CITL: A Multi-Agent System to Control the Intersection Traffic Lights

Fatemeh Khezerlou† and Farnaz Derakhshan††

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

Summary
In this paper, we used an intelligent method to control traffic lights at intersections based on multi-agent system called CITL. Our proposed method uses clustering and evolutionary computing algorithms to schedule the traffic lights, in addition to use exchanging of messages between different agents to control possible fluctuations. The main capability of our proposal is that our method performed distributed processing centers which avoid excessive computational load processing in each cycle and it reduces delay time compare with other methods.
For the design and implementation of our method, we used Tropos methodology and JADE library.
For evaluation, this method tested with 2000 laboratory data in both low and heavy traffic areas.
The average run-time in the sequential cycles in our combinatorial method is less than using the other single methods compared to the fixed-time methods, using our method leads to the improvement of the delay time of vehicles 24.5% and 35.5% in heavy-traffic and low-traffic areas, respectively. Also, CITL get improved the delay time 9.58% and 9.79% in heavy-traffic and low-traffic areas, compared to the similarmulti-agent systems.

Key words:
Intelligent traffic control; Multi-agent systems; Urban traffic control.

1. Introduction

If urban transportation infrastructures have not been developed along with increasing population of cities, lots of problems raised. One of the most applicable approaches to solve traffic problems would be using intelligent traffic control and management methods are.
Traffic control methods are mainly conducted by one of these two ways [3]: first, fixed-time methods. Second, intelligent methods. The fixed-time methods [4] acts offline. In this method, information of intersections would be recorded in several periods, and then according to obtained data for traffic control, scheduling would be done. In fact, the system acts based on its earlier information; therefore, it is unable to show the response to new events and also cannot manage unexpected conditions in traffic. Hence, this method can act properly in the independent intersections.
Using artificial intelligence methods to control traffic signals began in 1990 that machine learning techniques applied for control urban traffic. Scoot [5, 6] and Scats [7, 8] are primary and successful examples of intelligent transportation systems so far implemented. Machine learning techniques for using in intelligent traffic control methods divided into several categories including [9]: Reinforcement learning and Q-learning, (used in [10, 11]). Neural network, (used in studies [12, 13]). Fuzzy logic systems (used in [14, 15]) Genetic algorithm (used in [16, 17, 18, 19]) and Multi-agent systems (used in [20, 21, 22,23, 24]).
We have used both multi-agent systems and genetic algorithm to design and implementation of CITL. Urban traffic distributed along the urban transportation network and also it has high oscillations. Therefore, we could not have a complete view on whole network. In order to have a complete view, one solution would be to install lots of sensors across the network, then all sensors send information to a center, which would be a high-cost method. One of the most applicable intelligent methods for traffic control is using multi-agent system (MAS). According to distributive nature of urban traffic and transportation, multi-agent systems are efficient patterns to control urban traffic. A multi-agent system (MAS) uses various intelligent agents. The agents are interacting with each other and this method can be applied on complicated systems. Here, we mention a number of intelligent traffic systems based on multi-agent systems.
In our previous paper [25], we present a multi-agent based system for urban traffic control too. In our proposed method, after estimating the traffic load of intersections, we used K-means and genetic algorithms to schedule traffic lights. This method designed by five agents: traffic sensor agent, intersection agent, clustering agent, calculating agent and coordinator agent. After that, in our next paper [26], in order to improve our method, we change the way of calculate traffic load and process of genetic algorithm. In our earlier paper [27], we designed and implemented our proposed method with Tropos methodology and JADE library. This proposed system used four agents and has the ability of exchanging messages between agents. This proposed system tested with 1200 laboratory data for both low and heavy traffic areas. Our system improved the delay time of vehicles, compared to the fixed-time methods.

Manuscript received March 5, 2017
Manuscript revised March 20, 2017

Now, in this paper, we used an intelligent method to control traffic lights at intersections based on multi-agent system called CITL. In fact, we continue and improve our previous work. CITL is based on our previous work which uses clustering and evolutionary computing algorithms to schedule the traffic lights, Furthermore we used the ability of exchanging messages between different agents. Moreover, we designed new architecture to reduce delay time, compared with similar multi-agent systems and fixed-time methods. After this introduction, the paper is organized as follows: In Section 2, CITL is described in more detail, and the architecture of our proposed method is explained. In Section 3, we describe the implementation and running results of CITL at laboratory level for a limited zone of city and compared the results with other implemented methods. Finally, in Section 4, we make conclusion about CITL and explain our future works.

2. Our Proposed Method

Using intelligent methods is one of the efficient methods of scheduling the traffic lights at intersections to control traffic urban. In this paper, we used the combinatorial method to control urban traffic at intersections based on multi-agent system. Our proposed method has four main steps:

First, proposed system is calculating the traffic load of intersections which there exist several ways to calculate, such as [31]: induction loop tracking, vehicle video tracking, computing technologies, obtaining the data of moving cars using mobile phones and RFID. However, here we suppose that we already had this traffic load of intersections.

Second, our system uses clustering algorithm to find adjacent intersections which are related to each other. To do so, among several clustering algorithms, we use proposal two-stage K-means algorithm, considering the distance and congestion features of intersections. We chose K-means algorithm because of the simplicity in calculations.

Third, our proposed method uses evolutionary computing algorithm to scheduling the traffic lights. For this purpose, we used the genetic algorithm for scheduling. Genetic algorithm does not get stuck onto local optimum and will find the global optimal or close to it, if one chose proper exploration way.

Finally, our proposed method used the capability of exchange of messages between different agents to control possible fluctuations. To achieve this, we used Tropos methodology [29, 30, 31] to design our system and JADE [28] library to definition of agents and interaction of them.

Some advantages of CITL such as follows: The number of intersections is considered infinite. Also, the distance between intersections is not limited and adjustable. Furthermore, this method is applicable to multi-ways. Due to use multi-agent technique and clustering algorithm for intersections, proposed method has been performed distributed processing which avoid excessive computational load processing in each cycle of traffic lights. In this paper, we focused on appropriate scheduling of the traffic lights with new architecture of exchanging message between agents. According to the above-mentioned phases, in the next section, we explain our intelligent multi-agent based proposal for urban traffic control systems. In the following, our method will be described in more detail.

2.1 Defining Agents

First, to control the traffic at the intersection by using multi-agent systems is defining agents and their interaction. At first, we introduce the agents in our proposed method, and then we explain each agent tasks according to the study. In this system, defined agents are as follows:

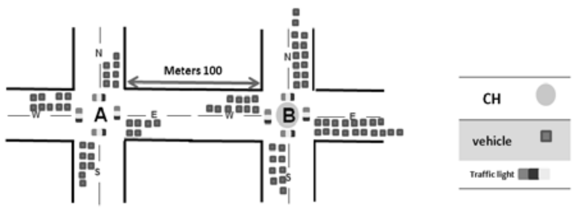
Traffic sensor agent: This agent calculates the load of all roads of intersection and sends this data to its related intersection agent.

Intersection agent: Each intersection agent has a traffic sensor agent. Intersection agent knows its distance from the other intersections which directly linked to it. This agent receives the traffic load information in each cycle of traffic-lights. Moreover, each intersection agent saves final results of scheduling of the calculator agent in each cycle.

In summary, each intersection agent has the following information: 1. Intersection name. 2. Current traffic load. 3. Best current schedule. 4. Distance from neighbour intersections. 5. Connections of any road of intersection with roads of other intersections. Fig. 1 shows the intersection.

Clustering agent: To find adjacent intersections we used clustering agent. Assuming the studied city is zoned, a clustering agent is assigned for each zone. Each clustering agent is in relationship with intersection agents and uses their information. Here, we use K-means clustering algorithm [32]. One of the main problems of K-means clustering algorithm [33] is the method of choosing cluster heads, in this study to solve this problem, each clustering agent considers a threshold for traffic load and finds a set of intersections among intersections of its zone as its cluster head (CH). Then, clustering agent runs K-means algorithm in two main steps, considering distance and traffic load of the intersections. There are three stages for the first step of clustering:

a) The first stage, CHs is determined by considered threshold. Applying this stage can distribute processing centers in the region and also it can solve the problem of K-means algorithm. In Fig. 2, suppose intersection A and intersection E have the traffic loads more than the determined threshold, so these intersections are appointed as CHs



.

Fig. 1: Each intersection information [27].

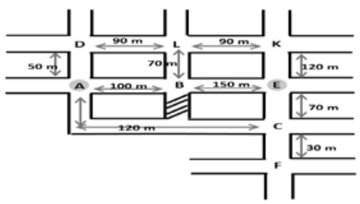


Fig. 2: arrangement the intersections of a specified area of the city with cluster heads [27].

b) The second stage, considering the shortest distances, the intersections which are connected directly to CHs, are set into the proportionate cluster as shown in Fig. 3.

c) The third stage, the intersections which are connected indirectly to CHs, are allocated to clusters by considering the shortest distance to any member of the clusters as shown in Fig. 4.

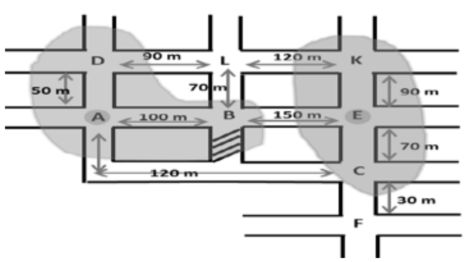


Fig. 3: The second stage of determine members of each cluster [27].

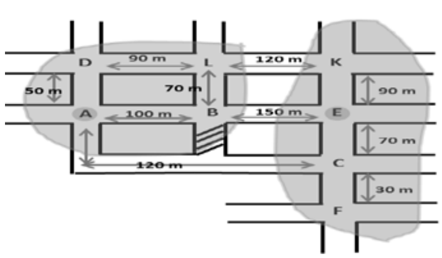


Fig. 4: The third stage of determination of members of each cluster [27].

The final result of first step of clustering is shown in Fig. 3, after determination the CHs, k-means algorithm implemented only based on the distance between the intersections.

In the second step of clustering, predetermined clusters will be examined based on traffic load. We apply this step to balance traffic load between clusters to avoid congestion.

The second step of clustering process in two stages. They are as follows:

a) The first stage, every cluster is labelled high-density or low-density. Traffic load of each intersection plus 20% of its traffic load is calculated. If the total traffic load is more than the determined threshold, then the algorithm counts these intersections. If the number of these intersections in each cluster is more than the total number of intersections in each cluster, the cluster will be labelled as high-density, if the clusters have not these conditions, then the cluster will be labelled as low-density.

The second stage, in high-density cluster each intersection which has transportation conditions, can be removed and transfer to low-density cluster.

These transportation conditions are as follows:

- Neighborhood with low-density clusters.
- Transferring to the new cluster could be fragmentation previous cluster.

If intersection K and Intersection C have the traffic loads close to the threshold, then cluster E labelled as high-density cluster. We can transfer the intersection K into the cluster A but intersection C will remain in own cluster because of fragmentation in cluster E. Fig. 5 shows how clusters changed.

The output of second step is determining the final clustering. Next, calculator agent will compute the best scheduling for intersections of each cluster.

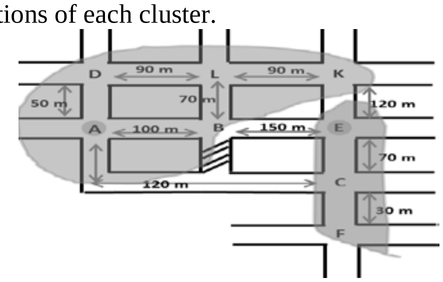


Fig. 5: The second stage of the second step of final determination of each cluster [27].

Calculator agent: Calculator agent is determining the best scheduling for the traffic lights of intersections. The calculator agent gets information sent by clustering agent, then calculates scheduling of traffic lights by using Genetic algorithm [31]. Finally, it sends the schedule to intersection agents to apply on their traffic lights.

Genetic algorithm is reliable, since it would not be trapped in local optimal and finds surely the global (or close to global) optimal Here, the global optimal is the same as the best possible schedule adjusted with traffic load of intersections.

Genetic Algorithm in Calculator Agent

Here, we explain the details of Genetic Algorithm for our method which is implemented. Although we explain this section in our previous work in Persian, here we recall in English.

The first stage in the implementation of the genetic algorithm is determining genes and the chromosomes. The minimum duration of the green light for traffic lights of the intersection is mostly 15 seconds and the maximum duration is 20 seconds, duration time per cycle of traffic lights is considered to be 70 seconds. The value of each gene will be between 0 and 5, in a way that the total time for the green light of each intersection is 70 seconds.

If cluster contains m intersections as four-way and n intersections as three-way the length of the chromosome will be calculated as follows:

Length of the chromosome: $\sum_{i=1}^{n} 3 + \sum_{i=1}^{m} 4 + \dots$

This makes no restrictions on scheduling for various types of intersections with different number of roads such as four-way or three-way. For example, as shown in Fig. 6, cluster A has 4 intersections as four-way and 1 intersection as three-way.

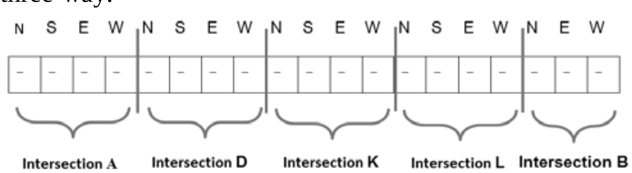


Fig. 6: The proposed chromosome for the cluster A [27].

A limitation is applied on genes of proposed chromosome. This limitation can control the sudden fluctuations of traffic load of intersection and it can show the good response. By using this limitation, there is no need to change the scheduling and repeat all of calculations per cycle. The considered confine for traffic load should not be too large or not be too small, since if it is too large, the system may face problem and be unable to solve traffic problem and the traffic would be increased gradually and if it is too small, it would have no effect and would be unable to improve system. Hence, 10% (tolerance) of traffic load is considered.

The proposed fitness function should be determined based on the following conditions (recall from [27]):1) Probable fluctuations of traffic load of each road up to 10 %. 2) The size of each chromosome is proportional to the size of the cluster. 3) Performance Index (G.T), (see Eq. 1).

While g is extra green light duration in the range of 0 to 5, S_i is the actual traffic load of road I.

$$G.T = 15\,s + g \tag{1}$$

Fitness function for road I of intersection, (see Eq. 2).

$$f_i = (S_i)/G.T_i \rightarrow P.I_i = (S_i + 0.1 \times S_i)/G.T_i \tag{2}$$

Fitness function for intersection A, (see Eq. 3).

$$f_A = f_1 + f_2 + f_3 + f_4 \tag{3}$$

Finally, fitness function is proportional to the chromosome for cluster A with 5 intersections will be calculated, (see Eq. 4). Objective is minimizing the F.

$$F = f_A + f_B + f_L + f_D + f_K \tag{4}$$

The initial population size is considered to be 50, and the number of generation is considered to be 10. Also in parental choice [31] process with regard to the features elitism the Roulette Wheel [31] method is used. One-point method [31] is used for crossover, next Uniform method [31] has been used for mutation, and finally Repair [31] operations for restrictions are considered. Eventually, the best chromosome is selected as the best schedule for each cluster of intersections.

2.2 How Messages Are Exchanged Between Agents

One of the most important features that JADE [28] agents provide is the ability to communicate. The communication paradigm adopted is the asynchronous message passing. Each agent has a sort of mailbox (the agent message queue) where the JADE runtime posts messages sent by other agents. Whenever a message is posted in the message queue the receiving agent is notified. If and when the agent actually picks up the message from the message queue to process it is completely up to the programmer however.

We consider 70 seconds for the duration of each cycle of traffic lights. After 70 seconds, the traffic load of intersection will be changed, thus, we will be faced with five situations:

1- In the first situation, traffic load of intersection will be faced with increase or decrease in the load of traffic. If the increase in traffic load of intersection is not more than 10%, the previous schedule applies (since we have considered possible fluctuation up to 10% for each road of intersection).

2- In the second situation, traffic load of intersection has changed and the increase in traffic load of road is more than 10%. Therefore, at the first step, intersection agent calculates the extra time. If it was able to recoup extra time, its time will be changed, it is probable that one of roads will be faced with increasing traffic load and the other one with decrease in the traffic load. This problem was solved by itself.

The method of calculating extra time:

$$S_i = S_i + 10\% * K \tag{5}$$

$$T_e = T - (T \times S_i / K) \tag{6}$$

While T is the optimal allocated time, Si is the actual traffic load of road; K is the amount of traffic load considered with tolerance of fluctuations in genetics and T_e is the amount of extra time for road I.

If the value of T_e was positive, it represents the amount of extra load and otherwise it represents the lack of time.

3- In third situations, the increase in traffic load of intersection is more than 10%. Therefore, at first, the intersection calculates the extra time. As it is not capable of recouping the extra time, it sends the CFP message to its CH and requested additional time. The CH will calculate its own extra time and other members of own cluster, If the CH can provide the requested additional time, the CH updates time of the members of the cluster and its own time and sends the proposed message to the applicant intersection. This problem was solved by CH.

4- In forth situation, the traffic load of the intersection has been changed and the increase in the traffic load of road is more than 10%. Thus, at first, the intersection calculates the extra time, but it is not capable of recouping the extra time. Therefore, it sends the CFP message to its CH and requested additional time. The CH will calculate its own extra time and the time of other members of its own cluster, if the CH cannot provide the requested additional time, it sends the CFP message to the adjacent CHs that are connected directly to the requested intersection. The CHs will calculate themselves extra time and the time of other members of themselves clusters, then if the adjacent CHs can provide the requested additional time, they send CFP message to related CH. Then, the CH updates time of the members of the clusters and sends the CFP message to the applicant intersection. This problem was solved by adjacent CHs.

5- In this situation, the traffic load of the intersection has been changed and the increase in the traffic load of road is more than 10%. Thus, at first, the intersection calculates the extra time, but it is not capable of recouping the extra time. Therefore, it sends the CFP message to its head cluster and requested additional time. The CH will calculate its own extra time and the time of other members of its own cluster, if the CH cannot provide the requested additional time, it sends the CFP message to the adjacent CHs. The adjacent CHs will calculate themselves extra time and the time of other members of themselves cluster, then if the adjacent CHs cannot provide the requested additional time they send Refuse message to related CH. Then, the CH sends Refuse message to applicant intersection and Inform message to clustering agent of region. So clustering agent, clusters intersections of own region and sends the CFP message to calculator agent and requests to calculate optimal trimming with new traffic load for the traffic lights of intersections. Calculator agent reply message to clustering agent by accepting proposal subject.
Fig. 7 shows how messages are exchanged between agents in a zone of the city in each cycle of traffic lights.

As you see, clustering and scheduling algorithms based on new traffic load are done just in one situation in each cycle. So CITL does not need to execute the program in every cycle of traffic lights. Then it can reduce delay time with a fine exchange of messages between different agents. In next section, we show modelling and implementation of our method, followed by running of the model.

2.3. Architecture of CITL

In this section, the architecture of our proposed method has been introduced. Fig. 8, shows architecture of CITL in a region of city.

2.4 Modeling and Implementation of CITL

In order to modelling our agent-based method, we used Tropos [26, 27, 28] which is an agent-oriented software engineering methodology that covers the whole software development process. Tropos is based on two key ideas. First, the notion of agent and all related mentalistic notions are used in all phases of software development, from early analysis down to the actual implementation. Second, Tropos covers also the very early phases of requirements analysis, thus allowing for a deeper understanding of the environment where the software must operate, and of the kind of interactions that should occur between software and human agents.
The proposed methodology spans four phases: 1) Early requirements 2) Late requirements 3) Architectural design 4) Detailed design. The output of late requirements is two diagrams: 1) Actor Diagram 2) Goal Diagram. Fig .9, shows Actor diagram of CITL by using Tropos methodology. In this diagram, the circle represents the agent, ellipses indicate goals, and cloud shapes indicate soft and dashed indicate the several sub-systems of this type that can be placed in the environment.

3. Discussion

We compare our system with other implemented methods for schedule and control traffic lights. Table. 1, shows the overall view of proposed method and other various implemented systems [35, 36]. In Table. 1, columns show the main features of control traffic lights systems and rows show the variety of systems. As can be seen, based on the mentioned features, the fixed-time system is known as the worst system and the local intelligent system, which is one of the newest implementation system in cities and has the

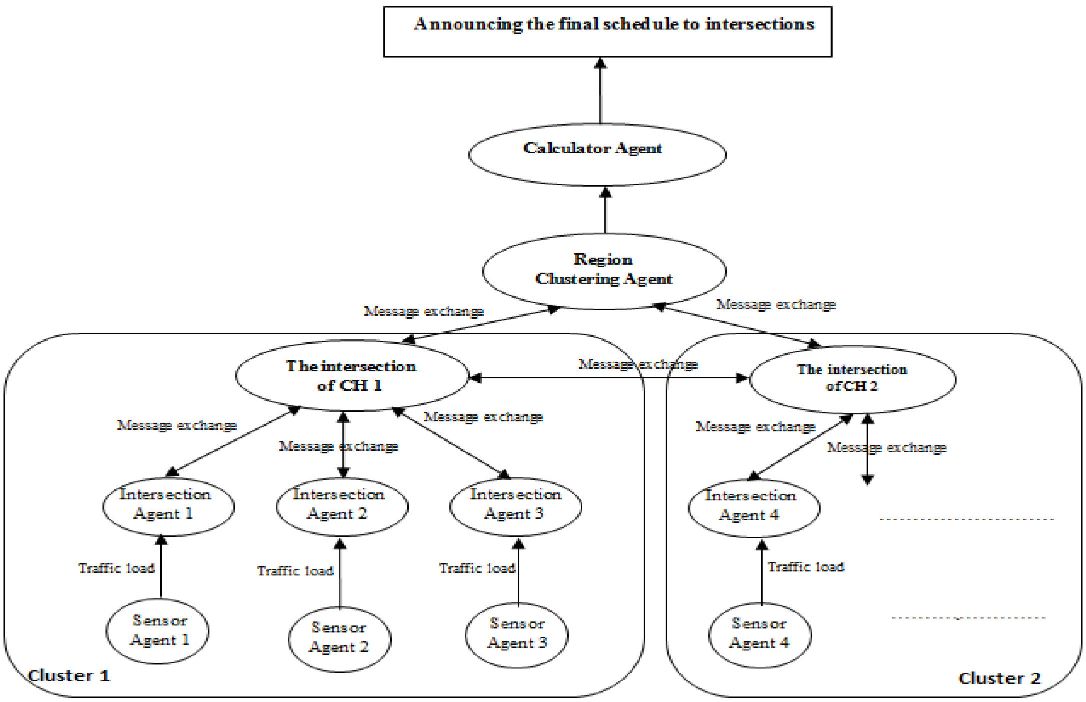


Fig. 8: Architecture of CITL in a region of city.

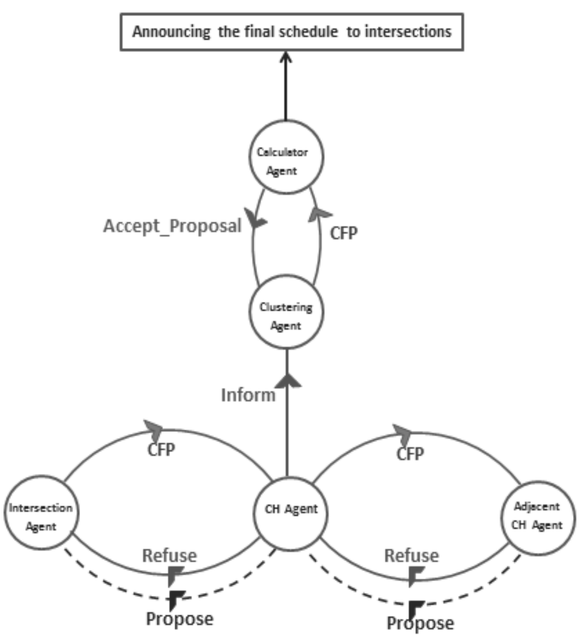


Fig. 7: show exchanged messages between agents in CITL.

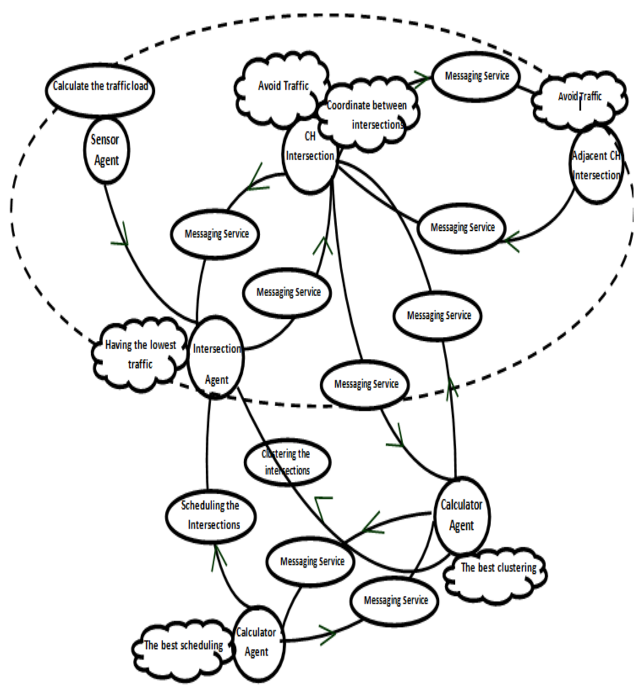


Fig. 9: Actor Diagram for CITL by using Tropos methodology.

local intelligent traffic control system, do not have the autonomous capability of agents and also, it is limited in the number of controlled intersections and computing centers. We tried to consider all existing capabilities in design and implementation of proposed system. The implementation results are listed in tables 2 and 3 in more detail.

We implement our proposed method by JADE [28] library with version 4.4.0. Then we run our code on a system with a Core i7 processor and 4 GB of main memory.

The total number of vehicles in low-traffic area: 500, The total number of vehicles in heavy-traffic area: 1500.

Then, we execute codes for 10 times. Tables 2 and 3 show results in terms of execution time. In tables 2 and 3, columns show one example of multi-agent systems and one example of systems which using the single intelligent methods to control traffic lights and rows show average duration time (milliseconds or seconds) of updating traffic lights by intersection, CH, adjacent CHs and finally delay time of vehicles in various system.

As can be seen in tables 2 and 3, in the first cycle, since the clustering and timing operations are implemented associated with it, the run time compared with other systems in the same region, is high, but the next cycles do not need to run all the calculations. As mentioned in the previous sections, we have used the self-autonomous agent and exchanged messages among agents. In addition, the potential fluctuations in traffic and the calculation centers are considered in the distributed form in the region. For this reason, the average time delay for vehicles on consecutive cycles significantly reduced in comparison with other similar systems [18, 27].

The most important difference of the CITL with the previously proposed method is that we could request extra time by changing the architecture and exchanging messages between agents in the event of a sudden increase in traffic, in addition to requesting the additional time from the cluster head and adjacent clusters (in case of direct connection to the troubled intersection). This reduces the lag time approximately 10% in the same system [27].

The processing stages in proposed system are carried out hierarchical. Initially, the clustering and timing operations are performed. Then, in the next cycle, by a sudden increase in traffic more than 10%, the additional time requested by the intersection is investigated. If it was unable to solve the problem, the current cluster head is assessed, and if not, the adjacent cluster head. Finally, if it was unable to solve the problem a new timing is applied corresponding to the new traffic load by the clustering and calculator agents. It is necessary to mention that we run our proposed method at laboratory level for limited zone of city, however, we did not test it by real data; therefore, we cannot have accurate assessment compared with two systems, namely variable timing system and local intelligent system. Thus, we compared our method with fixed time system and a similar multi-agent system in detail.

In CITL, we schedule traffic lights by considering traffic load and proximity intersections. Thus, the waiting time of vehicles at red lights is reduced.

In a more fixed time system used in cities, 20 seconds are allocated for all traffic lights. In our proposed method, duration time of traffic lights is between 15 and 20 seconds. We allocated time for each traffic light according to their traffic load.

We performed our proposed method 10 times in heavy-traffic and low-traffic areas and we studied the results on a specific intersection. In comparison with fixed-time systems for traffic light control, the average run-time in the sequential cycles of intersections in our combinational method is less than using the other single methods. In addition, we can reduce delay time from 20 seconds to 15.1 (in average) seconds in heavy-traffic areas and reduce delay time from 20 seconds to 12.9 (in average) seconds in low-traffic areas. Thus, in heavy-traffic and low-traffic areas, we had 24.5% and 35.5% (in average) improvement in, respectively, delay time of vehicles and the fixed-time methods. In addition, we can reduce delay time from 16.7 seconds to 15.1 (in average) seconds in heavy traffic heavy-traffic areas and reduce delay time from 14.3 seconds to 12.9 (in average) seconds in low-traffic areas. Thus, in heavy-traffic and low-traffic areas, we had 9.58% and 9.79% (in average) improvement, respectively, in delay time of vehicles, compared to similar [27] methods.

Here we list the result of our comparison as follows:

Our proposed method does not need to run all processing in the all of cycles. 2. Sudden fluctuations in each traffic light cycle can be controlled by exchanging messages. 3. By controlling range of proposed tolerance of fluctuations, repeated processes in many cycles can be prevented. 4. By consideration autonomous capability agents, all of agents (Intersections, CHs adjacent, CHs) can solve problem of sudden increase in traffic load problems hierarchically, so the number of complete runs and delay time will be reduced. 5. Our proposed method is performed distributed processing centers which avoid excessive computational load processing in each cycle. 6. Our proposed method has better performance in heavy-traffic area than low-traffic area, because tolerance (10%) considered involves more numbers of vehicles in sudden fluctuations, so we do not need to run all processing.

Table. 1: comparing CITL with related methods for implementation, scheduling of traffic lights

System	Considering load fluctuations	repeat the process in each cycle	Timing according to the traffic load of intersections	Distributed processing centers	The ability to self-autonomy	Lack of restrictions on the number of intersections
Fixed-Timed System	NO	-	NO	NO	NO	NO
Variable timing system	NO	Yes	Yes	NO	NO	NO
Local intelligent system	Yes	Yes	Yes	NO	NO	NO
CITL: Our new method	Yes	NO	Yes	Yes	Yes	Yes

Table. 2: Average run-time for 10 performances in low- traffic area.

Average duration of time updating traffic light	In [18] (Just using genetic algorithm)	In [27] (MAS based system)	CITL: Our new method
Run complete of program (by considering exchange message)	101.3 ms	142.7 ms	136.4 ms
Time updating traffic light by own intersection	-	2.9 ms	3.1 ms
Time updating traffic light by own CH.	-	18.3 ms	19.1 ms
Time updating traffic light by adjacent CHs	-	-	21.1 ms
Delay time for special intersection	20 s	14.3 s	12.9 s

Table. 3: Average run-time for 10 performances in heavy- traffic area.

Average duration of time updating traffic light	In [18] (Just using genetic algorithm)	In [27] (MAS based system)	CITL: Our new method
Run complete of program (by considering message)	114.2 ms	166.5 ms	151.8 ms
Time updating traffic light by own intersection	-	2.4 ms	2.7 ms
Time updating traffic light by CH.	-	14.6 ms	15.3 ms
Time updating traffic light by adjacent CHs	-	-	17.6 ms
Delay time for special intersection	20 s	16.7 s	15.1 s

4. Conclusions

We proposed a new method to control traffic lights of intersections based on multi-agent system called CITL. The main agents of CITL are as follows: proposed system get traffic load of each intersection. Next, uses two-step K-means algorithm to find adjacent intersections. Then, it uses evolutionary computing algorithm to scheduling the traffic lights and also it has used exchange of messages between different agents to control possible fluctuations. The main capabilities and advantages of proposed method for simulating a multi-agent system for intelligent urban traffic control at the intersection are as follows: The number of intersection is considered infinite. Also, the distance between intersections is not limited and adjustable. Furthermore, this method is applicable to multi-ways. Due to use multi-agent technique and clustering algorithm for intersections, proposed method has been performed distributed processing which avoid excessive computational load processing in each cycle of traffic lights. We used Tropos methodology and JADE library. For evaluation, this method tested with 2000 laboratory data in both low and heavy traffic areas

In addition, we compared main features of our proposed method with other implemented systems [35, 36] and we can reduce delay time of vehicles, compared to fixed-time system and similar multi-agent methods in [18, 27]. The average run-time in the sequential cycles in our combinatorial method is less than using the other single methods. Using our method, there were 24.5% and 35.5% improvement respectively in delay time of vehicles, compared to the fixed-time methods. Also, we had9.58%

and 9.79% improvement in heavy-traffic and low-traffic areas in delay time, compared to the similar multi-agent systems.

For future works, we want to test CITL in vast area of city with actual data. For this propose, we may need to modify or improve steps of our method to be applicable for actual data.

References

[1] Purnasir Kennedy M, PirkhahAqbash A, Derakhshan F. Urban Traffic Control and bus priority based on multi-agent systems. Seventh International Conference on Information and Knowledge Technology. Uremia University. Iran: Uremia ; 2015.

[2] Alam M, Ferreira J, Fonseca J. Intelligent Transportation Systems. 2015; p. 1-17.

[3] Araghi S, Khosravi A, Creighton D. A review on computational intelligence methods for controlling traffic signal timing. Expert Systems with Applications. 2015;42(3):1538-1550.

[4] Jamshidnejad A, Papamichail I, Papageorgiou M, De Schutter B. A Model-Predictive Urban Traffic Control Approach with a Modified Flow Model and Endpoint Penalties. IFAC-PapersOnLine. 2016;49(3):147-152.

[5] SELBY DPOWELL R. URBAN TRAFFIC CONTROL SYSTEM INCORPORATING SCOOT: DESIGN AND IMPLEMENTATION. Proceedings of the Institution of Civil Engineers. 1987;82(5):903-920.

[6] SELBY DPOWELL R. URBAN TRAFFIC CONTROL SYSTEM INCORPORATING SCOOT: DESIGN AND IMPLEMENTATION. Proceedings of the Institution of Civil Engineers. 1987;82(5):903-920.

[7] SCATS: The benchmark in urban traffic control [Internet]. Scats.com.au. 2016 [cited 21 December 2016]. Available from: http://www.scats.com.au

[8] Zhao YTian Z. An Overview of the Usage of Adaptive Signal Control System in the United States of America. Applied Mechanics and Materials. 2012;178-181:2591-2598.

[9] KokMun N, Mamun Bin Ibne R, MohdAlauddinMohd A, Tae Guy Chang. A brief survey on advances of control and intelligent system methods for traffic-responsive control of urban networks. TehnickiVjesnik. 2013;20(3):555-562

[10] 7. Sutton RBarto A. Reinforcement learning. 1st ed. Cambridge, Mass.: MIT Press; 1998.

[11] Araghi S, Khosravi A, Creighton D. Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network. Expert Systems with Applications. 2015;42(9):4422-4431.

[12] Lian Z, Wang H, Li D. Design of a Novel Variable Bit Rate Video Traffic Prediction Model Based on Neural Networks. Advanced Materials Research. 2014;989-994:4143-4146.

[13] Zhang MLu Y. Adaptive Network Traffic Prediction Algorithm based on BP Neural Network. International Journal of Future Generation Communication and Networking. 2015;8(5):195-206.

[14] Qiu J, Gao H, Ding S. Recent Advances on Fuzzy-Model-Based Nonlinear Networked Control Systems: A Survey. IEEE Transactions on Industrial Electronics. 2016;63(2):1207-1217.

[15] Daneshfar F, RavanJamJah a, Mansoori F, ZahirAzami B. Vehicular Technology Conference. IEEE 69th; 2009.

[16] Tang J, Zhang G, Wang Y, Wang H, Liu F. A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. Transportation Research Part C: Emerging Technologies. 2015;51:29-40.

[17] Odeh S. Management of An Intelligent Traffic Light System by Using Genetic Algorithm. Journal of Image and Graphics. 2013;1(2):90-93.

[18] Singh L, Tripathi S, Arora H. time optimization for traffic signal control using genetic algorithm. Internatinal Journal of Recent Trends in Engineering. 2009;2(2).

[19] Gora P A. Genetic Algorithm Approach to Optimization of Vehicular Traffic in Cities by Means of Configuring Traffic Lights. in Emerging Intelligent Technologies in Industry. 2011; 1-10.

[20] Roozemond D. Using autonomous intelligent agents for urban traffic control systems. In Proc of the 6th World Congress on Intelligent Transport Systems. 1999.

[21] Hernandez J.Z, CuenaJ,Molina. M.Real-time Traffic Management through Knowledge-based Models: The TRYS approach. ERUDIT Tutorial on Intelligent Traffic Management Models. Finnland. 1999.

[22] Abdoos M, Mozayani N, Bazzan A. Holonic multi-agent system for traffic signals control. Engineering Applications of Artificial Intelligence. 2013;26(5-6):1575-1587.

[23] Zhu F, Aziz H, Qian X, Ukkusuri S. A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework. Transportation Research Part C: Emerging Technologies. 2015;58:487-501.

[24] DresnerKStone P. Multi-agenttraffic management: an improved intersection control mechanism. in Proceedings of the fourth international joint conference on Autonomous agents and multi-agentsystems. 2005.

[25] Derakhshan F, Khezerlou F, Shahpasandi N. Urban traffic control at intersections based on multi-agentsystems. First International Conference on New Research Achievements in Electrical and Computer Engineering. Amir Kabir University of Tehran , Iran: Tehran; 2016.

[26] Derakhshan F, Khezerlou F, Shahpasandi N. Timing of Traffic Lights at Intersections based on Multi-Agent systems. First National Conference on Smart. Qom University , Iran: Qom; 2016.

[27] Derakhshan F, Khezerlou F. Design and Implementation of an Intelligent Combined Method for Urban Traffic Control at Intersections.Journal of Electrical Engineering, University of Tabriz, Iran: Tabriz ;(accepted) 2016.

[28] Caire G. Jade Site | Java Agent DEvelopment Framework [Internet]. Jade.tilab.com. 2009 [cited 30 December 2016]. Available from: http://jade.tilab.com

[29] Bergenti F, Gleizes M, Zambonelli F. Methodologies and software engineering for agent systems. 1st ed. Boston [Mass.]: Kluwer Academic; 2004.

[30] Bresciani P, Perini A, Giorgini P, Giunchiglia F, Mylopoulos J. Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems. 2004;8(3):203-236.

[31] Fischer K, Hahn C, Mora C. Agent-oriented software engineering: a model-driven approach. International Journal of Agent-Oriented Software Engineering. 2007;1(3/4):334.

[32] J. A SURVEY ON USING WIRELESS SIGNALS FOR ROAD TRAFFIC DETECTION. International Journal of Research in Engineering and Technology. 2014;03(13):97-102.
[33] KononenkoIKukar M. Machine learning and data mining. 1st ed. Chichester, UK: Horwood Publishing; 2007.
[34] Floares A. Computational intelligence. 1st ed. New York: Nova Science Publishers, Inc.; 2013.
[35] Malek s, Denney R, Halkias J. ADVANCED TRANSPORTATION MANAGEMENT TECHNOLOGIES PARTICIPANT NOTEBOOK. 1st ed. Washington DC: Federal Highway Administration; 1997.
[36] TTIC :Scientific Information and specialized freight traffic [Internet]. http://www.t tic.ir 2016 [cited 19 February 2015]. Available from:]. http://www.t tic.ir