Multiagent architecture applied in decentralized real-time urban road traffic control

Ovidiu Banias, Radu-Emil Precup, Daniel Curiac
Politehnica University of Timisoara, Timisoara, Romania
ovidiu.banias@aut.upt.ro

Abstract—In this paper we present a decentralized urban road traffic control solution over a multiagent architecture. Based on the proposed architecture and real-time traffic information received from the traffic sensing system, we implemented local decision making at crossroad level, every crossroad traffic lights being controlled by one local agent. Finally, we present a case study for different urban traffic conditions, making a parallel between our multiagent decentralized solution and static control (a priori set traffic lights timers) proving the improvements and benefits of our proposed strategy. In the end we conclude pointing the future directions of development and improvement of presented decentralized real-time control strategy.

I. INTRODUCTION

Urban road traffic control is a major problem in the big cities of today. The continuous growth of the number of vehicles and pedestrians in the urban traffic could not be stopped, the only solution to massive decongestion being the development of new infrastructure along with the implementation of dedicated intelligent control systems. Avoiding traffic jams is very important nowadays for the environment and also for the economy, and the Intelligent Transport Systems (ITS) [1] are necessary for traffic safety and road traffic decongestion [2].

The development of dynamic control strategies is the only solution with rapid impact to road traffic decongestion. On the other hand, building new infrastructure and offering growth of road capacity at least at the same rate as the growth rate of vehicles and pedestrians, is the only long lasting solution without affecting the adaptive control strategies previously implemented.

New approaches in urban road traffic decongestion are the optimizations through decentralized control. The old and most of current implemented solutions for traffic lights command in cities nowadays are centralized, the city traffic being hierarchically controlled, making decisions on higher levels[3].

In this paper we present a multiagent control architecture for urban road traffic decongestion and the results of decentralized control through implementation of local decision making at agent level for one complex crossroad. The paper is organized as follows. Section II describes the proposed multiagent architecture for decentralized urban road traffic control. In Section III we present our local traffic agent decision making solution for traffic lights command. In Section IV we present the results of decentralized control with local decision making at crossroad level based on real-time traffic information received from the sensing system, in comparison with static traffic light command. We conclude in Section V, also pointing future work.

II. MULTIAGENT ARCHITECTURE FOR URBAN ROAD TRAFFIC DECONGESTION

The proposed multiagent architecture has two hierarchic levels. First level is represented by the local traffic control agents (LTA), and the second level is represented by then zone traffic control agents (ZTA), every level having its own importance in the scope of distributed control. The local agent (LTA) will take traffic control decisions in each crossroad, with

Fig. 1. Multiagent architecture for urban road traffic control

Fig. 2. Traffic agents intercommunication
the goal of optimizing parameters like queue lengths, mean velocity, waiting time. The zone agent (ZTA) will interact with the LTA by sending messages in special situations like need for traffic lights synchronization and need for certain street blocking in case of reported accidents or road infrastructure maintenance. The multiagent architecture [4] is the solution for decentralization.

Our proposed architecture has the following features: a) the decision making is distributed, b) there is no agent controlling the whole traffic process, only tasks of coordination and collaboration, c) each agent makes local decisions accordingly with the information received from road traffic sensing system and from neighbor agents. (Fig. 1)

From Fig. 1 could be observed that each zonal agent $ZTA_{i}, i = 1, z$ communicates with all local agents $LTA_{i,j}, j = 1, l_{i}$ under his jurisdiction, where $z$ - the amount of zone traffic agents, $l_{i}$ - the amount of local traffic agents under coordination of zone agent number $i$. The control of the traffic lights in each crossroad is made by the local agents in cooperation with neighbor nodes and under coordination of the zone agent.

The communication between traffic agents is presented in Fig. 2. There are three types of communication between agents: LTA to LTA, ZTA to LTA and ZTA to ZTA. In the LTA to LTA communication, local agents communicate with each other after the following principle: first of all, a traffic zone is formed from a set of crossroads arranged in a certain order like a queue, than every local agent $i$ from the respective zone sends messages to the neighbor local agent $i + 1$. In this way the agent $i + 1$ receives information about the vehicles approaching from the previous crossroad and could take decisions accordingly to that. Another type of communication between the zone traffic agent and the set of local agents coordinated by it (ZTA to LTA communication) is closely related to the LTA to LTA communication. This type of communication is used for traffic lights synchronization and street blocking events, together with LTA to LTA communication offering important inputs for the agents decision making.

The last type of communication, ZTA to ZTA communication is used only for statistics and reports.

The communication protocol between agents is simple. The transmitted message from the sender agent is composed of five blocks: the id of the sender, the id of the receiver, the message body/information, start and finish time of message validity (Fig. 3). Splitting the sent message like described above, make it easy to parse by the receiver agent and interpret the traffic information or coordination instructions. Each message body is known at both ends, being stored in the agent’s database as LILO list.

Cooperation between agents is another important characteristic of our multiagent systems [5]. The general goal is to achieve distributed road traffic control and optimization by local decision making at the level of local traffic agents [3]. This could be done only by cooperation between the agents from each crossroad and also the zone traffic agent. The architecture presented here is able to support implementation of artificial intelligence solutions like knowledge based real time expert systems. The cooperation in urban road traffic optimization between neighbor crossroads of a certain traffic zone is done by selecting certain traffic lights command strategies from knowledge data-base regarding the neighbor agent previously chosen solution.

Authentication of the agents and securing the messanges in this context is another issue of the multiagent architecture. Due to the different types of agents communication enumerated above, message authentication is really necessary in real environment deployments. Message Authentication Codes (MAC) [6] are the solution for the multiagent architecture, being able to guaranty the message authenticity over an insecure channel.

III. LOCAL TRAFFIC AGENT

The local traffic agent offers automated solutions for traffic lights control, with the goal of decongesting the urban traffic in one crossroad. This agent is very important for the traffic lights controlling because from the total amount of running time (the period of time from starting to operate to the shutdown), the
most of the decisions are made at this local level without any coordination from the zonal agent. The interaction with zonal agent is made only in the rush hours when local strategies could be substituted with zonal coordination by traffic lights synchronization strategies. Also zonal coordination is used in exceptional cases when certain streets are blocked due to accidents or need to be blocked for road infrastructure maintenance. In this paper we implemented the LTA’s decision making as a knowledge based real time expert system (Fig. 4).

The LTA has the ability to learn from statistics and previous traffic lights commands. The implemented learning strategy is deductive [7], the agent being able to automatically learn and estimate the amount of time needed to free a certain queue length at a certain period of time. The learning method is always adapted to the latest measurements from the traffic, not forgetting about the previous statistics. This denotes the stochastic character of the LTA - different traffic lights commands depending of the current traffic conditions.

Offering automated decongestion solutions at crossroad level, and having inputs the waiting queue lengths and the road infrastructure, this local agent could be regarded as an automated substitute for the traffic policeman. We can make a parallel between the local traffic agent and the traffic policeman: the traffic information obtained from the road traffic detector represents for the LTA what the eyes represent for the traffic policeman, the artificial intelligence solutions and automated control strategies will make decisions for LTA as the human brain will make decisions for the traffic policeman (Fig. 5).

As follows we will present the implementation of the local traffic agent as a knowledge base real time expert system. The agent will make real time decisions for traffic lights controlling, accordingly to the real time traffic information received from the traffic detectors and taking in consideration previous traffic lights controls and statistics from the knowledge base.

We consider [8]:

- \( I_{NI} \) - street number \( i \) entering the crossroad,
- \( OUT_j \) - street number \( j \) leaving the crossroad,
- \( IN_i,OUT_j \) - traffic event pointing out that the vehicle enters the crossroad from street number \( i \) and leaves the crossroad on street number \( j \),
- \( C_i \) - traffic event highlighting that the pedestrians are crossing the street number \( i \).

We define the following set of possible events in the crossroad, \( E \), useful for later defined events graph:

\[
E = \{ IN_i,OUT_j / i = 1, nrIN, j = 1, nrOUT \} \bigcup \{ C_i / k = 1, nrc \}
\]  

(1)

where:

- \( nrIN \) - number of streets entering the given crossroad,
- \( nrOUT \) - number of streets leaving the given crossroad,
- \( nrc \) - number of pedestrian crossings in given crossroad.

We will define also the graph \( G(E, A) \) of possible events in one crossroad. In this context, in relation with (1), \( E \) is the set of vertices, each vertex being represented by one event, and \( A \) is the adjacency matrix, defining edges between vertices:

\[
A_{x,y} = \begin{cases} 
1, & E_x \& E_y \text{ are not criss-crossed} \\
0, & E_x \& E_y \text{ are criss-crossed}
\end{cases}
\]  

(2)

with \( A_{x,y} \) - elements of the adjacency matrix of graph \( G \). As shown in (2), there is an edge between two events if those events are criss-crossing each other. From the traffic control point of view, one edge between two events means those events should always have different traffic lights color.

In this paper we focus on the changing traffic lights case where at one traffic light phase only one street \( IN_i \) entering the crossroad will get “green color” for all the possible directions \( OUT_j \). This case could be easily generalized but is not the subject of the current paper, being rather a matter of optimized control than a matter of multiagent control. Grouping all the vehicles events \( IN_i,OUT_j \) in \( GR \), and depending the road infrastructure adding or not the pedestrian groups we obtain the total number of event groups with the property that there are no two events in the same group criss-crossing each other. Based on this apriori grouping and the real time queue lengths, the agent will make decisions on the sequence of green phases for each group and the corresponding amount of time.

We define the waiting time limit threshold for each event not getting “green color”, in other words the maximum waiting time on red traffic light color:

\[
\tau_i \in \{ \ell_{\text{min}}, \ell_{\text{min}} + 1, ..., \ell_{\text{max}} \}, \ell_{\text{min}}, \ell_{\text{max}} \in \mathbb{N},
\]

(3)

where \( \ell_{\text{min}} \) and \( \ell_{\text{max}} \) are apriori set.

This time limit could be found in almost all crossroads control, being important for the drivers in the traffic. A higher value will make the drivers nervous for too much waiting and a lower value will make the traffic lights control difficult and hard to optimize. After each traffic light phase the local traffic agent will decrease the current waiting time \( \theta_i \) for every event that didn’t receive green light and will set to the defined
threshold the current waiting time for every event that received green light:

\[
\theta_i(t) = \begin{cases} 
\max(\theta_i(t - 1) - \delta_{i-1}, 0), & \text{if the event } \ast \text{ has received red} \\
\tau_i, & \text{if the event } \ast \text{ has received green}
\end{cases}
\]  

(4)

In the following paragraph we will present a general pseudocode of the local traffic agent implementation:

<table>
<thead>
<tr>
<th>LocalTrafficAgents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin</td>
</tr>
<tr>
<td>InitAgent 'init variables</td>
</tr>
<tr>
<td>Repeat</td>
</tr>
<tr>
<td>ReadQueues; 'reads the real–time queue lengths in meters from traffic sensing system</td>
</tr>
<tr>
<td>CalculateQueueLengthsInMeters; 'estimates the next queue lengths in meters</td>
</tr>
<tr>
<td>CalculateQueueLengthsInTime; 'estimates the queue lengths in seconds</td>
</tr>
<tr>
<td>CalculateTimeLimits; 'calculates the current waiting time limit</td>
</tr>
<tr>
<td>FindNextGreen; 'returns the next street receiving &quot;green&quot; + amount of time</td>
</tr>
<tr>
<td>TrafficLightCommands; 'sends commands to change traffic light colors</td>
</tr>
<tr>
<td>SaveHistory; 'saves the traffic control history</td>
</tr>
<tr>
<td>StatisticsToKBS; 'learning from statistics</td>
</tr>
<tr>
<td>Until elapsedTime&lt;MaximumTime OR stopCommand ' repeats until the running time exceeds a previous set threshold or receives stop command</td>
</tr>
<tr>
<td>End</td>
</tr>
</tbody>
</table>

As presented above, after initialization the local traffic agent cycles until the needed running time exceeds a previous set threshold or receives the stop command. During each cycle the LTA will make local decisions and command the crossroads traffic lights based on real-time traffic information received from the sensing system and based on saved history and statistics. Each LTA cycle will start couple of seconds earlier than the finishing of current green traffic light phase, to have time for computation, and at each cycle will be chosen only one street for "green light" with the associated amount on time.

IV. CASE STUDY

In this section we present a case study starting from the LTA implementation described in the previous section. We make a comparison between our proposed implementation for the local traffic agent and the statical traffic control solution still used today based on Split strategy. The simulation is made for a simple four street crossroad like presented in the Figure 6, and could be easily extended to almost any complex crossroad.

By this comparison we want to emphasize the improvements brought by our proposed multiagent implementation and we show this fact by recording values of three traffic performance appreciation variables over the whole simulation period. The first appreciation variable \( s_m(t) \) measures the mean speed per traffic light phase. The second appreciation variable \( d_m(t) \) measures the sum of the dead times after each traffic light phase. We define the dead time as the amount of time in seconds unused by the waiting queues from the green light amount of time. The third appreciation variable \( l_m(t) \) measures the sum of the queue lengths after each traffic light phase.

We identified three test cases for different road traffic flows:

- high traffic density measured in [veh/m] is characterized by arriving speeds greater than 3 [m/s],
- medium traffic density characterized by arriving speeds in the interval \([1.5, 3]\) [m/s],
- low traffic density characterized by arriving speeds in the interval \([0, 1.5]\) [m/s].

We simulated both solutions, our proposed LTA solution and the statical control solution on each of the described traffic conditions. The restrictions and the settings of our simulation are the following: the arrival speed of the vehicles at each crossroad are considered constant per each minute (with the possibility to set arrival speeds at every minute), the crossroad has four streets (with the possibility of extension), the departure speeds are apriori set and the LTA solution will continuously learn and update these speeds, the running time is two hours for each situation (high, medium, and low traffic density).

In each of the Figures 7, 8, and 9 there are three graphs representing the comparison between our proposed LTA solution and the statical control solution, one for each of the measured performance appreciation variables. The comparison between the two solutions (LTA vs. static) is represented in the graph by two different functions, the LTA solution is plotted with
The results of the simulation in conditions of high traffic density (rush hours) are presented in Fig. 7. From the graph 7.(a) we observe that at high traffic flow density after a short period of time the measured appreciation variable \(s_m(t)\) is almost the same for both solutions. From the graph 7.(b) we observe that the measured variable \(d_m(t)\) is again almost the same, due to the big waiting queues for each street and the inability of the traffic light to serve almost all the vehicles in one queue at one traffic light phase. From the graph 7.(c) we observe that the appreciation variable \(l_m(t)\) becomes stable for both solutions after a short period of time. We can conclude that for high traffic density conditions, our proposed solution doesn’t make improvements to the traffic flow, due to the large waiting queues on each street.

The results of the simulation in conditions of medium traffic density (normal traffic) are presented in Fig. 8. From the graph 8.(a) we observe that the appreciation variable \(s_m(t)\) records almost the same values for both simulated solutions. From the graph 8.(b) we observe that the appreciation variable \(d_m(t)\) records values about 12% higher for the statical solution. The
proposed control method eliminates the dead times during traffic lights green phases. From the graph 8.(c) we observe that the measured variable \( l_m(t) \) has a continuous growth in favor of our LTA solution up to 5 times in two hours of simulation. We can conclude that for medium traffic density conditions, our proposed solution is better than the statical control solution, the appreciation variables recording improvements in the elimination of the dead times and in reducing the overall queue lengths.

The results of the simulation in conditions of low traffic density (free traffic) are presented in Fig. 9. From the graph 9.(a) we observe that appreciation variable \( s_m(t) \) records almost the same values for both simulated solutions. From the graph 9.(b) we observe that the appreciation variable \( d_m(t) \) records values about 50% higher for the statical solution, the proposed solution eliminating the dead times. From the graph 9.(c) we observe that the appreciation variable \( l_m(t) \) records values about 500% higher for the statical solution. We can conclude that for low traffic density conditions, our proposed solution is much better than the statical solution.

V. Conclusions

Urban road traffic congestion became an important problem nowadays, the only solution to this problem being the development of new infrastructure and intelligent control systems. In this paper we proposed a multiagent urban road traffic architecture and a local traffic agent implementation. We also presented three case studies (high, medium and low traffic density conditions) by comparing our local traffic agent decision making solution and the statical traffic control solution. We showed by simulation that our proposed strategy is better in the most of the cases, and also we showed the importance of decentralized control over multiagent architecture.

Future work will be developed in the direction of improving the current local traffic agent solution and implementing the zone traffic agent algorithm based on presented architecture.

References