overcrowding in urban streets, not only off-line timing, but also real-time control of traffic signals has been proposed recently [1–4].

A crossroad is the main node of the municipal transport network. Collection of the traffic data and control of the traffic flow surrounding it is a challenging research subject. There are two major techniques in signal control: off-line signal control and real-time signal control. Because of the stochastic nature of traffic flow, the off-line traffic control technique can be only used for less crowded crossroads. The real-time technique optimizes the signal control based on the information collected by sensors. There are several control techniques in literatures. For example in [3,4], a novel technique based on video reorganization is presented.

In some research activities, optimization algorithms are used. The major optimization algorithms include fuzzy logic system (FLS), neural network-fuzzy (NNF), multi-objective genetic algorithms (MOGA), and Markov Process [5–9].

Lately, Diakaki et al. [2] have suggested a model for traffic monitoring, and have used optimal linear quadratic regulator for controlling the model. While this study looks encouraging for smart control of traffic, the model and the control procedure have several limitations. In order to overcome these defects, a more comprehensive model and a robust control method have been suggested in [10,11]. The modeling procedures in [10,11] and also in [2] are based on the so-called store-and-forward modeling method which requires some statistical data related to traffic, for instance saturation flows and turning motion rates to create the model. So, this modeling procedure has flaws that it is fairly sophisticated to create a model from statistical data and that it might be too perfect to consider for an actual traffic problem.

The fuzzy logic controller (FLC) is credited with being an appropriate method for designing robust controllers that are capable of delivering a satisfactory efficiency against uncertainty; therefore the FLC has become a common solution to reactive traffic signal control in recent years [12]. The type-1 FLCs have the popular problem that they cannot be completely used for the linguistic and numerical uncertainties related to variable environmental conditions as they apply accurate type-1 fuzzy sets. Type-1 fuzzy sets employ the uncertainties related to the FLC inputs and outputs by applying accurate and crisp membership functions that the gainer believes that uncertainty is inhibited [13]. When the type-1 membership functions have been selected, all the uncertainty is eliminated, since type-1 membership functions are completely accurate [13,14]. The linguistic and numerical uncertainties related to variable environmental conditions create problems in specifying the accurate consequents membership functions over the design process.

* Corresponding author.
E-mail addresses: mhkhoban@gmail.com, khooban@sutech.ac.ir, mhk@et.aau.dk (M.H. Khooban).

http://dx.doi.org/10.1016/j.isatra.2016.10.011
0019-0578/

Please cite this article as: Khooban MH, et al. An optimal general type-2 fuzzy controller for Urban Traffic Network. ISA Transactions (2016), http://dx.doi.org/10.1016/j.isatra.2016.10.011

Lately, many researchers [15–18] pay attention to general type-2 fuzzy sets and systems because of their ability to deal with uncertainties and disturbances. Zadeh [17] in 1975 presented Type-2 fuzzy sets as an extension of type-1 fuzzy sets and have been used in engineering areas successfully. For instance, [19] demonstrates the efficient performance of IT2FLSs in comparison to type-1 fuzzy logic systems (T1FLS) when faced with various uncertainties such as dynamic uncertainties, rule uncertainties, external disturbances and noises. Available information for making the rules in a fuzzy logic system can be uncertain. Unlike interval type-2 fuzzy sets (IT2FS) and type-1 fuzzy sets (T1FS), general type-2 fuzzy sets can deal with the rule uncertainties. In literatures, only IT2FLSs have been mainly applied until now because general type-2 fuzzy sets and systems are computationally complex. Liu [19] proposed a useful fast process for computing centroid and type reduction of GT2FLS by using a recent plane representation theorem. In [20–23], an in-depth description of the zSlices-based representation, which enables the representation of and computation with general type-2 fuzzy sets and their associated third dimensions at a level of precision and associated computational overhead, which can be chosen as required by the respective application has given. Bilgin et al. [24] addressed the need to enhance transparency in Ambient Intelligent Environments by developing more natural ways of interaction, which allow the users to communicate easily with the hidden networked devices rather than embedding obtrusive tablets and computing equipment throughout their surroundings. A novel zSlices based general Type-2 Fuzzy PI (zT2-FPI) controller where the SMFs are adjusted in an on-line manner through a single tuning parameter is presented in [25].

Motivated by the aforementioned researches, the purpose of this paper is to present an Optimal General Type-2 Fuzzy Controller (OGT2FC) for controlling the traffic signal scheduling and phase succession to guarantee a smooth flow of traffic with the least wait times and average queue length. The parameters of input and output membership functions are optimized simultaneously by a novel heuristic algorithm called Modified Backtracking Search Algorithm (MBSA). Simulation results indicate the superiority of the proposed controller over the non-optimal type-1 fuzzy controller and optimal type-1 fuzzy controller.

2. General type-2 fuzzy sets and systems

A GT2FS in a universal set X can be defined as

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x \tag{1}$$

$$\mu_{\tilde{A}}(x) = \int_{u \in J_x} (z_x(u))/u, \ J_x \in [0, 1] \tag{2}$$

where in this formula $\mu_{\tilde{A}}(x)$ is called a secondary membership function (MF) and $z_x(u)$ is called secondary grade; J_x is the domain of the secondary MF which is called primary membership and u is a fuzzy set in [0, 1]. Fig. 1 illustrates a GT2FS where the upper and lower MFs are triangular and its secondary MF is also triangular. When $z_x(u)$=1 IT2FS is obtained that demonstrate a uniform uncertainty in the primary membership function and is simply described by its lower $\underline{\mu}_{\tilde{A}}(x)$ and upper $\bar{\mu}_{\tilde{A}}(x)$ MFs. Because of calculation simplicity, especially in the type reduction, many researchers use interval type-2 fuzzy sets instead of general type-2 fuzzy sets [16,18,20].

Lately, Liu [19] presented a new method for GT2FSs which is theoretically and computationally effective. Because this method resembles the α-cut for type-1 fuzzy sets, it is named a α-plane for type-2 fuzzy sets. $\tilde{A}_\alpha$ is the denotation of An α-plane representation for a GT2FS $\tilde{A}$. It is the union of all primary MFs whose secondary grades are greater than or equal to the special value α:

$$\tilde{A}_\alpha = \int_{x \in X} \mu_{\tilde{A}_\alpha}(x)/x \tag{3}$$

$$\mu_{\tilde{A}_\alpha}(x) = \int_{u \in J_x} (z_x(u) \geq \alpha)/u, \ J_x \in [0, 1] \tag{4}$$

Then a GT2FS $\tilde{A}$ based on α-plane representation theorem can be demonstrated in the following form:

$$\tilde{A} = \bigcup_{\alpha \in [1,0]} \alpha/\tilde{A}_\alpha \tag{5}$$

It is a beneficial representation because $\alpha/\tilde{A}_\alpha$ can be seen as an IT2FS with the secondary grade of level α. As a result, several IT2FSs may be made from the decomposition of a general type-2 fuzzy set with a corresponding level of α for each, where $\alpha = \{0,1/K,\ldots,(K-1)/K,1\}$. In simpler terms, a general type-2 fuzzy logic system can be seen as a huge collection of IT2FLSs with one IT2FLS for each value of α. However, Liu [24] showed that using only 5 to 10 α-plane can get the required accuracy for centroid calculation. Fig. 2 illustrates the new design for a general type-2 fuzzy system based on α-plane representation.

In general, a GT2FLS is made of a fuzzifier; fuzzy rule-based; fuzzy inference engine; type reducer and defuzzifier. Fuzzifier maps real values into fuzzy sets. Singleton fuzzifier whose output is a single point of a unity membership grade is used in this paper

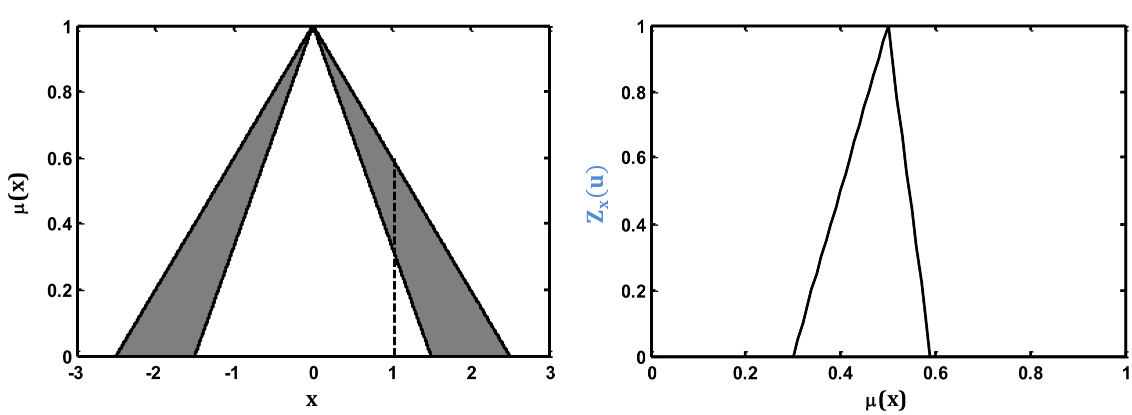

Fig. 1. A general type-2 fuzzy set with triangular upper and lower MFs where the secondary MF is triangular.

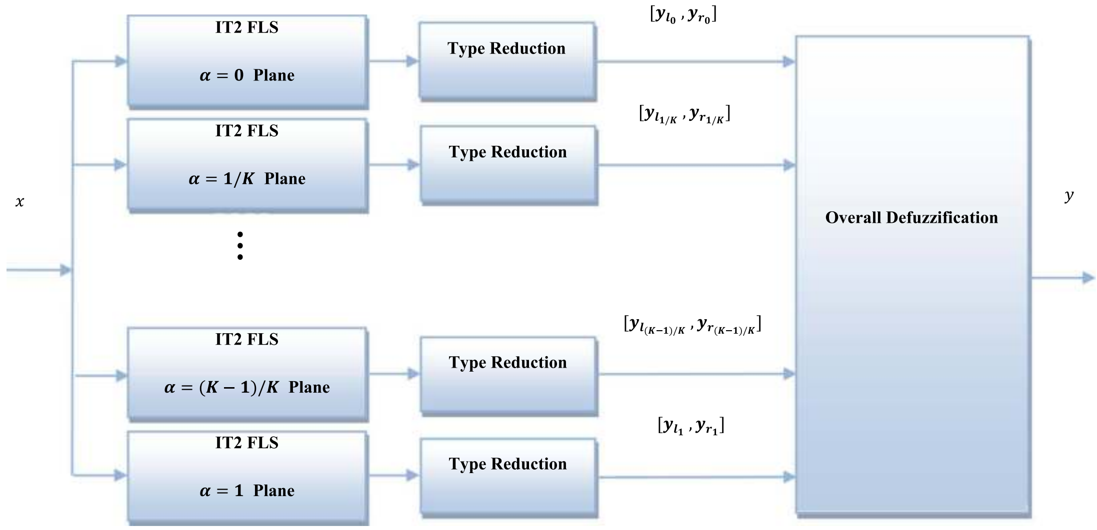

Fig. 2. Architecture for a general type-2 fuzzy Logic system.

because it is simple. Fuzzy rule base includes fuzzy IF-THEN rules. In the following The j^{th} rule in the GT2FLS is shown:

$$R^j\text{: If } x_1 \text{ is } \tilde{F}_1^j \text{ and } \dots\ x_n \text{ is } \tilde{F}_n^j \text{ then } y \text{ is } \tilde{G}^j \quad j = 1, 2, \dots, M \tag{6}$$

where x_i for $i = 1, \dots, n$ and y are the input and output of the GT2FLS, $\tilde{F}_i^j$ and $\tilde{G}^j$ are general type-2 antecedent and the consequent sets. A mapping from input GT2FSs to output GT2FSs is given by the inference engine that merges rules. Because α-plane representation for fuzzy set is used, the firing set for each related IT2FS is shown as following:

$$F_\alpha^j(X) = \left[\underline{f}_\alpha^j(X),\ \bar{f}_\alpha^j(X)\right] \tag{7}$$

$$\underline{f}_\alpha^j(X) = \underline{\mu}_{\tilde{F}^j_{1_\alpha}} * \underline{\mu}_{\tilde{F}^j_{2_\alpha}} * \dots * \underline{\mu}_{\tilde{F}^j_{n_\alpha}} \tag{8}$$

$$\bar{f}_\alpha^j(X) = \bar{\mu}_{\tilde{F}^j_{1_\alpha}} * \bar{\mu}_{\tilde{F}^j_{2_\alpha}} * \dots * \bar{\mu}_{\tilde{F}^j_{n_\alpha}} \tag{9}$$

Here $\underline{f}_\alpha^j(X)$ and $\bar{f}_\alpha^j(X)$ are the lower and upper MFs of the j^{th} rule with level of α, and * indicates product t-norm. a type reducer changes the output of the inference engine which is a type-2 fuzzy set into a type-1 fuzzy set before defuzzification. Five kinds of reducers which are based on calculating the centroid of an IT2FS are demonstrated in [25]. The output of the type reduction in IT2FLS is defined with its left-end point y_l and right-end point y_r due to uniformly secondary grade of IT2FLS.

KM iterative algorithms, introduced two algorithms for calculating these two end points in [25–27], as presented by Mendel and Karnik. In comparison to the other type reduction methods, center of sets (COS) is used a lot because of its computation simplicity [15]. If a singleton fuzzifier is used, product inference engine and COS type reducer, left and right end points for each part of GT2FLS based on α- representation theorem can be shown as follows:

$$y_{l_\alpha} = \frac{\sum_{j=1}^{L} \bar{f}_\alpha^j \theta_{l_\alpha}^j + \sum_{j=L+1}^{M} \underline{f}_\alpha^j \theta_{l_\alpha}^j}{\sum_{j=1}^{L} \bar{f}_\alpha^j + \sum_{j=L+1}^{M} \underline{f}_\alpha^j} = \theta_{l_\alpha}^T \xi_{l_\alpha} \tag{10}$$

where in this formula $\theta_{l_\alpha}^j$ is the left-end point of j^{th} consequent set with level of α, $\theta_{l_\alpha} = \left[\theta_{l_\alpha}^1, \dots, \theta_{l_\alpha}^M\right]^T$, $\xi_{l_\alpha}^j = \left[\frac{\underline{f}_\alpha^j}{D_{l_\alpha}}, \frac{\bar{f}_\alpha^j}{D_{l_\alpha}}\right]$, $D_{l_\alpha} = \sum_{j=1}^{L} \bar{f}_\alpha^j + \sum_{j=L+1}^{M} \underline{f}_\alpha^j$ and $\xi_{l_\alpha} = \left[\xi_{l_\alpha}^1, \dots, \xi_{l_\alpha}^M\right]^T$. In addition,

$$y_{r_\alpha} = \frac{\sum_{j=1}^{R} \bar{f}_\alpha^j \theta_{r_\alpha}^j + \sum_{j=R+1}^{M} \underline{f}_\alpha^j \theta_{r_\alpha}^j}{\sum_{j=1}^{R} \bar{f}_\alpha^j + \sum_{j=R+1}^{M} \underline{f}_\alpha^j} = \theta_{r_\alpha}^T \xi_{r_\alpha} \tag{11}$$

where $\theta_{r_\alpha}^j$ is the right end point of j^{th} consequent set with level of α, $\theta_{r_\alpha} = \left[\theta_{r_\alpha}^1, \dots, \theta_{r_\alpha}^M\right]^T$, $\xi_{r_\alpha}^j = \left[\frac{\underline{f}_\alpha^j}{D_{r_\alpha}}, \frac{\bar{f}_\alpha^j}{D_{r_\alpha}}\right]$, $D_{r_\alpha} = \sum_{j=1}^{R} \bar{f}_\alpha^j + \sum_{j=R+1}^{M} \underline{f}_\alpha^j$ and ξ_{r_α} $=\left[\xi_{r_\alpha}^1, \dots, \xi_{r_\alpha}^M\right]^T$ Meanwhile, performing KM iterative algorithm can specify R and L for each individual IT2FLS of level α. From the combination of all of these obtained intervals into a type-1 fuzzy set like Fig. 3, a crisp output can be obtained using centroid defuzzification as:

$$y = \frac{\sum_\alpha \alpha\left(y_{l_\alpha} + y_{r_\alpha}\right)}{2\sum_\alpha \alpha} = \frac{\sum_\alpha \alpha\left(y_{l_\alpha} + y_{r_\alpha}\right)}{K+1}, \ \alpha = \{0, 1/K, \dots, (K-1)/K, 1\} \tag{12}$$

where $K + 1$ shows the number of the α-planes or in other words it determines the number of individual IT2FLSs.

3. Optimization technique

Based on the updating process of genetic algorithm (GA) and differential evolution (DE), a new backtracking search algorithm (BSA) is developed as an evolutionary optimization (EO) technique in [28]. Similar to other EOs, the application of basic principles of

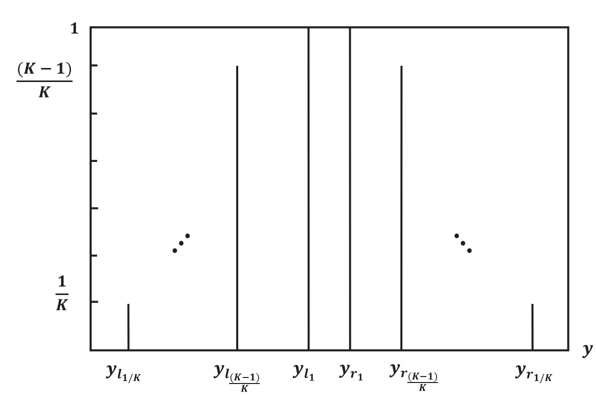

Fig. 3. Output of each individual IT2FLSs.

BSA to mathematical optimization begins with the random generation of an initial population [34]. An objective function is used to evaluate each individual of the initial set. It profits from a random mutation techniques which employs only the one direction individual instead of multiple ones comparing to the DE algorithm. Moreover, a non-uniform crossover procedure is implemented along with the BSA which is more complicated than the GA's crossover. In the crossover step, the variables of two selected individuals are combined to each other and in the mutation step the new solutions are randomly generated in order to diversify the solution search space and allow the BSA to eventually escape from locally optimal solutions. Furthermore, two new selection strategies are defined and used by the BSA. According to this information, the BSA can be divided into the following three steps. It should be noted that the proposed modification strategies are explained during the process of the BSA for the sake of explicitness.

3.1. Initialization

Initially the N number of individual is randomly generated and positioned at random locations on the search space. Here, the N represents the D parameters involved in the fuzzy controller. The individual position is characterized as an initial vector of length N and D as the problem dimension. The initial position vector can be represented as:

$$X=\begin{bmatrix} X_{1,1} & \dots & X_{1,j} & \dots & X_{1,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{i,1} & \dots & X_{i,j} & \dots & X_{i,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{N,1} & \dots & X_{N,j} & \dots & X_{N,D} \end{bmatrix} \tag{13}$$

Each array $X_{i,j}$ is generated randomly using the uniform distribution U as it can be expressed:

$$X_{i,j}=X_{min,j}+U_{i,j}(\bullet)(X_{max,j}-X_{min,j}) \tag{14}$$

Where, $X_{min,j}$ and $X_{max,j}$ are the lower and upper bounds of the jth member of the problem dimension. The $U_{i,j}(\bullet)$ is the random function generator in the range of $[0,1]$ from the uniform probability distribution function for the jth member of the ith agent.

- **First modification**

Chaotic opposition-based learning is suggested in this study to enhance the quality of the final solution and speed up the convergence rate of the algorithm. If the random initialization is not far away from the optimal solution, then the convergence is expected to be faster. In this learning scheme, the chaotic systems and the opposition-based learning technique are combined to generate the initial population. The population is divided into two groups. The first group is initialized based on the chaotic theory and the second one is generated according to the opposition-based learning. Here, a sinusoidal based chaotic operator is used to generate a chaotic number as follows:

$$C_{i,j,k+1}=\sin\left(\pi C_{i,j,k}\right),\ k=1,\ \dots,k_{max}$$
$$C_{i,j,1}=U_{i,j}(\bullet) \tag{15}$$

where k is the iteration counter of sinusoidal iterator and k_{max} is the maximum iteration number for chaotic mechanism. Accordingly, the first group (N_1=$N/2$) is generated as follows:

$$X_{i,j}^{C}=X_{min,j}+C_{i,j,k_{max}}(\bullet)\left(X_{max,j}-X_{min,j}\right),\ i=1,\ \dots,N_1 \tag{16}$$

Where $X_{i,j}^{C}$ is the jth variable of the ith individual which is initially generated by the chaotic system. The rest of the population is initialized as follows [29]:

$$X_{i,j}^{o}=X_{min,j}+X_{max,j}-X_{i,j} \tag{17}$$

Where $X_{i,j}^{o}$ is the jth variable of the ith individual which is initially generated by the opposition-based learning scheme.

3.2. Selection-I

The BSA uses the $\boldsymbol{X}^{old}$ vector to update the historical population at the beginning of each iteration. This vector is initialized like the $\boldsymbol{X}$ and updated through the following 'If-Then' rule [28]:

$$\text{If } U_1(\bullet)<U_2(\bullet) \text{ Then } \boldsymbol{X}^{old}\bowtie{:}X \tag{18}$$

where $\bowtie$: is an operation which is used for updating process of $\boldsymbol{X}^{old}$. It means that the BSA has a memory and remembers the unchanging population from the previous generation until it is changed. After determining the current $\boldsymbol{X}^{old}$, the following process is implemented to modify the order of the individuals in the $\boldsymbol{X}^{old}$ in the current iteration as follows:

$$\boldsymbol{X}^{old}=\boldsymbol{X}^{old}\left(randperm\left(N\right)\right) \tag{19}$$

where $randperm\left(N\right)$ is a random shuffling function and randomly selects the index from N.

3.3. Mutation, crossover and selection II

Following the embodiment of the selection-I process, a mutation is a next implementation step in the BSA. Based on this, the trial solutions are generated as follows [28]:

$$\boldsymbol{X}^{trial}=\boldsymbol{X}+\left(F.\ \boldsymbol{M}\right).(\boldsymbol{X}^{old}-\boldsymbol{X}) \tag{20}$$

where F is a parameter that controls the amplitude of the search direction vector $\boldsymbol{X}^{old}-\boldsymbol{X}$ and considered to be $3\times rndn$ in which $rndn\sim N(0,1)$ (N is the standard normal probability distribution function). Moreover, $\boldsymbol{M}$ is the binary integer-valued vector of size $N\times D$ which is used for the crossover purpose. It is generated as follows:

If $U_1(\bullet)<U_2(\bullet)$

For $i=1{:}N$	For $i=1{:}N$
For $j=1{:}D$	For $j=1{:}D$
If $j\in Q_1$	If $j\in Q_2$
$M_{i,j}$=1	$M_{i,j}$=1
Else	Else
$M_{i,j}$=0	$M_{i,j}$=0
End	End
End	End
End	End
End	

(21)

where

$$Q_1=u\left(1:ceil\left(U(\bullet)\right).\ D\right)$$
$$u=randperm\left(D\right) \tag{22}$$

$$Q_2=randi\left(D\right) \tag{23}$$

with $ceil(S)$ as a function which rounds S toward the upper integer. Also, the $randi$ produces random integers from a uniform discrete distribution.

It is notable that the trial solutions with better fitness values are replaced with the same position of population $\boldsymbol{X}$ for the next iteration of the BSA.

- **Second modification**

Following the embodiment of chaotic opposition-based learning process, a new self-adaptive learning strategy is presented and applied to the original BSA in order to diversify the problem search space. Based on the learning experience, a self-adaptive learning guides the decision making for the modification rules. Three modification rules are adopted to extend the exploration and exploitation area in the method, which are expressed as follows:

– Modification rule 1:

$$\boldsymbol{X}_{i,mod1} = \begin{cases} \boldsymbol{X}_{i,new} + \upsilon(\boldsymbol{Best} - \boldsymbol{Worst}) \text{if } \upsilon>0 \\ \boldsymbol{X}_{i,new} + \upsilon(\boldsymbol{Worst} - \boldsymbol{X}_{i,new}) \text{otherwise} \end{cases} \tag{24}$$

where $\boldsymbol{X}_{i,mod1}$ is the generated trial solution for the ith individual through the modification rule 1. The $\boldsymbol{Best}$ and $\boldsymbol{Worst}$ are the current best and worst solutions, respectively. The υ is formulated as follows:

$$\upsilon = \frac{1}{\sqrt{2}} e^{\left(\frac{-\phi^2}{2d^2}\right)} \cos\left(\frac{5\phi}{d}\right) \tag{25}$$

where ϕ is a random number in the range of $[-2.5d, 2.5d]$. The amount of d is set to change with the iteration number and expressed as follows:

$$d = e^{\ln(iter_{max})\left(1-\left(1-\left(\frac{iter}{iter_{max}}\right)\right)^{-0.5}\right)} \tag{26}$$

where $iter_{max}$ is the maximum number of iterations for the MBSA.

– Modification rule 2:

$$\boldsymbol{X}_{i,mod2} = \boldsymbol{X}_{i,new} + U_i(\bullet)(\boldsymbol{X}_{m,new} - \boldsymbol{X}_{i,new}) \tag{27}$$

where $\boldsymbol{X}_{i,mod2}$ is the generated trial solution for the ith individual through the modification rule 2. The m is randomly chosen individual index from the population.

– Modification rule 3:

$$\boldsymbol{X}_{i,mod3} = \boldsymbol{X}_{i,new} + U_i(\bullet)(\boldsymbol{Best} - \boldsymbol{Mean}) \tag{28}$$

where $\boldsymbol{X}_{i,mod3}$ is the generated trial solution for the ith individual through the modification rule 3. The $\boldsymbol{Mean}$ is the mean value of the current population which is provided as follows:

$$\boldsymbol{Mean} = [me_1 \ldots me_j \ldots me_D] \tag{29}$$

$$me_j = \sum_{i=1}^{N} \frac{\boldsymbol{X}_{i,j,new}}{N} \tag{30}$$

While modification rule 1 improves both the global and local search capacity, modification rules 2 and 3 enhance the local and global search capacities, respectively. The selection of modification rules depends on the following normalized probability model:

$$Prob_r = \frac{Prob_r}{Prob_1 + Prob_2 + Prob_3} \tag{31}$$

$$Prob_r = 0.85 Prob_r + 0.15 \frac{ac_r}{10} \tag{32}$$

$$ac_r = ac_r + \frac{w_{ii}}{N_r} \quad ii = 1, \ldots, N_r \tag{33}$$

$$w_i = \frac{\log(N - i + 1)}{\log(1) + \ldots + \log(N)} \quad i = 1, \ldots, N \tag{34}$$

where N_r is the number of population which selects the modification rule r.

Finally, the roulette wheel mechanism on the basis of normalized probability (31) is employed in order to choose the rth modification rule for each individual.

4. An intelligent approach for the single crossroad

4.1. State-space equations

The two-phased signalized crossroad is shown in Fig. 4 [30]. In this figure, the Leg 1 and Leg 3 are phase 1, and the Leg 2 and Leg 4 are phase 2.

The average queue length is a significant parameter that characterizes the traffic state of a crossroad. The queue is defined as follows

$$Q_i(n+1) = Q_i(n) + q_i(n) - d_i(n)S_i(n) \tag{35}$$

where $i = 1, 2, \ldots, M$ is the index of the traffic flows; $n = 0, 1, \ldots, N-1$ is the index of the discretized time intervals; $Q_i(n)$, based on number of automobiles, is the queue length of the ith flow at the beginning of the nth time interval; $q_i(n)$ is the number of automobiles that join the ith queue in the nth time interval; $d_i(n)$ is the number automobiles that leave the ith queue in the nth time interval; and $S_i(n)$, which is equal to 0 (for stopping) or 1 (for going), is the signal state of the ith flow in the nth time interval. q_i and d_i are typically considered random signals.

In fixed time monitoring and fuzzy smart control, the control variables are studied as follows [30]. For phase 1 crossroad, $(S_1, S_2, S_3, S_4) = (0,1,0,1)$, which means traffic signal is green in lanes 2 and 4 and red in lanes 1 and 3. So, the automobiles can move in

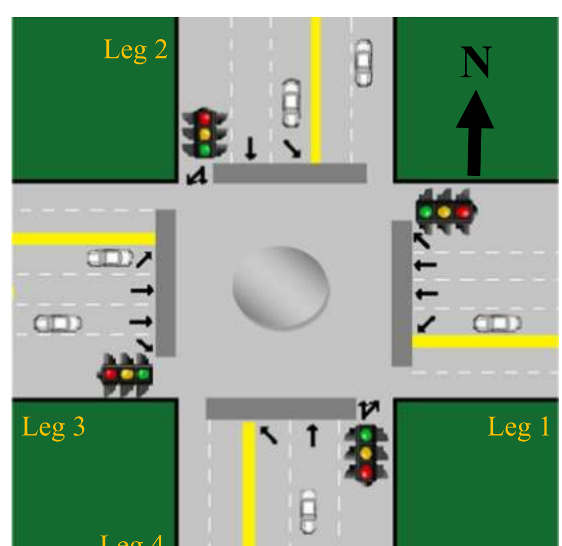

Fig. 4. Two phase signalized crossroad. (For interpretation of the references to color in this figure,the reader is referred to the web version of this article.)

Please cite this article as: Khooban MH, et al. An optimal general type-2 fuzzy controller for Urban Traffic Network. ISA Transactions (2016), http://dx.doi.org/10.1016/j.isatra.2016.10.011

lanes 2 and 4 and they should stay in lanes 1 and 3. On the other side, for phase 2 crossroad, $(S_1, S_2, S_3, S_4) = (1,0,1,0)$, which means traffic signal is green in lanes 1 and 3 and red in lanes 2 and 4. So, the automobiles can move in lanes 2 and 4 and they should stay in lanes 2 and 4.

Integrating the average queue length due to the time, results the average waiting time of automobiles in the queue. Let T is short sufficiently, the automobiles arrivals can be assumed as being identical in each time interval. Therefore, integrating (24) yields

$$W_i(n+1) = W_i(n)+TQ_i(n)+\frac{1}{2}Tq_i(n)-\frac{1}{2}Td_i(n)S_i(n) \tag{36}$$

where $W_i(n)$ is the average waiting time of the ith queue since the start of the time interval to the beginning of the nth time interval.

Eqs. (35) and (36) are the state-space equations characterizing the dynamic development of the traffic state at a separate crossroad. The length of queue and the average waiting time are two common efficiency indices for signal controls. The average waiting time is utilized here as the efficiency index. So, the cost function is [31–33]:

$$min\left\{ W(N) = \sum_{i=1}^{M} W_i(N) \right\} \tag{37}$$

To expedite the formulation, the state-space equations and the cost function can be overridden in matrix form as:

$$X(n + 1) = AX(n)+B(n)S(n)+C(n) \tag{38}$$

$$y(n)=CX(n) \tag{39}$$

where

$X(n)=\left[Q_1(n)Q_2(n)..\,.Q_M(n)W_1(n)W_2(n)..\,.W_M(n)\right]^T$ are the state variables and $S(n)=\left[S_1(n)S_2(n)..\,.S_M(n)\right]^T$ are the control variables. The different coefficient matrices and vectors are [32]

$$A = \begin{bmatrix} I_M & 0 \\ TI_M & I_M \end{bmatrix}$$

$$B(n) = -\begin{pmatrix} d_1(n) & 0 & \dots & 0 \\ 0 & d_2(n) & \dots & 0 \\ \vdots & \vdots & \ddots & \dots \\ 0 & \vdots & \dots & 0 \\ \frac{1}{2}Td_1(n) & 0 & \dots & d_m(n) \\ 0 & \frac{1}{2}Td_2(n) & \dots & 0 \\ \vdots & \dots & \ddots & 0 \\ 0 & 0 & \dots & \frac{1}{2}Td_M(n) \end{pmatrix}$$

$$C = \begin{bmatrix} I_M & 0 \\ 0 & I_M \end{bmatrix}$$

$$C(n) = \left[q_1(n)\cdots q_M(n)\frac{1}{2}Tq_1(n)\cdots\frac{1}{2}Tq_M(n)\right] \tag{40}$$

4.2. Optimal Type-2 fuzzy controller for a single crossroad

Designing an optimal Type-2 fuzzy controller for a single signalized crossroad comprises three steps. In the following, we will present these steps in detail.

The first step in designing of a fuzzy controller is determination of input and output variables. Since the goal is to minimize the cost function (37), $W_i(n)$ are chosen as the input variables. Furthermore, the states $Q_i(n)$ are directly related to the number joining and leaving automobiles (i.e. $q_i(n)$ and $d_i(n)$). Therefore, we select $Q_i(n)$ as the other input variables. Consequently, the input variables of the fuzzy controller are state variables of the state-space system given in (38) and (39). Also, the control variables $S_i(n)$ are the output of the fuzzy controller.

The second step in designing of a fuzzy controller is fuzzification. The fuzzification is the process of making a crisp quantity fuzzy. The input variables $W_i(n)$ and $Q_i(n)$ for $i = 1, \dots, 4$ are divided into three fuzzy membership functions: "Low(L)," "Medium(M)," and "High(H)" and the output variables $S_i(n)$ are divided into two set: "Going(G)" and "Stopping(S)". Going is the continuation of green phase and Stopping is the red phase. Best locations of the left and right "feet" or base points of the triangle and also best location of the triangle peak are set by MBSA. In other words, the MBSA optimization is performed to compute the lower and upper MFs of the type-2 MF. To do this, the MBSA is carried out based on the fitness function (37), the initial population given in section III, and random values for $q_i(n)$ [35].

The next step in designing of a fuzzy controller is determination of fuzzy rules. A more precise analyze of the single intersection traffic problem in Fig. 4, reveals that the traffic lights of the legs 1 and 3 (traffic light pair S_1-S_3) and the legs 2 and 4 (traffic light pair S_2-S_4) should change simultaneously and conversely. In other word, when the pair (S_1-S_3) changes to green, the other pair (S_2-S_4) turns to red. Furthermore, the two pairs (S_1-S_3) and (S_2-S_4) are dependent to each straight road overall queue length and waiting time (i.e. $Q_1(n)+Q_3(n)$, $Q_2(n)+Q_4(n)$, $W_1(n)+W_3(n)$, and $W_2(n)+W_4(n)$). Consequently, one fuzzy controller is needed for these two pairs. The the jth fuzzy rule the controller is as follows:

$$R^j\text{: } If\ \left\{Q_1 + Q_3\ is\ F_1^j\right\},\ \left\{Q_2 + Q_4\ is\ F_2^j\right\},$$
$$\left\{W_1 + W_3\ is\ F_3^j\right\}\ and\ \left\{W_2 + W_4\ is\ F_4^j\right\}$$
$$Then\ \left\{S_1\ and\ S_3\ are\ G^j\ and\ S_2\ and\ S_4\ are\ not\ G^j\right\}$$

The parameters $W_i + W_{i+2}$ and $Q_i + Q_{i+2}$ for $i = 1{,}2$ are specified by fuzzy membership functions as demonstrated in Figs. 5–7. Consequently, the number of fuzzy rules for the controller is computed as $3 \times 3 \times 3 \times 3 = 81$. Some examples of fuzzy rules of the controller are provided in Table 1. Another part of the fuzzy controller is inference engine. An inference engine is a computer program that attempts to obtain answers from a Rule-base. It is the "brain" that expert systems use to reason about the data in the Rule-base for the ultimate purpose of formulating new results.

The last part of designing a fuzzy controller is defuzzification. The defuzzification is the conversion of a fuzzy quantity to a precise quantity, just as fuzzification is the conversion of a precise quantity to a fuzzy quantity. But before applying the defuzzification, in type-2 fuzzy sets and systems we have another part, namely type-reducer. A type-reducer is required to convert them

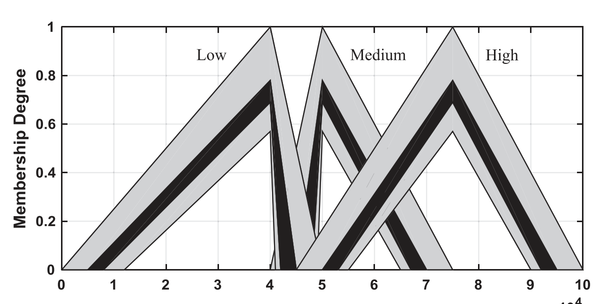

Fig. 5. Fuzzy membership functions for $W_1 + W_3$, $W_2 + W_4$.

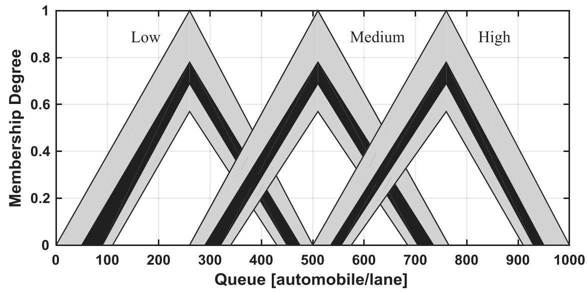

Fig. 6. Fuzzy membership functions for $Q_1 + Q_3$, $Q_2 + Q_4$.

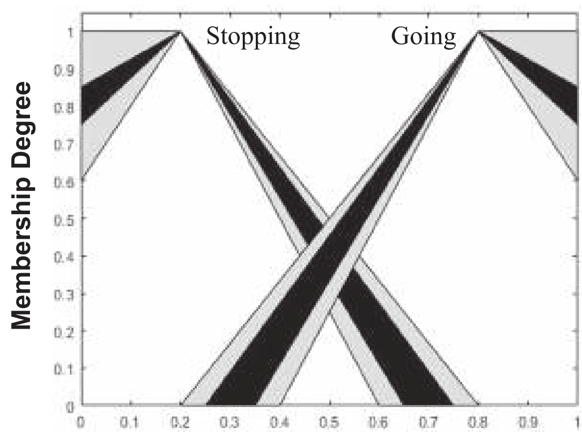

Fig. 7. Fuzzy membership functions for S_1, S_2, S_3, and S_4.

Table 1
Example of fuzzy rules of the proposed controller.

Rule j	F_1^j	F_2^j	F_3^j	F_4^j	G^j
1	Low	Low	Low	Low	Going
2	Low	Low	Low	Medium	Stopping
3	Low	Low	Low	High	Stopping
4	Low	Low	Medium	Low	Going
5	Low	Low	Medium	Medium	Going
19	Low	Medium	Medium	Low	Stopping
51	Medium	High	High	Low	Stopping
79	High	High	High	Low	Going
80	High	High	High	Medium	Going
81	High	High	High	High	Going
…	…	…	…	…	…

into a T1 FS before defuzzification can be performed. The precise numerical amount of the automobiles that join the ith queue in the nth time interval for the specified inputs is computed using Eq. (37). General scheme of the proposed controller is shown in Fig. 8.

5. Discussion of results

The proposed controller can appropriately control traffic stream under both regular and abnormal traffic circumstances. In simulation, the sampling time is considered 5 s ($T = 5$), while a full cycle time of traffic light is considered 100 s. Simulation has been done with the following assumptions:

a) The crossroad is composed of four ways, and each way consists of three lanes.
b) The entrance of automobiles into crossroad is independent on each lane.
c) Pedestrian crossing at the crossroad is embedded.

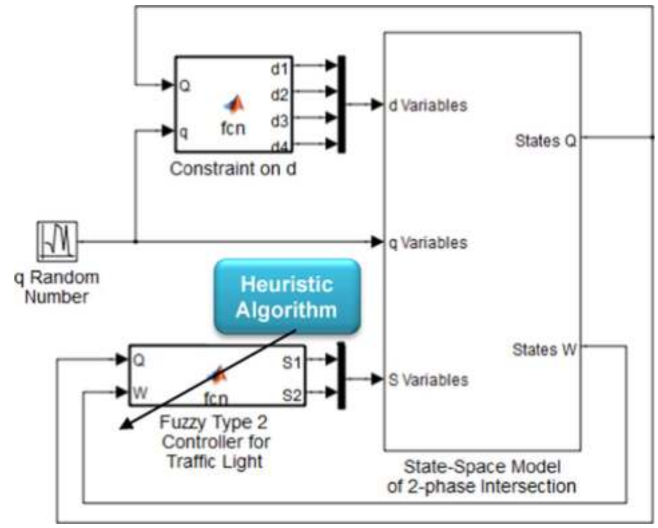

Fig. 8. General scheme of the proposed controller.

d) Sensors are located at specified distance from the crossroad, the highest number of automobiles in the queue that can be identified by sensors, is 30 automobiles.
e) Maximum and minimum time to cross the crossroad is respectively 40 and 5 s.

The number of automobiles that leave the ith queue in the nth time interval is obtained by the following equation

$$d_i(n)=\min\left(Q_i(n) + q_i(n),\, d_{si}(n)\right) \tag{41}$$

So that saturation stream rate is

$$d_{si}(n) = d_{cons.}(n)+\beta q_i(n) \tag{42}$$

for $i = 1,2,3,4$. The $d_{cons.}$ parameter is greater equal fifty, ($d_{cons.}\geq 50$). The β parameter is in the interval [0 1], so that it's changes are listed in Table 2. In fact, the β parameter indicates the level of uncertainty, where $\beta = 0$ corresponds to the highest possible uncertainty, while $\beta = 1$ corresponds to the situation in which the exact travel time through the network is known with complete certainty. The traffic data was saved every 5 s and was utilized in the simulations. The q_i and d_i variables are random signals with normal distribution.

5.1. Results of fixed-time monitoring

The traffic signals of Leg 1 and Leg 3 in Fig. 4 were supposed green in 140 s and red in after 60 s. On the other side, the traffic signals of Leg 2 and Leg 4 in Fig. 4 were supposed red in 140 s and green in after 60 s. The purposes of optimization are length reduction of queues and reducing the average waiting time of automobiles in the queues. The total numbers of automobiles in queues on crossroad were shown in every 5 s in Fig. 9. The simulation time is 1000 s.

Table 2
Level of uncertainty presented by β parameter.

The Position of Traffic	β
Non Saturation	$\beta \geq 0.7$
Saturation	$0.4 \leq \beta \leq 0.6$
Super Saturation	$0.1 \leq \beta \leq 0.3$
Instable	$\beta=0$

Please cite this article as: Khooban MH, et al. An optimal general type-2 fuzzy controller for Urban Traffic Network. ISA Transactions (2016), http://dx.doi.org/10.1016/j.isatra.2016.10.011

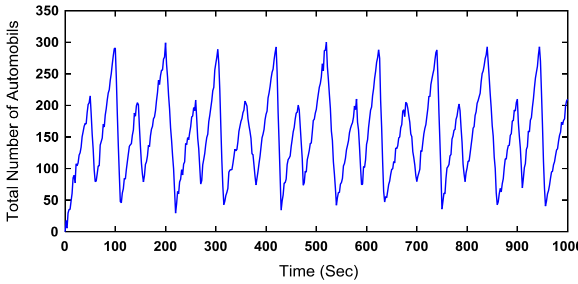

Fig. 9. The total number of automobiles in queues on crossroad without controller.

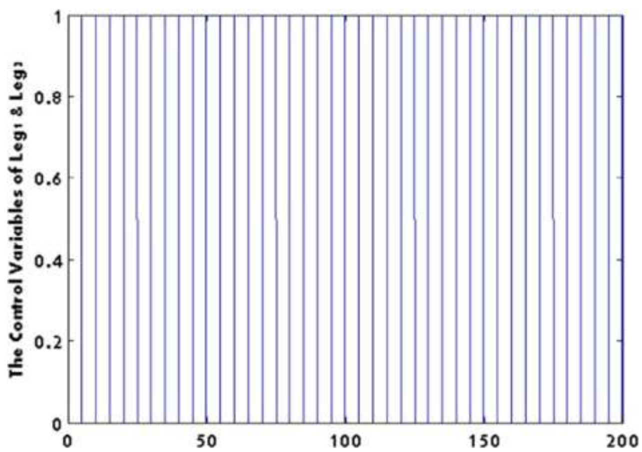

Fig. 10. The control variables for Leg 1 and Leg 3.

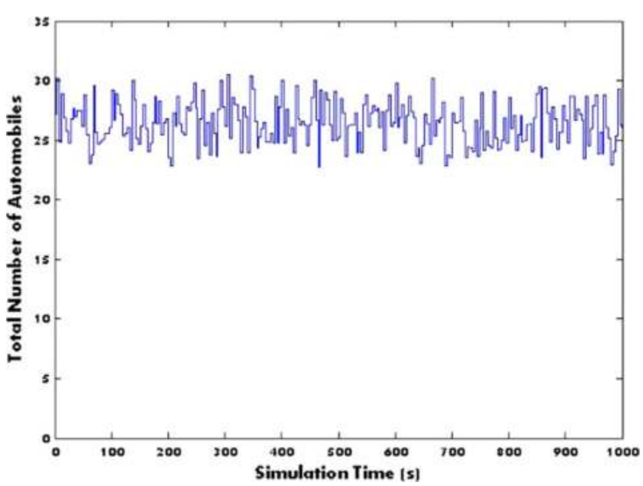

Fig. 11. The total number of automobiles in queues on crossroad with (OT2FC).

5.1.1. Results of fuzzy intelligent control

The output of controller is the set of control variables (S_i). These control variables for the Leg 1 and Leg 3 of crossroad were shown in Fig. 10. Also, the total numbers of automobiles in queues on crossroad, when the fuzzy controller has been used, were shown in every 5s in Fig. 11. The simulation time is 1000 s.

In order to show the superiority of proposed method, the Table 3 is presented, where it can be observed that in most tests the proposed model has a better performance compared with other methods. The results in this section show that the non-optimal and optimal type-1 fuzzy controllers and the interval Type-2 fuzzy controller can still work but there exists room for improvement. On the other hand, the Optimal General Type-2 Fuzzy Controller resulted on length reduction of queues and reduction of the average waiting time of automobiles in the queues.

Table 3
The Comparison between average waiting in any lane of (OGT2FC).

Average Waiting Time (s)	Queue 1	Queue 2	Queue 3	Queue 4	Summation of Queues
Fixed Time Control	11,000 s	5000 s	11,000 s	5000 s	32,000 s
The Controller Presented in [30]	1350 s	1335 s	1300 s	1314 s	5300 s
Optimal Type-1 Fuzzy Control	1215 s	1295 s	1250 s	1273 s	5033 s
Interval Type-2 Fuzzy Control	1101 s	1173 s	1127 s	1149 s	4550 s
Optimal Type-2 Fuzzy Control	1005 s	985 s	910 s	930 s	3830 s
Improvement Percentage	90.86	80.3	91.72	81.4	88.125

The proposed strategy can achieve control goals with faster response and smoother manner. The proposed method can increase capacity of the road network in acceptable time and decrease congestion. Moreover, the computing time of proposed control strategy is less than 5sec.

6. Conclusion

In this paper an Optimal General Type-2 Fuzzy Controller (OGT2FC) was proposed for a single crossroad. The parameters of input and output membership functions were optimized simultaneously by a novel heuristic algorithm namely Modified Backtracking Search Algorithm (MBSA). The overall purpose of the optimization was length reduction of queues and reduction of the average waiting time of automobiles in the queues. Simulation results show the superiority of the proposed controller to the non-optimal and optimal type-1 fuzzy and the interval type-2 controllers (Table 3).

Acknowledgements

The authors would like to thank the reviewers for their very useful comments, which help us to improve the quality of the paper. The reviewers' efforts are gratefully appreciated. The authors wish to thank Prof. Jerry M. Mendel and Dr. Maryam Homayounzadeh for their assistance in the present work.

References

[1] B. De Schutter, Optimal traffic light control for a single intersection. In: Proceedings American control conference, p. 2195–2199; 1999.
[2] Diakaki C, Papageorgiou M, Aboudolas K. A multivariable regulator approach to traffic-responsive network-wide signal control. Control Eng Pract 2002;10:183–95.
[3] K. Iwaoka, T. Otokita and S. Niikura, Optimization strategy on urban signal control. In: Proceedings the 8th world congress on ITS '01; 2001.
[4] Oda T, Leo J. Optimization of coordinated traffic signal timings in urban road network. Trans ISCIE 1998;11(7):364–74 [in Japanese].
[5] L.A. Klein, Traffic parameter measurement technology evaluation. In: Proceedings of the IEEE-IEE vehicle navigation and information systems conference, 1993, pp. 529–533.
[6] F. Woelk, S. Gehrig, and R. Koch, A monocular image based intersection assistant". Intelligent vehicles symposium 2004 , IEEE, 2004, pp. 286–291.
[7] Chen XF, Shi ZK. A dynamic optimization method for traffic signal timings based on genetic algorithm. [Chinese Association for System Simulation.Beijing]. J Syst Simul 2004;16(6):1155–61.
[8] X.F. Chen, Z.K. Shi, Real-coded genetic algorithm for signal timing optimization of a single intersection. In: Proceedings of 2002 international conference on machine learning and cybernetics, vol 3, 2002, pp.1245–1248.
[9] S. Takahashi, H. Nakamura, H. Kazama, and T. Fujikura, Genetic algorithm approach for adaptive offset optimization for the fluctuation of traffic flow. In: IEEE Proceedings of the 5th International Conference on Intelligent

Please cite this article as: Khooban MH, et al. An optimal general type-2 fuzzy controller for Urban Traffic Network. ISA Transactions (2016), http://dx.doi.org/10.1016/j.isatra.2016.10.011

Transportation Systems, 2002, pp. 768–772.
[10] Y. Wakasa, K. Iwaoka and K. Tanaka, Modelling and robust control of traffic signal systems. In: Proceedings European Control Conference 2003 (CD-ROM), 2003.
[11] Y. Wakasa, K. Hanaoka, K. Iwaoka and K. Tanaka, Controllability of network-wide traffic signal systems and state-feedback controller tuning. In: Proceedings 11th World Congress on ITS Nagoya, Aichi 2004, Japan (CD-ROM) 2004.
[12] Chinyere OU, Francisca OO, Amano OE. Design and simulation of an intelligent traffic control system. IJAET 2011;1(Issue 5):47–57.
[13] Mendel J. Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Upper Saddle River, NJ: Prentice-Hall; 2001.
[14] John R. Type 2 fuzzy sets: an appraisal of theory and applications. Int J Uncertain Fuzziness Knowl Based Syst 1998;6(6):563–76.
[15] Chiclana F, Zhou S-M. Type-reduction of general type-2 fuzzy sets: the type-1 OWA approach. Int J Intell Syst 2013;28(5):505–22.
[16] Ghaemi Mostafa, Hosseini-Sani Seyyed Kamal, Khooban Mohammad Hassan. Direct adaptive general type-2 fuzzy control for a class of uncertain non-linear systems. IET Sci, Meas Technol 2014;8(6):518–27.
[17] Zadeh LA. The concept of a linguistic variable and its application to approximate reasoning–I. Inf Sci 1975;8(3):199–249.
[18] Khooban Mohammad, Hassan, Niknam Taher. A new and robust control strategy for a class of nonlinear power systems: adaptive general type-II fuzzy. Proc Inst Mech Eng, Part I: J Syst Control Eng 2015 [0959651815571621].
[19] Liu F. An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. IEEE Comput Intell Soc Walter J Karplus Summer Res Grant Rep 2006.
[20] Wagner C, Hagras H. Toward general type-2 fuzzy logic systems based on zSlices. Fuzzy Syst, IEEE Trans 2010;18(4):637–60.
[21] Bilgin, A., Hagras, H., van Helvert, J. and Alghazzawi, D., A linear general type-2 fuzzy logic based computing with words approach for realising an ambient intelligent platform for cooking recipes recommendation.
[22] Kumbasar T, Hagras H. A self-tuning zSlices-based general type-2 fuzzy PI controller. Fuzzy Syst IEEE Trans 2015;23(4):991–1013.
[23] Mendel JM. Uncertain rule-based fuzzy logic systems: introduction and new directions. Upper Saddle River, NJ: Prentice-Hall; 2001.
[24] Liu F. An efficient centroid type-reduction strategy for general type-2 fuzzy logic system. Inf Sci 2008;178(9):2224–36.
[25] Karnik NN, Mendel JM. Centroid of a type-2 fuzzy set. Inf Sci 2001;132:195–220.
[26] Mendel JM, Wu D. Perceptual computing: aiding people in making subjective judgments. Hoboken, NJ: Wiley-IEEE Press; 2010.
[27] Wu D, Mendel JM. Enhanced karnik-mendel algorithms. IEEE Trans Fuzzy Syst 2009;17(4):923–34.
[28] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 2013;219(15):8121–44.
[29] Shaw B, Mukherjee V, Ghoshal SP. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. Int J Electr Power Energy Syst 2012;35(1):21–33.
[30] Azimirad E, Pariz N, Naghibi Sistani MB. A novel fuzzy model and control of single intersection at urban traffic. [March]. Netw, IEEE Syst J 2010;4(1):107–11 [March].
[31] Khooban Mohammad, Hassan Davood Nazari Maryam, Abadi Alireza, Alfi, Siahi Mehdi. Swarm optimization tuned Mamdani fuzzy controller for diabetes delayed model. Turk J Electr Eng Comput Sci 2013;21(Suppl 1):2110–26.
[32] Khooban Mohammad Hassan. Design an intelligent proportional-derivative PD feedback linearization control for nonholonomic-wheeled mobile robot. J Intell Fuzzy Syst: Appl Eng Technol 2014;26(4):1833–43.
[33] Alfi Alireza, Kalat Ali Akbarzadeh, Khooban Mohammad Hassan. Adaptive fuzzy sliding mode control for synchronization of uncertain non-identical chaotic systems using bacterial foraging optimization. J Intell Fuzzy Syst 2014;26(5):2567–76.
[34] Modiri-Delshad M, Rahim NA. Solving non-convex economic dispatch problem via backtracking search algorithm. Energy 2014;77:372–81.
[35] Talab Hamid, Saadat Hadiseh, Mohammadkhani, Haddadnia Javad. Controlling multi variable traffic light timing in an isolated intersection using a novel fuzzy algorithm. J Intell Fuzzy Syst 2013;25(1):103–16.