Design and implementation of an urban traffic control system for public transport using multi-agent systems

Farnaz Derakhshan[†] **and Nasrin Shahpasandi**[††]

[†, ††]Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

Summary
In this research, we aim to design and implement an agent-oriented system to prioritize public transport, particularly buses in urban intersections. In our proposed system, we assume that the bus works on the basis of a timetable on every station. In addition, at the station that is just before the intersection, based on the amount of delay and the number of passengers (estimated by using innovative weight sensor), a priority value is assigned to each bus. Then, the bus requests the intersection agent its required green light time to cross the intersection. The intersection agent also uses the estimated number of vehicles, as well as bus priority value and then produce the best and optimized timing using genetic algorithms. In this research, we proposed a new multi objective fitness for our genetic algorithms where the goal is to reduce traffic as well as determination the minimum time for traffic lights. The advantages of our proposed method are as follows: using genetic algorithm, the calculation of the best timing for traffic lights and our new proposed fitness reduces the number of calling genetic algorithm for timing of the lights by about 30 percent compared to normal genetic algorithm. In addition, taking into account the number of vehicles at intersections also improves the overall traffic situation. Also, considering the number of the bus passengers as the bus prioritization factor improves passenger satisfaction and subsequently will be effective on reducing urban traffic. We compared our result with three traffic control systems for intersections including fixed time system, no priority system and prioritize systems that used only multi-agent systems. The results show that: our proposed method reduces bus delays about 80%, compared to the fixed time and without priority methods. In addition, using our proposed method is more effective on traffic than other priority method that used only multi-agent systems method.

Key words:
Traffic control, Public transport, Multi-agent systems, Genetic algorithm, Multi objective fitness.

1. Introduction

Today, public transport has become an economic method for solving the problems of urban life, such as population growth and concerns about environmental issues, air and noise pollution [1]. In many cities, buses, is the common form of public transport. Due to high carrying capacity, buses provide efficient use of the route. So they can have a substantial contribution to reduce traffic congestion. Giving priority to public transport, particularly buses that have delay has an important role in improve the quality of bus services and traffic, increasing the speed and reliability. It would be done by the changing time of traffic lights to favor buses that are approaching intersections that normally, done by extending the period of green light or calling a green light stage. This type of giving priority to buses in many cities in America, Great Britain, Japan, France, Denmark, Sweden, Switzerland, Finland, Germany, Australia, Austria, Italy and New Zealand has been implemented [2].

According to decentralized nature of disruption data in urban traffic during the urban transport network, multi-agent systems are suitable method and pattern for urban traffic control [3].

Hence, in this article, we design a new multi-agent system to give priority to buses at intersections. The aim of giving priority to late buses is to prevent buses to stop at traffic lights.

Our proposed agent based system has five agents that one of the agents is responsible for estimating the number of vehicles in the intersection called intersection agent. Then, bus agent asks its required time from intersection agent. And intersection agent determines the appropriate timing for traffic lights, according to bus priority by using genetic algorithm. In this system, the delay and the number of bus passengers are used for calculating the bus priority too. Since one of the objectives of the control of public transport is passenger satisfaction, considering these criteria would be very effective. Our proposed system avoids delays buses, and also controls general traffic at the same time.

After this introduction, this paper is organized as follows: in Section 2, a short background of the research is explained. Then, in Section 3, we describe the traffic load calculation methods and tools. In Section 4, we discuss our proposed approach, followed by the evaluation of our work in Section 5. Finally, we end up with conclusion in Section 6.

Manuscript received October 5, 2016
Manuscript revised October 20, 2016

2. Background of Research

Bus priority at traffic lights can be done in two ways: passive or active [4]. The passive way is an off-line regulation in favor of buses by giving more green light time to routes with more buses. In the active approach, giving priority to buses is given with respect to bus information such as delay [4]. There are several ways to give priority to buses including [2]: first, renew and extend the green light time, second, calling a green light phase, third, always green for the bus (if the traffic light is red, switch to green light and if traffic light is green remains green until passing the bus , forth, the predictions of bus arrival time and makes green lights for that time, finally the fifth method, creating a new stage in the lights as soon as a bus arrived (this method can include skipping or delaying other stages).

One of the basic steps in bus priority is to enable intersections making decision and timing of the traffic lights. Timing of the traffic lights can be done by two methods: Fixed time or intelligent methods.

2.1 Fixed time method

Control by the fixed time is the simplest and most basic way to control traffic lights. In this method, at all hours of the day, fixed timing independent of the crowded or deserted intersection traffic will apply to traffic lights. The main disadvantages of this system is unnecessary delays, increased travel time or non-allocation of sufficient time at peak hours. Although this method is the cheapest method in terms of traffic control is totally unacceptable and rejected. This method is offline that will not be able to respond to traffic fluctuations.

2.2 Intelligent method

Traffic control by using intelligent method began in 1990 that used learning techniques for intelligent urban traffic control. As an early example of this method can name intelligent transportation systems implemented Scoot [5] and Scats [6].They aim to optimize the signal cycle time.

Using intelligent technique in traffic control is divided in several ways that we briefly describe them as follow

Reinforcement Learning and Learning Q: In learning Q [7,8], agent learns relation with other agents and its environment while interacting with the environment and does not need a specific model of the environment.

Neural network: In methods based on neural network [7], system learns by using a set of learning. In papers [9] and [10] this method is used for intelligent control traffic.

Fuzzy logic systems: In fuzzy-based methods [7], the traffic situation is evaluated based on a set of rules. In the study [11], fuzzy logic is used to control traffic lights based on multi-agent systems to deal with problems such as accidents.

Genetic Algorithm: Genetic algorithm [7], is a kind of evolutionary algorithms based on biology techniques such as inheritance and mutation. In the study [12], genetic algorithm is used for optimum timing of traffic lights.

Multi-agent systems: The other intelligent method, is multi-agent systems [7], which is composed of several intelligent agents. These agents interact together. According to decentralized nature of disruption data in urban traffic during the urban transport network, multi-agent systems are suitable method and pattern for urban traffic control [3].

Two key issue in the design of multi-agent systems are presented: the first is design of agents and the second is design of environment for the function and communicate between agents. In designing an agent, how to build an autonomous agent that acts independently is considered. In designing the agent environment, the key point is how to design agents that are able to communicate with each other. The purpose of this communication is the cooperation, coordination and dialogue between agents. This is essential for successful performance on the tasks that they have set is necessary because not all agents have common goals or cannot be made with the same interests.

In research [13], [14] and [15], multi-agent systems is proposed to traffic control in cities for traffic lights control. In these researches is trying to use the exchanged messages between agents, traffic lights are set that are able to control the traffic.

3. Traffic load calculation methods

To enable the system to do the control operations in real time, the estimation of traffic load must be available at any moment to perform operation timing. Some traffic load estimation methods include: RFID [16], image processing and camera techniques [16] [17], wireless sensor networks [17], networks of vehicles [17], mobile signal [17], global positioning system (GPS). The use of RFID is not reliable in atmospheric conditions and during the extensive road network will be costly. Using the camera, sensor networks and wireless networks are very expensive. The processing of Global Positioning System has delay. Hence, in this article, we have used the mobile signal method to estimate the traffic load.

Using mobile signals, is a huge success in the management of road traffic because it is always available. One of the major drawbacks in the use of mobile phone signals that are active when they can be used. Therefore, there is a significant amount of error in predicting the density of vehicles. But today's at least one or two of passenger of vehicle has mobile, this error is much reduced. In metropolitan areas that the distance between the antennas is shorter, the data are more accurate. This method does not require any additional infrastructure except the mobile network, and this is one of its advantages. Hence, this method can be used with acceptable error percentage and this method is very cost effective [16].

4. Our proposed method for prioritizing bus

In this section, we present our proposal that is a multi-agent system to give priority to buses.

4.1 Determining agents

In this article, we simulated our proposed system using five agents are designed as follows:

- **Sensor agent:** in our proposed method, mobile signals are used to calculate intersections traffic load. Sensor agent estimates traffic load by collecting and analyzing these signals. Because, today more than one passenger of any vehicle has mobile phone, signal counting, do not give us the right statistics of traffic load. So, after collecting signals, TTSAS sequential algorithm [18] can be applied to the data. In this algorithm, two thresholds, T1 and T2, are set. The first signals form a cluster. With the other signals, if its distance from each of the clusters is less than T1, the signal is placed in the cluster and if greater than T2, the new cluster is formed and if its distance from each of the clusters is between these two thresholds, is determine in the next step because it may be placed in a cluster after other members. At the end, number of these clusters is equal to the number of vehicles or the traffic load. Every intersection has a sensor agent.

- **Intersection agent:** in the proposed system, every intersection has one intersection agent each cycle has a maximum value that can be set by the user. This agent is associated with the sensor agent, and gets information of its intersection traffic load at any time from sensor agent. Intersection agent is responsible for determining the best timing for traffic lights by using genetic algorithm and gives the suitable answer to all bus agents request as prevents stopping delayed buses at a red light and also be able to control the traffic.

- **Route agent**: Any direct street that connects the two intersections has a route agent. Buses are logged to the system with a fixed frequency of F and the bus route agent must try this frequency to be maintained between buses. The route agent has two tables: Station table and Time table. In station table, distance between stations and to intersection is inserted. Table 1 shows an example of a station table for two stations.

In this table, just distance from next stations is recorded. So the distance from previous station would be zero, because bus move forward and does not comeback.
The second table is a time table that shows the times that bus must pass the station. In most papers in the field of traffic control (focused on public transport), it is assumed that this timetable is manually given by user to the system. But keeping such a table needs space, in addition, the searching is difficult in such a table, as well as, working with the table is awkward. Hence, in this article, the route agent gets start time, end time and frequency of buses from the user and forms the time table. The specified frequency must be observed in columns and rows. Let the frequency of entering buses to the system to be 5 minutes, starting time 7:00 am, the bus number 2 and the number of stations is two. The route agent forms Table 2.
When each of the buses arrive to the station where just before the intersection, two parameters (including the id of bus and the time that must be on the station) will send to the next route agent and update the information of that bus in the time table with information of new bus.
The other task of the route agent is to set the priority for buses. Also, if there is a traffic in the bus route report it to supervisor agent.

- **Bus agent:** Each bus agent represents a bus in the real world. Each bus has a GPS (to find out the geographic location), a unique identifier (to communicate with) and a weight sensor (to estimate the number of passengers).

Table 1: An example of the stations table (distances is in meters)

*	Station 1	Station 2	Intersection
Station 1	0	7	17
Station 2	0	0	10

Table 2: An example of a timetable for 2 buses and 2 stations

*	Station 1	Station 2
BUS 1	7:00	7:05
BUS 2	7:05	7:10

To estimate the number of passengers, before the project, the average weight of people who use the bus can be obtained by the census. Then, the weight that calculates by weight sensor is divided to average weight and estimate number of passenger (see Eq. 1). Although this is not an exact number and is only an estimate, the computational load is very low and it can be calculated quickly.

$$N_{idB} = [W_s / W_{avg}] \quad (1)$$

N_{idB} is the number of passengers on the bus with id *idB*, W_s is sensor weight and W_{avg} is the average weight. Since N_{idB} is a positive integer, the result of division of weights is rounded to the nearest real number.

- **Supervisor agent:** Supervisor agent checks the position of buses at any moment using their GPS. If a bus is late and the traffic on the route is not reported, supervisor agent penalizes the bus. This makes the bus agent dose not refuse from his duty.

4.2 Our proposed architecture

Here, we describe our proposed architecture. First, we explain how agents work together, followed by a diagram of message passing of agents in our system. Then we explain the details of genetic algorithm used in our proposed method.

When a bus is placed in the route of the bus, gets the station table from route agent. When the bus agent reaches to every bus stations, a message including its name and the bus station ID sends to the route agent. Rout agent calculates the time that the bus must wait at the station to respect to the order [17] by follow:

$$W_T = T_{T\ V} - T_C \quad (2)$$

Where T_C is the arrival time of bus to station and $T_{T\text{-}V}$ is the bus arrival time that should arrive at the station according to the timetable of bus.

If W_T is greater than zero, the bus should wait in the bus station until the current time is equal the time that the bus should reach the station, but if it is less than or equal to zero, bus needs to leave quickly.

When the bus arrives to the station before the intersection, estimates the number of passengers by using weight sensor and then sends number of passenger, its id and station's id to the route agent. Route agent measures the criteria for the punctuality of buses [19]:

$$V_P = T_{T\ V} - (T_C - |W_T|) \quad (3)$$

If this measure is equal to zero, the bus is on schedule and does not need to prioritize. But if smaller than zero means that the bus is late at its timing. So it should be given priority to that bus at intersection to avoid waiting at a red light.

Then route agent measures bus priority, by using punctuality criteria of bus and the number of passengers (see Eq. 4), and sends priority and time waiting for the bus.

$$U_{BusId} = |V_P|(W_1 |V_P| + W_2 N_{IdB}) \quad (4)$$

Where N_{idB} is the number of passengers on the bus and V_p is the measure accuracy (means how much the bus was not on time by its timetable) or delay of the bus. Since Vp of the late bus is negative, we have the absolute value for it. Since if bus is not late, Vp becomes zero. So Multiply | V_P | in $(W_1 \mid V_P \mid + W_2 N_{idB})$ ensures that If the bus has not any delay, priority is not given to it. It also makes punctuality criteria of buses to be considered more than the number of passengers. Two variables W_1 and W_2, are the weight of each priority criteria. If the order is more important to us, consider W_1 to a large number and if passenger satisfaction was more important for us, consider W_2 to a large number. Here we consider the weight equal one.

The route agent gives the calculated priority to the bus. Then, bus agent estimates the arrival time to intersection according to the distance to the intersection. Thus, the bus agent calculates the amount of required green time, and sends along with its ID, its route ID and priority which it is calculated by route agent to the intersection agent.

Time requested by a bus by a start time (T_{begin}) and an end time (T_{end}) is determined. The beginning time specifies the moment when the bus arrives to the traffic lights. Requested green start time includes bus arrival times to traffic lights and the time needed to evacuate the street. Therefore, start and end times can be calculated [20]:

$$T_{begin} = T_0 + (L/V - N/D) \quad (5)$$

$$T_{end} = T_{begin} + (N/D) + T_{evabus} \quad (6)$$

Where L is the distance from the station to the intersection, V is speed bus to km / h, $T0$ is current time, N is number of vehicles on the streets, D is Street saturation capacity in terms of number of vehicles that can exit that street in each seconds and $T_{evacbus}$ is requested time to catch the bus from one street to another. In the article [18], $T_{evacbus}$ is considered equal to 2 seconds, so do we in this paper.

The intersection agent stored the reservation requests in its database. The intersection agent gives time to buses that have priority. If two requests have overlapping, the intersection agent first will respond to the request with higher priority. To do so, the intersection agent act according to the information received [20]:

$$U_{Roadi} = e^{Wi} + e^{bi} \quad (7)$$

Where $W_i = N_i / C_i$ is criterion of the degree of congestion in the streets, C_i is the capacity of street i that means the number of vehicles that can be placed in that street, bi is current number of buses and e is Euler constant that is greater than 1.

In the first step, the intersection agent determines the proper timing for traffic lights using the genetic algorithm. Each traffic light has a minimum value of green light. When intersection agent receives a request from the bus agent, if needed time is more than cycle value, all external time is given to that traffic light and the other traffic light values with minimum value of traffic light. Otherwise, the requested time is assigned to the relevant street and for three other lights, genetic algorithm is run.

In emergency cases, if the traffic of an intersection street is more than a threshold, the intersection agent informs critical situation and needed time to discharge the street to the respective route agent and then route agent wait its buses in the station a little more and until intersection problem is solved. So route agent calculates how many buses stay on the station:

$$Wait_{Busi} = \frac{T_{interrs} - T_{Busi}}{N_{stop}} \quad (9)$$

Where T_{inters} is needed time to evacuate, T_{Busi} is needed time to reach the bus i to the intersection and N_{stop} is the remaining number of bus to reach the bus i to the intersection.

A supervisor agent is described in this proposed method that checks location of buses by their GPS to ensure that no bus does not neglect their duty as well as informs the number of buses in the intersection to the intersection agent. Figure 1 shows how the exchange of messages between the agents of system.

Here, we explain in detail the genetic algorithm used in this study. The first step in using genetic algorithm is to define the chromosome. When we run genetic algorithm for four streets, our chromosome has four genes and when we run it for three street, our chromosome has three genes that each gen determines the optimized time for the green light of the relevant street.

There is a minimum value for traffic lights that can be set by the user. This limit helps that the amount of green light of any street is not be zero. For example, let a maximum of cycle is 20 and minimum time required for any street lights is 2. So the minimum time would be equal to 2* the number of traffic light that intersection needs for light. In this case, this value is 8. Thus the extra time to divide between lights is 20-8=12. So, the value of genes is between 0 to 12. Consider that sum of value of genes of a chromosome must be less or equal to maximum cycle.

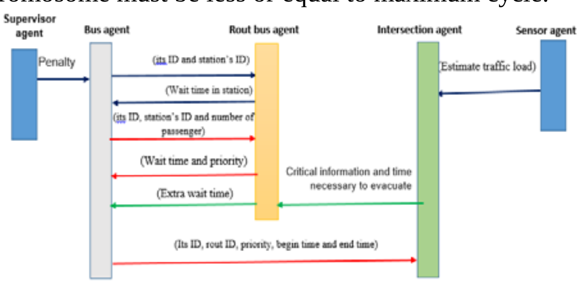

Fig. 1 Diagram of how the exchange of messages between agents of systems

Figure 2 shows an example of the proposed chromosomes.

7	2	3	0

Fig. 2 An example of the proposed chromosomes

So by observing these limitations, we developed a set of chromosomes. We use multi objective problem for simulation our system. We aim to reduce the traffic of streets and also determine minimum time for traffic light to be green. If we consider the minimum time, traffic congestion become more, however, if we want to reduce traffic congestion, we need more time. Therefore, this is a multi-objective problem.

The advantage of our proposed genetic algorithm is that the number of needed run algorithm for timing traffic light is about 30% less than normal genetic algorithm. Because, our fitness determines the less time to solve traffic, so it is possible that the sum of genes of chromosome is less than the maximum value of cycle. This difference is recorded in a variable. When a bus requests time from intersection, intersection agent first checks this variable. If this value is less or equal to the needed time, genetic algorithm is not run and this value is used. The fitness function for each chromosome is defined as:

$$F = \frac{1}{3}\left(\left(\frac{2}{3}\right)\left(\sum_{i=1}^{n} \mathrm{TRAFFIC}_i\right) + \left(\frac{1}{3}\right)\left(\sum_{i=1}^{n} \mathrm{TIME}_i\right)\right) \tag{10}$$

Where n is the number of gens, *TRAFFICi* is traffic of the street i (number of vehicles in street i) and *TIMEi* is the determination time for street i.

We consider two weights for the fitness that determines importance of traffic and time. Our main goal is the reduction of traffic, so weight of traffic (2/3) consider more than the weight of time (1/3). Since the range of time and traffic is not the same, we bring them to a same range:

$$TRAFFIC = \frac{traffic - traffic^{+}}{traffic^{-} - traffic^{+}} \tag{11}$$

Where *traffic* is remind traffic after run green light, *traffic+* is the best traffic that is equal zero and *traffic-* is the worst traffic that happened when no extra time dose not associate to it.

$$TIME = \frac{time - time^{+}}{time^{-} - time^{+}} \tag{12}$$

Where *time* is the value of gen, *time+* is the best time that is zero because one of our goal is to minimal green time and *time-* is the worst time that is equal extra time. Then we multiple time to reverse utility:

$$TIME = TIME * \frac{1}{UTILITy} \tag{13}$$

After defining the fitness function, a subset of the initial population is selected by using the wheel. Then, the rest of the steps of genetic such as crossover and mutations on chromosome are applied. Finally, after producing several successive generations, the best answer means the optimal duration for traffic lights intersection is obtained.

5. Implementation

Here we describe the implementation of our proposed method. We implemented our proposed method by system with core i3 processor and 2 Gig RAM and version 4.4 of JADE library. We tested our proposed method for 8 intersections and 8 buses.

Then, we calculated amount of bus delays in four methods (fixed time, without priority, with priority and the proposed method). Also, we calculated the amount of reduction of traffic loads in these four methods. Figure 3, shows eight intersections with their traffic loads. Traffic lights of intersections were calculated by using genetic algorithm.

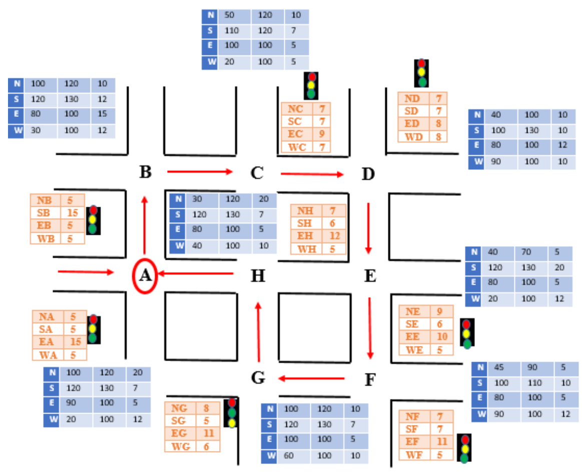

Fig. 3 An example of 8 intersections and their traffic loads

The red arrows show the routes that bus should go. The bus starts in intersection A, and finally when the bus arrives to intersection A, its work is finished.

As we mentioned before, when bus arrives to the station, the bus sends a message to route agent. To detection station, there are several ways:

1) Using camera in station that is very expensive

2) Using GPS of buses that one needs always check the location of buses

3) When arrives to a station, driver push a key (that is boring and maybe driver forgets)

4) Using kilometer counter of bus.

We proposed to use kilometer counter of bus. Because, all buses have kilometer counter, and this method does not require any additional infrastructure, and even when bus stops, it works correctly.

As we did not access kilometer counter information, to simulation this system we use speed formula as follow [22]:

$$V^2-V_0^2 = 2a\Delta x \; (a = (V^2-V_0^2)/2\Delta x \qquad (14)$$

Where V is the speed of the bus, V0 is the initial speed of the bus and because the bus is stopped in station and moved again, its initial speed is zero, a is average acceleration and Δx is displacement where here is the distance between the two stations.

As we use speed formula, so we consider that there is no traffic in the route bus and bus stops only at station. Using the kilometer counter does not have this restriction.

6. Evaluation of our proposed method

We compare our proposed system with fixed time, without priority and with priority that use only multi-agent system. Results are shown in Table 3 and 4.

Table 3 shows the average delay of buses for 10 runs. We ran the application for 10 times. Then, the time required for late bus to pass through the intersection is subtracted from when the time of the traffic lights. The difference between these two is the amount of time waiting for the late bus. The average delay of buses in fixed time method is 45. It means that the scheduling system of the traffic lights by fixed time, the late buses linger on red lights for 45 seconds. But in with priority and proposed method this wait time for late buses is 7.

Table 3: Average delay of buses

System type	Average delay of buses
Fixed time	45
Without priority	40
With priority	7
Our Proposed method	7

Table 4 shows the average reduction of traffic in the four methods. We calculate the number of vehicles that remind at red lights (we subtracted the number of vehicle before the green light from number of vehicle that cross the intersection during green light). To calculate the number of vehicle that cross the intersection during green light, as we said every street has a saturation capacity value in terms of number of vehicles that can exit that street in each seconds. So, we multiplied this value to the value of traffic light of that street.

Table. 4: The reduction in overall traffic

System	The reduction in overall traffic
Fixed time	41.32%
Without priority	52.27%
With priority	50.73%
Our Proposed method	51.75%

Results showed that our system has less delay than the fixed time method and the method without priority and is equal with priority that uses only multi-agent system. Traffic reduction of our system is more than fixed time and with priority. As results showed, traffic reduction of our system is less than without priority method, but when delay of buses become less, people use buses more and traffic becomes less.

7. Conclusion

In many cities, the bus is the most important and common form of public transport. Giving priority to buses plays an important role in protecting bus services against the effects of traffic congestion, improving the frequency, speed and reliability [2]. Thus, in this paper, we propose a method based on multi-agent systems to give priority to buses.

The advantage of the proposed method include: the use of mobile phone signals with an acceptable percentage of errors (without having to build any hardware) is very affordable. In addition, in our method, the best timing for traffic lights is calculated using genetic algorithm. Considering the number of vehicles at intersections to calculate the priority of streets to set traffic lights by genetic algorithm, also improves the overall traffic situation. Considering number of bus passengers to give priority, makes that passenger satisfaction is also considered in this category, which has subsequently important role in reducing traffic.

Unlike the methods proposed so far, the proposed method is the combination of two intelligent method, multi-agent systems and genetic algorithms, and this makes it more accurate than the previous method. Also, none of the methods proposed so far, did not consider number of bus passengers, while this factor has an important role in satisfaction and encourage passengers to use public transport and it impacts on the traffic.

In this study, we implemented our multi-agent system using Jade, and we represent the results of our implementation in small scale, followed by the evaluation of our proposed method with other works.

In our future work, we are going to go a step further in this proposal. We will test our method on a large scale with real data, and also we will consider the number of passengers who wait in the bus station. Also, we can consider the time of day to increase the number of buses during rush hour.

References

[1] S. Darmoul and S. Elkosantini, "Artificial immunity to control disturbances in public transportation systems: Concepts, mechanisms and a prototype implementation of a knowledge based decision support system," Knowledge-Based Systems , vol. 68, p. 58–76, 2014.

[2] B. Ahmed, "Exploring new bus priority methods at isolated vehicle actuated junctions," Transportation Research Procedia, vol. 4, p. 391 – 406, 2014.

[3] M.A. Purnasir Kennedy, A. Pirkhah Aqbash, F. Derakhshan, “Urban Traffic Control and bus priority based on multi-agent systems“, Seventh International Conference on Information and Knowledge Technology, University of Uremia, 2015.

[4] N. Bhouri, F. . J. Mayorano , P. A. Lotito, H. H. Salem and J. . P. Lebacque, "Public Transport Priority for Multimodal Urban Traffic Control," CYBERNETICS AND INFORMATION TECHNOLOGIES, vol. 15, 2015.

[5] J. Luk, "Two traffic-responsive area traffic control methods: SCATS and SCOOT," Traffic Engineering and Control, pp. 14-20, 1984.

[6] P. R. Lowrie, SCATS, SYDNEY CO-ORDINATED ADAPTIVE TRAFFIC SYSTEM: A TRAFFIC RESPONSIVE METHOD OF CONTROLLING URBAN TRAFFIC, Sydney, Australia: Roads & Traffic Authority, 1990.

[7] S. Araghi , A. Khosravi and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," Expert Systems with Applications , vol. 42 , p. 1538–1550, 2015.

[8] J. Glick and M. Candidate, Management Science & Engineering, 2015.

[9] T. Royani, J. Haddadnia and M. Alipoor, "Control of Traffic Light in Isolated Intersections Using Fuzzy Neural Network and Genetic Algorithm," International Journal of Computer and Electrical Engineering, vol. 5, 2013.

[10] M. Zhang and Y. Lu , "Adaptive Network Traffic Prediction Algorithm based on BP Neural Network," International Journal of Future Generation Communication and Networking, pp. 195-206, 2015.

[11] M. Salehi, I. Sepahvand and M. Yarahmadi, "TLCSBFL: A Traffic Lights Control System Based on Fuzzy Logic," International Journal of u- and e- Service, Science and Technology, vol. 7, pp. 27-34, 2014.

[12] M. Suhail, "Management of an intelligent traffic light system by using genetic algorithm TRAFFIC DETECTION," International Journal of Research in Engineering and Technology, 2014.

[13] S. . S. Sandhu, N. Jain , . A. Gaurav, N. C. Sriman and N. Iyengar, "Agent Based Intelligent Traffic Management System for Smart Cities," International Journal of Smart Home, vol. 9, pp. 307-316, 2015.

[14] M. Abdoos, N. Mozayani and . A. Bazzan, "Holonic multi-agent system for traffic signals control," Engineering Applications of Artificial Intelligence, p. 1575–1587, 2013.

[15] F. Zhu, H. Abdul Aziz , . X. Qian and . V. Ukkusuri, ".A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework," Transportation Research Part C, vol. 58, p. 487–501, 2015.

[16] J. Mathew and P. M. Xavier, "A SURVEY ON USING WIRELESS SIGNALS FOR ROAD TRAFFIC DETECTION," International Journal of Research in Engineering and Technology, vol. 3, 2014.

[17] S. Kulkarni, V. Dass and A. Sharma, "Intelligent Traffic Light Control with Wireless Sensor Networks," International Journal of Innovative Research in Computer Science & Technology (IJIRCST), vol. 2, 2014.

[18] S. Theodoridis and K. Koutroumbas, "Clustering Algorithm I," in PATTERN RECOGNITION, Third Edition ed., 2006, pp. 517-540.

[19] N. Morri, S. Hadouaj and L. B. Sa, "Multi-agent Modelling for a Regulation Support System Of Public Transport," in 17th International Conference on Enterprise Information Systems, Barcelona, 2015.

[20] M. Tlig and N. Bhouri, "A Multi-Agent System for Urban Traffic and Buses Regularity," Procedia Social and Behavioral Sciences 20, p. 896–905, 2011.
[21] "Second year high school physics formulas," [Online]. Available: http://physics-department.ir/wp-content/uploads/%D9%81%D8%B1%D9%85%D9%88%D9%84-%D9%87%D8%A7%DB%8C-%D9%81%DB%8C%D8%B2%DB%8C%DA%A9-%D8%B3%D8%A7%D9%84-%D8%AF%D9%88%D9%85.pdf. [Accessed 11 07 **2016].**

FarnazDerakhshan received her Ph.D. degree in Artificial Intelligence from Liverpool University, UK in 2008. She is assistant professor of the Faculty of Electrical & Computer Engineering, University of Tabriz, Iran since 2009 who is interested in multi-agent systems field.

Nasrin Shahpasandi currently is a graduate student in Artificial Intelligence in the Faculty of Electrical & Computer Engineering, University of Tabriz, Iran who is interested in intelligent traffic control subject.
Systems, and wireless access network using stratospheric platforms. He now with DDI Tokyo Pocket Telephone, Inc.