Delft University of Technology

Delft Center for Systems and Control

Technical report 15-007

Urban traffic control using a fuzzy multi-agent system*

A. Jamshidnejad, B. De Schutter, and M.J. Mahjoob

If you want to cite this report, please use the following reference instead:
A. Jamshidnejad, B. De Schutter, and M.J. Mahjoob, "Urban traffic control using a fuzzy multi-agent system," *Proceedings of the 2015 European Control Conference*, Linz, Austria, pp. 3046–3051, July 2015.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/15_007.html`

Urban traffic control using a fuzzy multi-agent system

Anahita Jamshidnejad[1], Bart De Schutter[1], and Mohammad J. Mahjoob[2]

Abstract— **This paper presents a fuzzy multi-agent system for control of the traffic signals of an urban network, where the aim is to reduce the total average delay time of the vehicles. The large-scale traffic system is first divided into sub-areas and an agent, using a fuzzy controller, is considered for each sub-area. In order to develop the fuzzy rule bases for the agents, an extensive set of data is collected and the corresponding origin-destination (OD) matrices are calculated. The OD matrices are then clustered using a new clustering algorithm proposed in this paper. Finally, the resulting cluster centers, which are matrices of the same dimension as the OD matrices, are mapped to a 2-dimensional space and the corresponding triangular fuzzy sets are extracted. An optimal cycle plan is found for each fuzzy set and this way the fuzzy rule bases are constructed (the triangular fuzzy sets form the IF-parts and the optimal cycle plans form the THEN-parts). In a case study the proposed multi-agent control system is applied to a microscopic urban traffic network and the results are compared with a controller that applies optimal signal plans calculated by PASSER V. The results show that the proposed fuzzy multi-agent system outperforms the non-fuzzy control system, where the average delay time of the traveling vehicles is decreased by 19% using the multi-agent control system.**

I. INTRODUCTION

Mobility is increasing rapidly in different areas of the world as a result of population growth and technology developments. As a result, efficient methods are required to manage the traffic such that its side-effects such as traffic congestion, accidents and injuries, waste of resources, noise/air pollution can be reduced. In order to reduce the traffic problems, various strategies have been proposed, including enhancement of road capacity through construction of new roads, improvement of public transportation systems, road pricing and congestion charging, and development of modern Intelligent Transportation Systems (ITS).

The idea of intelligent transportation systems (ITS) was initiated in the late 1980s to automatically control traffic [1]. ITS incorporate traffic management measures such as on-line route guidance and real-time information exchanges among the road users [2], [3], variable speed limits, dynamic signal planning, and use of high occupancy vehicle lanes.

Multi-agent architectures have widely been considered for traffic management and ITS, since they can deal with the large-sized traffic systems. In [4] two different multi-agent traffic control systems (TRYS and TRYSA2) were introduced for motorways, where one of the architectures uses a decentralized coordination approach among the agents and the other one uses a centralized approach. Another multi-agent control system suggested for motorways is presented in [5] where a hierarchical centralized coordination algorithm for the agents is considered. An adaptive multi-agent control system based on ant colony behavior is proposed in [6].

In a multi-agent architecture for urban traffic networks, there are two options: the first is that the control instruments (e.g., the traffic signals) are supposed to be intelligent agents, while the system users (the vehicles) are not considered as intelligent agents and do not directly exchange information with other agents [7], [8]; the second option is that both the controllers and the users of the system are considered as agents that communicate with each other [9], [10].

Soft computing techniques, including fuzzy systems, have been widely used and shown great performance in traffic control applications [11], [12], [13].

This paper introduces a multi-agent fuzzy control system for an urban traffic network, where the traffic network is divided into sub-areas each composed of an intersection and the corresponding traffic signals and sensors that count the flows on the entrances and exits of the intersection. Each intersection is controlled by an agent that uses a fuzzy controller. The process of designing the agent-based control system for each sub-area includes collecting and clustering a huge number of data in an off-line stage. A fuzzy rule base is then constructed and a fuzzy control system is designed. The agents will finally exchange information with their neighboring agents in order to make the best global decision.

The main contributions of the paper include:

- introducing a new clustering algorithm to cluster the OD matrices such that the resulting clusters have small total within-group variance and are also well separated,
- proposing a mapping method for the OD matrices to reduce their size and to consequently reduce the number of fuzzy sets that are required for the rule bases,
- considering a MISO fuzzy inference engine that is based on estimates of the entering flows to the intersection in the near future.

The remainder of the paper is organized as follows: Section II presents the off-line stage in detail, where the process of constructing the agent-based fuzzy control system for each intersection is discussed. Section III explains how the designed fuzzy controllers are applied on-line via a multi-agent architecture. Section IV illustrates the proposed approach based on a network with 25 intersections and presents the corresponding results. Finally, the conclusions are given in Section V.

[1]Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands {a.jamshidnejad , b.deschutter}@tudelft.nl

[2]School of Mechanical Engineering, University of Tehran, Iran mmahjoob@ut.ac.ir

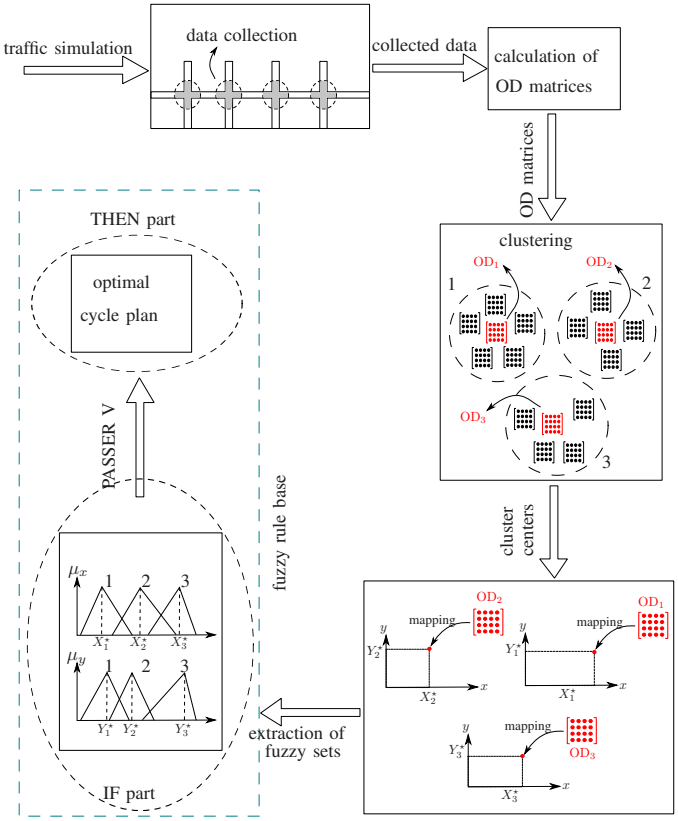

Fig. 1. Steps of the off-line stage for creating the fuzzy rule bases

II. Off-line stage

In this section we will explain the off-line stage, where a fuzzy controller for each agent is constructed. This stage consists of six steps that are demonstrated by Fig. 1. These six steps of the off-line stage include:

1) collecting an extensive data set
2) calculating the origin-destination (OD) matrices from collected data,
3) clustering the resulting OD matrices and finding the cluster centers,
4) mapping the matrices into a 2-dimensional space,
5) extracting a type-1 fuzzy set for each cluster (the fuzzy sets will be called the *traffic patterns* of the intersection),
6) finding the optimized signal plan for each traffic pattern.

The following sections will explain these six steps in more detail.

A. Data collection

As the first step of constructing an agent-based fuzzy controller, for each intersection we collect an extensive set of traffic flow data. The data set should be collected carefully and in an extended period of time such that it covers different traffic scenarios (peak and off-peak, free and congested flow, etc.).

B. Data Analysis: calculation of the local OD matrices

As the next step, we determine the origin-destination (OD) matrices from the collected data. For each intersection and for every sampling step, an OD matrix is calculated. This yields a matrix of dimension $N_{\mathrm{e}} \times N_{\mathrm{o}}$ with N_{e} and N_{rmo} being the number of entrances and exits of the intersection respectively. The $(i,j)^{\mathrm{th}}$ entry of the OD matrix at sampling step l, denoted by $g_{i,j}(l)$, is the flow from entrance i through exit j that enters the intersection during one sampling step. To generate the OD matrices from the collected data, the method described in [11] can be used.

Now, if we define $b_{i,j}(l)$ as the probability that a vehicle observed at entrance i in period $[l-p,l]$ will leave the intersection through exit j, with

$$\begin{aligned} & b_{i,j}(l) \geq 0, \\ & \sum_{j=1}^{N_o} b_{i,j}(l) = 1 \end{aligned} \qquad \text{for} \qquad \begin{aligned} & i = 1,\ldots,N_{\mathrm{e}} \\ & j = 1,\ldots,N_{\mathrm{o}} \end{aligned} \tag{1}$$

then the following optimization helps us find $g_{i,j}(l)$:

$$\min f = \min \left[\sum_{s=l-p}^{l} \sum_{j=1}^{N_{\mathrm{o}}} \left(y_j(s) - \sum_{k=1}^{N_{\mathrm{e}}} q_k(s) b_{k,j}(l) \right)^2 \right] \tag{2}$$

where p is the number of previous sampling steps that is selected for estimating the current traffic conditions, and $q_k(s)$ and $y_j(s)$ show the entering and exiting flows from entrance i and exit j during a time interval of length s.

Finally, $g_{i,j}(l)$ for $i = 1,\ldots,N_{\mathrm{e}}$ and $j = 1,\ldots,N_{\mathrm{o}}$ will be obtained as:

$$g_{i,j}(l) = b_{i,j}(l) \cdot q_i(l) \tag{3}$$

C. Clustering the OD matrices

The calculations of Section II-B will result in a huge number of OD matrices, since an OD matrix is calculated for each intersection and for each sampling step. Therefore, the next step will be to cluster this huge number of matrices and to find the OD matrices that correspond to the same general traffic pattern. Another benefit of clustering the OD matrices is that if new intersections are added to the network, the corresponding traffic patterns might belong to the existing clusters; otherwise, we can add new clusters with no need to change the structure of the control system.

In [14] three clustering approaches are discussed: *hierarchical*, *k-means*, and *two-stage* clustering methods. Based on the discussions, for very large data sets a two-stage clustering algorithms is the most beneficial one. Therefore, we use a two-stage clustering algorithm for the OD matrices, where the first stage is inspired by [15], and the second stage uses the hierarchical clustering algorithm based on fuzzy graph connectedness proposed in [16].

The same two-stage clustering approach has been applied to the traffic OD matrices in [11]. However, we have made changes to the algorithm applied in the first stage to improve the results of the clustering by producing clusters that contain more similar matrices, while the clusters are well separated.

In the proposed algorithm all vertices are put in one cluster initially and there is no need to consider any special ordering.

Like [15], we start by constructing a weighted linkage graph (WLG) from the data set. Each vertex of the WLG corresponds to an OD matrix and the edge connecting two vertices represents the *degree of similarity* between the two vertices/OD matrices. Similarity is defined as the inverse of the Euclidean distance of the two OD matrices:

$$S(\mathrm{v}_v, \mathrm{v}_u) = \left(\sqrt{\sum_{i=1}^{N_e} \sum_{j=1}^{N_o} \left(\mathrm{v}_v(i,j) - \mathrm{v}_u(i,j)\right)^2} \right)^{-1} \quad (4)$$

where v_v and v_u are two vertices/OD matrices of the WLG, and S is the degree of similarity between these two vertices.

Both the algorithm given in [15] and the algorithm proposed here will result in two clusters at each step. Therefore, the algorithms will be applied iteratively to create a predefined number of clusters. Considering two separate clusters C_1 and C_2 that together contain all the vertices, the clustering algorithm will make the vertices move between the clusters based on their "gain" values, where the gain for a vertex $\mathrm{v}_v \in \mathrm{C}_1$ is defined as:

$$\mathrm{gain}(\mathrm{v}_v) = \sum_{\mathrm{v}_u \in \mathrm{C}_2} S(\mathrm{v}_v, \mathrm{v}_u) - \sum_{\mathrm{v}_w \in \mathrm{C}_1} S(\mathrm{v}_v, \mathrm{v}_w) \quad (5)$$

For initial partitioning in [15], the criterion for putting two vertices in the same cluster is whether or not their degree of similarity given by (4) is greater than or equal to a predefined similarity threshold. Moreover, the criterion based on which the algorithm decides whether to move a vertex from its current cluster to the other cluster, is the gain value of the vertex defined by (5).

Instead, for the initial partitioning we first put all vertices in a single cluster and then let them one by one move to the other clusters based on their gain value as is given by the following algorithm:

1) Suppose V is the set of all vertices and construct two separate clusters $\mathrm{C}_1 = V$ and $\mathrm{C}_2 = \emptyset$
2) Move vertices from C_1 to C_2 one by one based on the maximum gain values calculated by (5)
3) For the i^{th} point moved store the gain:

   $$g_i = \mathrm{gain}\left(i^{\mathrm{th}} \text{ point moved}\right)$$

   and store the corresponding clusters:

   $$\mathrm{C}_{1,i} \leftarrow \mathrm{C}_1$$
   $$\mathrm{C}_{2,i} \leftarrow \mathrm{C}_2$$

   Continue step 3 until $\mathrm{C}_1 = \emptyset$ and $\mathrm{C}_2 = V$
4) Choose the clusters $\mathrm{C}_{1,i^\star}$, $\mathrm{C}_{2,i^\star}$, where

   $$i^\star = \min \left(\underset{i=2,\ldots,|V|}{\arg\min} \left(g_i - g_{i-1}\right) \right)$$

5) Keep $\mathrm{C}_{1,i^\star}$ and go to step 1 with $V = \mathrm{C}_{2,i^\star}$

After the clusters are found, we calculate the cluster centers. We first normalize the OD matrices with respect to the maximum and minimum flow matrices. The cluster centers are then found by averaging the corresponding normalized entries of all matrices in each cluster. From now, we use the center of a cluster as a representative of that cluster.

D. Extraction of type-1 fuzzy sets

In this step, we present each of the resulting clusters by type-1 fuzzy sets, called traffic patterns. We first check what is the least number of fuzzy sets required to represent each cluster. In the literature such as [17] and [18] only one fuzzy set has been considered for each matrix and all the entities of the matrix have been averaged. This way the matrix was mapped to a one-dimensional space. However, if we map the OD matrices to a one-dimensional space many of the different matrices will be mapped to the same point.

Instead, we just average the entries of the OD matrix that represent the flow values that are controlled by the same traffic signal. Finally, we reduce the number of fuzzy sets to N_f, where $N_\mathrm{f} \leq N_\mathrm{o}$. Then, N_f triangular fuzzy sets will be constructed using the common methods in the literature such as in [17], where the centers of these sets are the N_f resulting cumulative flows.

Suppose that N_f is 2, and the matrices are mapped into a two-dimensional space (this happens for an intersection with two-lane streets and a traffic light located at each entering lane), then the two averaged entries/flows are denoted by $X_i^\star(l)$ and $Y_i^\star(l)$ for the i^{th} cluster/OD matrix.

E. Finding the optimized signal timing plan

A fuzzy rule base is composed of fuzzy rules of the form:

$$\text{IF } x_1 \text{ is } A_1 \ \&\ \ldots \&\ x_n \text{ is } A_n \text{ THEN } y \text{ is } B$$

where $x = (x_1, \ldots, x_n)$ and y are linguistic variables (the input and the output respectively), and $A_i,\ i = 1, \ldots, n$ and B are linguistic values determined using fuzzy sets. In our application, the traffic patterns found for each intersection form the IF-parts of the fuzzy rules. The THEN-parts include the optimal cycle plans, which could be found using available traffic software (e.g., Passer V [19]). To find the optimal cycles, traffic flows on the lanes are needed which are given for each cluster by its center. A cost function , e.g., the total travel time of the vehicles, is minimized and the corresponding cycle plan forms the THEN-part.

III. On-line stage

In the previous section we explained how the fuzzy controllers are constructed for each intersection in an off-line stage. In this section, we consider the problem of controlling the traffic in an urban area that consists of a number of intersections, where we should apply the controllers on-line via a multi-agent architecture. The on-line stage consists of four main steps illustrated by Fig. 2. These steps include:

1) real-time data collection and information exchange for each intersection and for every sampling step,
2) making an estimate of the traffic flow in the near future, i.e., T_s time units from now, based on the collected and exchanged data and calculating the OD matrix,

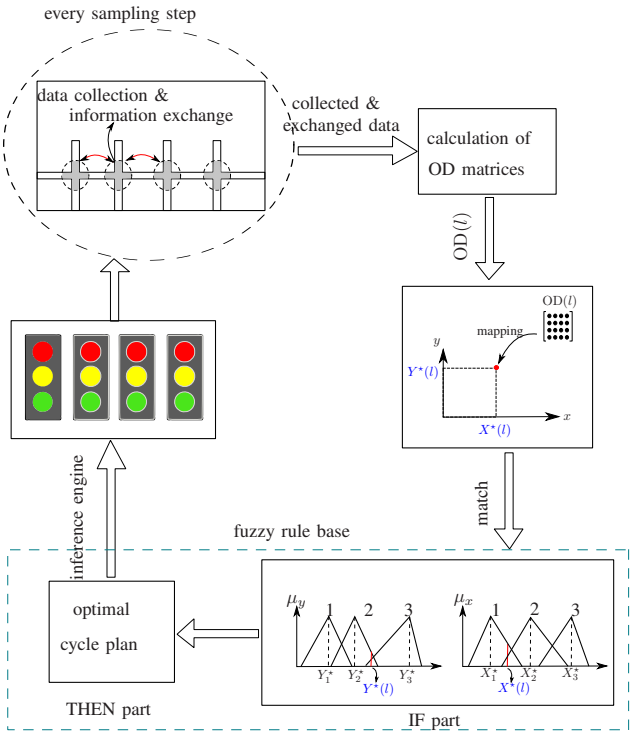

Fig. 2. On-line control of traffic signals using the fuzzy rule bases

3) mapping the OD matrix corresponding to each intersection into a 2-dimensional space,
4) finding the optimal cycle plan using a fuzzy inference engine that matches the resulting values in the 2-dimensional space with the fuzzy rule base of that intersection.

A. Real-time data collection and information exchange

At this step, every agent makes use of the data collected by its local sensors, and the data it receives from the agents that are corresponding to the neighboring intersections (since the flow on the neighboring intersections will affect the flow of the agent's intersection in the near future).

B. Flow estimation in near future & finding the OD matrices

Based on the collected information, the agent makes three estimates about its local traffic situation in the near future, i.e., T_s time units from now. For the first estimate, the traffic flow on all entrances and exits of the intersection in the near future is assumed to be the same as the last reported data at these locations. The second estimate is based on the information the agent receives from the neighboring agents, where the entering flow through a specific lane in the near future is assumed the same as the exiting flow of its neighboring intersection through the same lane. The third estimate is the average of the first and the second estimates. These three estimates are not necessarily close in value unless the traffic is distributed uniformly through the lane. The difference between these three estimates becomes more significant when there is a disturbance such as a car crash in the corresponding lane.

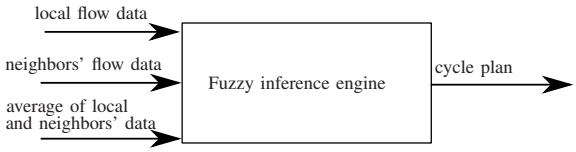

Fig. 3. MISO fuzzy engine used by each agent

Finally, three different OD matrices are calculated for the three estimated flow scenarios (see Section II-B).

C. Mapping the OD matrices

Next, the three OD matrices are normalized with respect to the maximum and minimum flows found in the off-line stage. Then the normalized matrices are mapped to the two-dimensional space (explained in Section II-D), and the two resulting values are denoted by X_j^* and Y_j^*, for $j = 1, 2, 3$.

D. Using the MISO fuzzy inference engine

The resulting values $X^\star$ and $Y^\star$ are sent to the fuzzy inference engine as inputs (see Fig. 3). The fuzzy inference engine then uses the constructed fuzzy rule bases in the off-line stage to determine the membership degree of these values for each of the triangular fuzzy sets.

Finally, the fuzzy inference engine produces the optimal cycle plan for the three pairs of mapped OD matrices. At the end, the fuzzy inference engine selects the final timing plan as the maximum value of the cycle plans.

IV. Simulation and Results

In this section, we will present the simulation results for the proposed fuzzy multi-agent control system. First, we apply the off-line stage explained in Section II to construct the fuzzy rule bases for each intersection. Then for the on-line stage explained in Section III, the agents will exchange their information and choose the best signal plans for their intersections.

A. Simulations for the off-line stage

1) Set-up: An urban traffic network consisting of 25 sub-areas is considered, where each sub-area contains:

- an intersection consisting of two-lane roads with four entrance and four exit lanes, and a sensor located at each entrance and each exit of the intersection (see Fig. 4)
- four traffic lights at the entrances
- an agent that controls the traffic lights of the sub-area using a fuzzy controller

Note that vehicles can make any turns except U-turns at the intersection. Every two oppositely located traffic lights have the same timing plan.

2) Data collection: To simulate the multi-agent fuzzy control system and to provide an environment for modeling the traffic, for collecting data, and for evaluating the control system, a microscopic traffic simulator has been developed within NetLogo, an agent-based environment in Java. Different times of day and weekdays were modeled changing the flow on the lanes. Traffic flow at the location

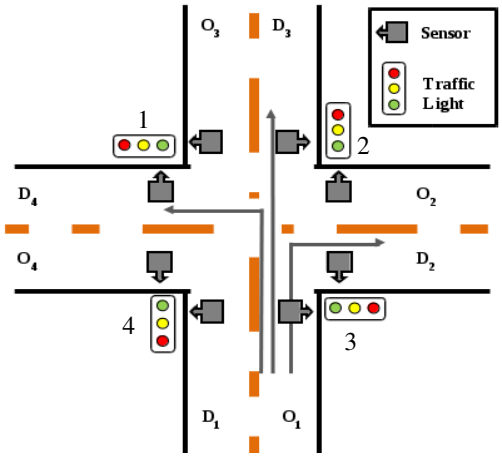

Fig. 4. Each intersection containing four entrances and exits

of the sensors was collected in a 14-hour simulation, where data was recorded every 5 minute. The option of changing the demands during the simulation and the possibility of modeling car crashes make the scenario more realistic.

3) Calculating and clustering the OD matrices: The OD matrices for each intersection were calculated for each sampling step using the procedure explained in Section II-B. The resulting OD matrices were clustered using the two-stage clustering algorithm discussed in Section II-C. In order to evaluate the performance of the algorithm proposed in this paper and the one proposed in [15] for the traffic data set, two indicators called the "total within group variance" [20], I_1, and the "separation of clusters", I_2, were considered:

$$I_1 = \frac{1}{N_{\mathrm{C}}} \sum_{u=1}^{N_{\mathrm{C}}} \sum_{v=1}^{N_{\mathrm{D}}} \mu_{uv} ||\mathrm{v}_v - \mathrm{c}_u||^2 \tag{6}$$

$$I_2 = \min_{\substack{u=1,\ldots,N_{\mathrm{C}},\\ v=1,\ldots,N_{\mathrm{D}},\ u\neq v}} ||\mathrm{c}_v - \mathrm{c}_u|| \tag{7}$$

where N_{C} and N_{D} are the number of clusters and data points respectively, v_v denotes the v^{th} data point, and c_u is the center of the u^{th} cluster, and μ_{uv} is the membership degree of the v^{th} data to the u^{th} cluster.

A smaller total variance is desirable, since it shows that the data in the same cluster are close in values and therefore are similar. In addition, a satisfactory clustering should result in relatively bigger separation values, which indicates that the resulting clusters are well separated.

Tables I and II show the results of applying the algorithm in [15] and the clustering algorithm proposed in Section II-C for five intersections that are representative for the other 20 intersections in the network, since the results calculated for the rest of the intersections was almost the same as the results obtained for these five intersections. The total within-group variances were estimated and compared with each other, and based on a predefined separation threshold, it was decided whether or not (some of the) resulting clusters need to be merged again.

From Table I, for the algorithm in [15] for all cases merging was necessary, while for the algorithm proposed in this paper, only in one case some of the clusters needed to be merged. The total within-group variance is smaller for the proposed algorithm (see Table II) compared to the algorithm in [15].

TABLE I

RESULTS FOR THE GRAPH-BASED CLUSTERING ALGORITHM IN [15]

Intersection #	Total within group	Merging needed?	# clusters
1	36.4	Y	27
2	25.8	Y	16
3	32.3	Y	12
4	61.2	Y	13
5	71	Y	42

TABLE II

RESULTS FOR THE PROPOSED GRAPH-BASED CLUSTERING ALGORITHM IN SECTION II-C

Intersection #	Total within group	Merging needed?	# clusters
1	33.4	Y	17
2	21.2	N	20
3	30.5	N	25
4	46.6	N	18
5	39	N	20

After the clusters were found, the centers were calculated by averaging the corresponding normalized entries of the matrices with respect to the maximum and minimum observed flows. Each resulting cluster represents a traffic pattern of the corresponding intersection, where these traffic patterns were given by type-1 fuzzy sets and the center of these fuzzy sets were found following the same trend explained in Section II-D.

Since traffic lights 1 and 3 and also traffic lights 2 and 4 have the same cycle plans, the flows entering the intersection through 1 and 3, and also through 2 and 4 (see Fig. 4) should be summed.

Finally, PASSER V which needs the geometrical information of each intersection and uses an optimization algorithm (we have used GA) was used to produce the optimum cycle plans, where the objective was defined as the minimum travel time for the vehicles in the intersection.

B. Simulations for the on-line stage

The proposed multi-agent control system was evaluated for the urban traffic network described in Section IV-A.1. Since every agent needs to receive the flow information from those intersections that affect the flow of the intersection under the control of that agent, we put each intersection together with its northern, southern, eastern, and western neighbors in a group that should exchange information with each other. The entering, delay, and travel times of all vehicles were recorded by the simulator during the online stage. This data was used to calculate the average delay time of the vehicles in order to assess the control system. The entering and exiting flows were recorded by the sensors every 5 seconds.

Fig. 5 shows the average delay time vs. time for two control strategies. The blue curve corresponds to a non-fuzzy

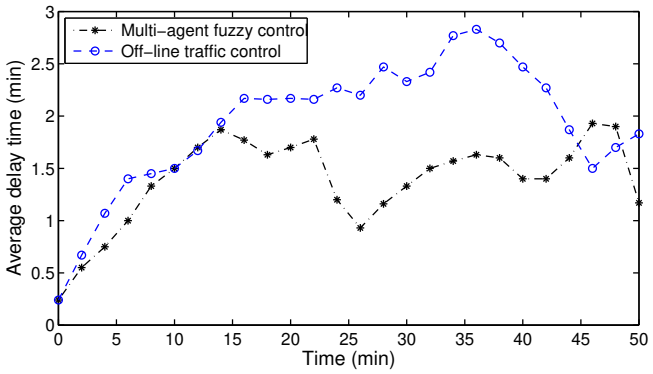

Fig. 5. Comparing a controller that uses the optimal cycles from PASSER V found off-line with the multi-agent control system proposed in this paper

control system that applies the optimal cycle plans obtained from PASSER V, and the black curve corresponds to the multi-agent control system proposed in this paper where the agents exchange information with their neighboring agents and use the resulting estimates explained in Section III.

The fuzzy multi-agent control system reduces the total average delay time of the vehicles by 19% compared with the non-fuzzy control system. It is clearly seen from Fig. 5 that the average delay is almost always smaller than the one for the off-line method.

V. Conclusions and Future Work

This paper has presented a fuzzy multi-agent control system for an urban traffic network. The agents control the traffic lights of the intersections using fuzzy rule bases that are constructed in an off-line stage. Each agent has access to the flow data of its own sub-area/intersection, and it can exchange information with its neighboring agents. Based on the estimated future flow, the agent matches this value with the IF-parts of its fuzzy rule base and decides which cycle plan should be applied using a fuzzy inference engine.

To reduce the number of OD matrices, we have proposed a clustering algorithm that results in clusters with small total within group variance, and are also well separated as is demonstrated in the case study.

We have compared the performance of the designed fuzzy multi-agent control system with a non-fuzzy controller that was designed using PASSER V for optimization of the cycle plans. Simulation results show that the fuzzy multi-agent control system reduces the total average delay time of vehicles by 19% compared with the non-fuzzy system.

We propose for future work to consider changes to the fuzzy inference engine and the type of the fuzzy sets, e.g., using a linear combination of the outputs as proposed in the Takagi-Sugeno framework, or a mixture of different types of fuzzy sets rather than triangular types. Another option for further research is to consider real-life traffic data for extraction of the fuzzy rules in the off-line stage.

References

[1] J. M. Sussman and M. S. Bronzini, "A review of: perspectives on intelligent transportation systems," *Journal of Intelligent Transportation Systems*, vol. 10, no. 2, pp. 101–102, 2006.

[2] J. L. Adler, G. Satapathy, V. Manikonda, B. Bowels, and V. J. Blue, "A multi-agent approach to cooperative traffic management and route guidance," *Transportation Research Part B*, vol. 39, no. 4, pp. 297–318, 2005.

[3] W. H. Lee, S. S. Tseng, and W. Y. Shieh, "Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system," *Information Sciences*, vol. 180, no. 1, pp. 62–70, 2010.

[4] J. Z. Hernandez, S. Ossowski, and A. Garcia-Serrano, "Multiagent architectures for intelligent traffic management systems," *Transportation Research Part C*, vol. 10, no. 5-6, pp. 473–506, 2002.

[5] B. Chen, H. H. Cheng, and J. Palen, "Integrating mobile agent technology with multi-agent systems for distributed traffic and management systems," *Transportation Research Part C*, vol. 17, no. 1, pp. 1–10, 2009.

[6] H. M. Kammoun, I. Kallel, J. Casillas, A. Abraham, and A. M. Alimi, "An adaptive multiagent system based on hybrid ant-hierarchical fuzzy model," *Transportation Research Part C*, vol. 42, no. 1, pp. 147–167, 2014.

[7] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intelligent Transportation Systems*, vol. 4, no. 3, pp. 177–188, 2010.

[8] M. Keyarsalan and G. A. Montazer, "Designing an intelligent ontological system for traffic light control in isolated intersections," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 8, pp. 1328–1339, 2011.

[9] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *Third International Joint Conference on Autonomous Agents and Multi-agent Systems*, pp. 163–172, July 2004.

[10] K. Dresner and P. Stone, "Multiagent traffic management: An improved intersection control mechanism," in *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 05)*, pp. 471–477, July 2005.

[11] E. Angulo, F. P. Romero, R. Garca, J. Serrano-Guerrero, and J. A. Olivas, "An adaptive approach to enhanced traffic signal optimization by using soft-computing techniques," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2235–2247, 2011.

[12] F. Daneshfar, J. RavanJamJah, F. Mansoori, H. Bevrani, and B. Z. Azimi, "Adaptive fuzzy urban traffic flow control using a cooperative multi-agent system based on a two stage fuzzy clustering," in *6th IEEE Vehicular Technology Conference*, pp. 1–5, April 26-29 2009.

[13] P. G. Balaji and D. Srinivasan, "Type-2 fuzzy logic based urban traffic management," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 12–22, 2011.

[14] M. J. Norusis. IBM SPSS Statistics 19 Statistical Procedures Companion, US: Pearson Education, 2011.

[15] H. Kawaji, Y. Yamaguchi, H. Matsuda, and A. Hashimoto, "A graph-based clustering method for a large set of sequences using a graph partitioning algorithm," *Genome Informatics*, vol. 12, pp. 93–102, 2001.

[16] Y. Dong, Y. Zhuang, K. Chen, and X. Taib, "A hierarchical clustering algorithm based on fuzzy graph connectedness," *Fuzzy sets and Systems*, vol. 157, no. 13, pp. 1760–1774, 2006.

[17] M. Genero, J. A. Olivas, M. Piattini, and F. P. Romero, "Knowledge discovery for predicting entity relationship diagram maintainability," in *Thirteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2001)*, pp. 203–211, June 13-15 2001.

[18] M. Genero, M. Piattini, and C. Calero, "Empirical validation of class diagram metrics," in *International Symposium in Empirical Software Engineering*, pp. 195 – 203, 2002.

[19] M. Hardy and K. Wunderlich. Traffic Analysis Tools Volume IX: Work Zone Modeling and Simulation - A Guide for Analysts, Washington, DC 20590: U.S. Department of Transportation, Federal Highway Administration, Office of Operations, 2009.

[20] C. Goutte, P. Toft, E. Rostrup, F. A. Nielsen, and L. K. Hansen, "On clustering fmri time series," *NeuroImage*, vol. 9, no. 3, pp. 298–310, 1999.