Using Cooperative Mediation to Coordinate Traffic Lights: a Case Study

Denise de Oliveira and Ana L. C. Bazzan
Instituto de Informática, UFRGS
Caixa Postal 15064 CEP 91.501-970
Porto Alegre, RS, Brazil
{edenise, bazzan}@inf.ufrgs.br

Victor Lesser
University of Massachusetts
Department of Computer Science
Amherst, MA 01003
lesser@cs.umass.edu

ABSTRACT

Several approaches tackle the problem of reducing traffic jams. A class of these approaches deals with coordination of traffic lights in order to allow vehicles traveling in a given direction to pass an arterial without stopping at junctions. In short, classical approaches, which are mostly based on offline and centralized determination of the prioritized direction, are quite inflexible since they cannot cope with dynamic changes in the traffic volume. More flexible approaches have been proposed based on implicit coordination and implicit communication (e.g. derived from game theory and swarm intelligence). These have advantages as well as shortcomings. The present paper presents an approach based on cooperative mediation which is a compromise between totally autonomous coordination with implicit communication and the classical centralized solution. We use a distributed constraint optimization algorithm in a dynamic scenario, showing that the mediation is able to reduce the frequency of miscoordination.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems, Coherence and coordination*

General Terms

Algorithms

Keywords

Coordination, Traffic Control, Constraint Optimization Problems

1. INTRODUCTION

Approaches to reduce traffic jams have been proposed in several disciplines like traffic engineering, physics, and artificial intelligence, among others. A classical one is to coordinate or synchronize traffic lights so that vehicles can traverse an arterial in one traffic direction, with a specific speed, without stopping as in [4, 8]. Thus, coordination here means that if appropriate signal plans are selected to run at the adjacent traffic lights, a "green wave" is built so that drivers do not have to stop at junctions.

Operationally, the parameters of the coordination can be computed offline as well as online. Example of the former is Transyt[8]. There are also adaptive systems like SCOOT [4], which focuses on coordination of traffic lights along a fix path. These approaches work well in traffic networks with defined traffic volume patterns like for instance morning and afternoon peaks. However, in cities where these patterns are not clear, this approach may not be effective. This is clearly the case in big cities where the business centers are no longer located exclusively downtown, in which case the existing approaches are not flexible enough.

One reason why approaches from traffic engineering rely predominately on linear programming (Transyt) or local adjustment of the parameters of the traffic signal plan (SCOOT), is that totally decentralized approaches could impose communication bottlenecks (for the negotiation) and/or would have to end up with a traffic expert mediating the conflicts which could arise. Thus, flexible and robust approaches based on multiagent systems (MAS) are not only attractive, but necessary. In Section 2 we review some of the classical approaches as well as those based on MAS and swarm intelligence.

In the present paper, we formulate the traffic lights coordination problem in a way which is a compromise between totally autonomous coordination with implicit communication, and the classical centralized solution, namely as an on-line optimization problem in order to use cooperative mediation based on the Optimal Asynchronous Partial Overlay (OptAPO) algorithm [6].

The goal of this paper is then twofold: to test the OptAPO algorithm – which was not specifically designed for dynamic environments – and to propose a new approach for synchronization of traffic lights which reduces the need for traffic expert intervention.

In the next section we briefly review some approaches to coordination of traffic lights, as well as the OptAPO algorithm. Section 3 presents the problem of traffic light coordination as one of distributed constraint optimization (DCOP). Section 4 discusses the simulations and the results obtained, as well as the qualitative comparison with the approaches based on implicit coordination and communication.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

We give the conclusions and outline future directions in Section 5.

2. RELATED WORK

2.1 Traffic Light Coordination

The goal of coordinated or synchronized systems is *to synchronize the traffic signals in adjacent intersections* in order to allow vehicles, traveling at a given speed, to cross those without stopping at red lights. We use the terms intersections, crossing, junction, and traffic light interchangeably since in each intersections, *only one signal-timing plan* runs in a set of traffic lights (despite the fact that one sees two or three of these) so that the set of traffic lights must be seen as a single entity.

Signalized intersections are controlled by signal-timing plans which are implemented at traffic lights. A signal-timing plan (we use signal plan for short) is a unique set of timing parameters comprising the cycle length L (the length of time for the complete sequence of the phase changes), and the split (the division of the cycle length among the various movements or phases).

The criteria for obtaining the optimum signal timing *at a single intersection* is that it should lead to the minimum overall delay at the intersection. Several plans are normally required for an intersection (or set of intersections in the case of a synchronized system) to deal with changes in traffic volume. Well-designed signal plans can achieve acceptable results if they are synchronized.

In general, the more neighbors that are synchronized, the shorter the queues. Synchronization in two opposing directions of an arterial can be achieved depending on the geometry of the arterial: in a Manhattan-like grid where the spacing among intersections is the same, synchronization in opposite directions is possible. Synchronization in four traffic directions is not possible in practice. Therefore, an agent at a junction must *select* a signal plan to give priority to a particular traffic direction.

In order to explain our approach, we will use a Manhattan-like grid, thus allowing synchronization in either north-south *and* south-north or east-west *and* west-east directions. No extension in the approach proposed here is necessary in case this constraint is dropped. However, more signal plans would have to be designed.

Traffic responsive approaches for arterial appeared in the 1980s and, although they have been operating successfully in Europe, they have had limited use in the U.S. One reason is that these systems are complex to operate and pose high costs, both in terms of hardware and communication costs. In any case, traffic responsive systems are designed to consider only a main path (arterial or similar). Besides, a priori determination of the appropriate signal plans for the different times of a day is a complex task that requires a lot of expert knowledge about dynamic traffic flow. A more flexible system which would for instance change the direction of coordination *and the design* of the coordination (who is coordinating with who in which direction), is not reported in the traffic engineering literature.

Therefore, as said before, more robust and flexible approaches for online, adaptive traffic signal coordination are being developed based on multi-agent, swarm intelligence and self-organization.

In [1, 2] a MAS based approach is described in which each traffic light is modeled as an agent, each having a set of pre-defined signal plans to coordinate with neighbors. Different signal plans can be selected in order to coordinate in a given traffic direction or during a pre-defined period of the day. This approach uses techniques of evolutionary game theory: self-interested agents receive a reward or a penalty given by the environment. Moreover, each agent possesses only information about their local traffic states.

The benefits of this approach are threefold: it is not necessary to have a central agent to determine the direction of the coordination; agents can dynamically build subgroups of traffic light coordination which meet their current needs in terms of allowing vehicles to pass in one given direction; it avoids communication between agents when they have to decide in which direction to coordinate, i.e. there is no explicit communication or negotiation.

However, payoff matrices (or at least the utilities and preferences of the agents) are required, i.e these figures have to be explicitly formalized by the designer of the system. This makes the approach time-consuming when many different options of coordination are possible (for example all four directions: south, north, east, and west) and/or the traffic network is complex (for instance, not only a main arterial has to be considered but also many transversal and parallel streets).

In [7] an approach based on swarm intelligence is proposed aiming at relaxing the need of payoff matrices. In this approach, each traffic light behaves like a social insect. The signal plans are seen as tasks to be performed by the insect without any centralized control or task allocation mechanism. The stimulus depends on the number of vehicles waiting or passing the traffic lights, among other things.

Finally, approaches based on self-organization of traffic lights via thresholds[1] or reservation-based systems [3] have still to solve low-level abstraction issues in order to be adopted by traffic engineers and have a chance to be deployed.

2.2 Distributed Constraint Optimization Problem and Cooperative Mediation

A distributed constraint optimization problem (DCOP) is a kind of DisCSP [9], but relates to optimization and not to satisfaction. In a DCOP, each agent is assigned to one ore more variables and these have interdependencies. A DCOP is formally defined by m variables taking their values d from m domains, and a set of constraints and their values. A constraint is associated with a functional relationship C_i, which in turn has a cost function $f_i(d_{i1}, \ldots, d_{ij})$ where each f_i is defined as a Cartesian product $D_{i1} \times \ldots \times D_{ij}$. The problem is to find an assignment $A = \{d_1, \ldots, d_m \mid d_i \in D_i\}$ such that the global cost F is minimized. The global cost[2] is the sum of the cost functions f.

The optimal asynchronous partial overlay (OptAPO) is a cooperative mediation based DCOP protocol. The algorithm, presented and detailled in [6], allows the agents to extend the context they use for local decision-making to a *relationship graph*. An agent is in each node and the links are the problem constraints.

[1] published on line at
`http://www.nature.com/news/2004/041129/full/041129-12.html`

[2] We follow the nomenclature given in [6] where F_i is the global cost for the *particular agent* i

Each constraint or functional relationship has an associated cost; this cost will be zero if there is no conflict between the agents in this relationship. Within its graph, one of the agents has to act as a mediator, computing a solution for this extended context and recommending values for the variables associated with the agents involved in the mediation session. This is possible because agents construct a *good_list* – which holds the names of agents that have direct or indirect relationship to the list owner – and an *agent_view* to hold the names, values, domains, and functional relationships of related agents.

During the problem solving process, each agent tries to improve the value of its subproblem (the one it can solve within its relationship graph). The priority to take the mediation role will be given to the agent with more information about the problem.

The algorithm has three stages: initialization, checking the agent view, and mediation. Details of these stages can be found in [6]; we give here a brief description and an example from the traffic light scenario (Section 3.2). During the initialization, the agent sets its variables: current value (d_i), variable's name (x_i), priority (p_i), domain (D_i), functional relationships (C_i), good_list and agent_view.

The agents' goal it to improve the solution of its subproblem (represented by F_i). Thus, during the second stage, the agent_view is used to calculate the current cost F_i within the relationship subgraph given by i's good_list. If $F_i > F_i$ (F_i being the optimal value of the subsystem), then agent i conducts either a passive or an active mediation session, after what the value of F_i is recomputed.

Agent i will set a passive mediation if its priority to mediate is lower then other agent in the subsystem, otherwise it will set a temporary mediation flag ($m_i^{'}$) as active. If an active mediation flag is on, the agent can actually mediate only if there is no other agent with a higher priority to mediate and with an active mediation flag. The agent tests if a change in its local value would cause a local cost to reach the optimal cost. If it does, then the agent changes its value and does not start the mediation process. If the agent is with a passive mediation flag, it starts a passive mediation process.

In the mediation stage, agents receiving a mediation request either evaluate or send a wait message. Evaluation means looking at each of the domain elements, labeling them with the names of the agents which share functional relationships with cost $f_i > f_i$, and returning these in a message. The mediator conducts a branch-and-bound search to minimize the cost of the subproblem in its good_list (primary criteria), as well as the costs for agents outside the mediation session.

In [6] OptAPO was used in the graph-coloring problem with an static assignment for different number of variables. As said, one of the goals of this paper is to use the OptAPO in a dynamic environment. In the next section we formulate the coordination of traffic lights as a DCOP.

3. DISTRIBUTED COORDINATION OF TRAFFIC LIGHTS USING COOPERATIVE MEDIATION

3.1 Description

As in [6], we also consider only the case where each agent is assigned to a single variable. Therefore, in this scenario, the variables of the DCOP are the coordination direction for each traffic light. Thus, the domain for all variables is given by two possible values of coordination: D={NS/SN, EW/WE}.

The constraints in this problem arise from the fact that, in each node of the graph, a traffic light cannot coordinate with neighbors located in a different direction at the same time. For our purposes here, this means that if a traffic light coordinates with its north and south neighbors, the traffic flow coming from these directions receives more green time; it is impossible to coordinate also with the east and west neighbors. A conflict occurs when two neighbors want to give priority to different directions.

As a measure of effectiveness of coordinated systems (no matter if online or offline), one generally seeks to optimize a weighted combination of stops and delays or any other measure related to the occupancy of the roads (number of vehicles) in the network. In this paper we use the latter: we define a cost function which is based both on the number of incoming vehicles (in a junction) and on whether two adjacent agents are coordinating or not.

Algorithm 1 Calculates the global cost F_i

if $(IncomingVehicles(NS) \geq (IncomingVehicles(WE)$ **then**
  for all j in the North/South direction $\in$ good_list of agent i **do**
    $F_i \leftarrow F_i + f(x_i, x_j)$
  end for
else
  for all j in the East/West direction $\in$ good_list of agent i **do**
    $F_i \leftarrow F_i + f(x_i, x_j)$
  end for
end if

Algorithm 2 Calculates a relation cost $f(x_i, x_j)$

if (d_i is the direction with higher number of incoming vehicles) **then**
  if $(d_i == d_j)$ **then**
    $f(x_i, x_j) \leftarrow 0;$
  else
    $f(x_i, x_j) \leftarrow \frac{IncomingVehiclesFrom(x_j)}{TotalIncomingVehicles()};$
  end if
else
  $f(x_i, x_j) \leftarrow 2 * \frac{IncomingVehiclesFrom(x_j)}{TotalIncomingVehicles()};$
end if

The global cost function for each agent i, calculated as in Algorithm 1, depends on the direction with the highest number of incoming vehicles. If there are more vehicles coming from north and south, then the agent will calculate its global cost (F_i) taking into account the sum of the costs it has, i.e. regarding the functional relationships with the neighbors in these traffic directions. A relationship here is associated with two neighbors, and maps the signal plans both are running to a cost.

The cost of each relationship between two agents ($f(x_i, x_j)$) is computed as in Algorithm 2, based on the direction with the highest number of incoming vehicles (for each crossing) and the current state (given by the coordination direction of the two agents). This cost is calculated as follows:

Situation I if both agents in the relationship are running coordinated plans ($d_i = d_j$) and this plan prioritizes the direction with the highest number of vehicles, then this relationship has cost zero ($f(x_i, x_j) = 0$);

Situation II if agent i is prioritizing the direction with highest number of incoming vehicles but the agent j is not synchronizing in this direction, then the relationship has a cost given by the number of vehicles approaching node x_i from node x_j (given by the $IncomingVehiclesFrom(x_j)$ function), divided by the sum of the incoming vehicles from all neighbors that agent x_j has in its good_list (return value from function $TotalIncomingVehicles()$);

Situation III if the direction prioritized by the agent x_i is not the direction with higher number of incoming vehicles, the cost will by twice the cost of the previous situation, because the agent is not running the best plan for the current traffic situation.

The goal of the optimization problem is to coordinate traffic lights to minimize the global cost (sum of all relationship costs). The interaction cost (f) is always between 0 and 2 as shown above.

We have implemented the traffic scenario and the OptAPO algorithm using a simulation tool called SeSAm (Shell for Simulated Agent Systems), that provides a generic environment for modeling and experimenting with agent–based simulations [5]. In this tool, the agent behavior is modeled as a Finite State Machine (FSM) with four states: *Initialize*, *Check Agent View*, *Mediate* and *Choose Solution*.

Initially, the agents are assigned one of the two possible coordination values (directions). The priority of each agent to mediate is given by the number of the incoming vehicles in the intersection the agent controls. During the initialization, the agent includes itself and all its neighbors in its good_list, because it is directly connect to all of them.

After the initialization all agents go to the *check agent view* state, where they compute new mediate intentions (active or passive) and decides whether to conduct an active or passive mediation. The agent with active mediation flag and higher priority go to the *mediate* state and start a mediation process. The mediator goes to the *choose solution* state, after it evaluates all agents in its good_list and have filled its preferences list.

At the *choose solution* state the agent performs a branch-and-bound search in order to minimize the cost for all agents in its good-list. After choosing the best solution and setting the new values for the agents in the mediation session (agents in its good_list and not involved in other mediation) it returns to the *Check Agent View* state.

3.2 Example

As an example, consider the graph in Figure 1 (left) that represents a traffic network, with 25 nodes (intersections) and 40 associated edges (functional relationships). Dotted circles represent the agents running a plan giving priority to the NS/SN directions, and full line circles agents with EW/WE plans. Figure 1 (right) depicts a portion of that network in a particular configuration of states. Outgoing edges are not shown for sake of clarity.

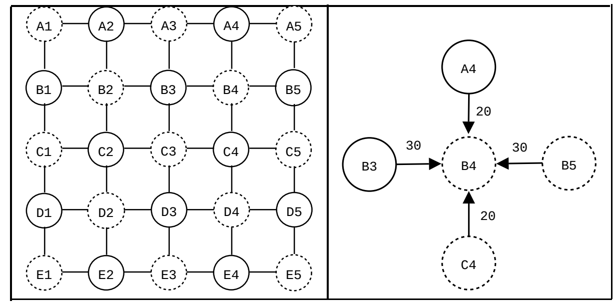

Figure 1: Left: a Network of 25 Intersections (dotted and full-line circles show intersection with SN/NS and EW/WE signal plans respectively). Right: a Particular Configuration for B4 Subsystem

Following the OptAPO algorithm, for node B4 in Figure 1 (right), the priority value is the total number of vehicles coming from the nodes A4, B3, B5, and C4, i.e. $p_{B4} = 100$ because its inputs are: 20 vehicles from A4, 20 from C4, 30 from B5, and 30 from B3. After the initialization, parameters for agent B4 have the following values:

- $p_i = 100$;
- $d_i =$ **NS/SN** (north-south direction);
- $m_i =$ **active** (due to the initialization);
- $mediate =$ **none**;
- $good\_list = \{A4,\ B3,\ B4,\ B5,\ C4\}$;
- $agent\_view = \{A4,\ B3,\ B5,\ C4\}$.

Using the Algorithm 1, B4 compares the sum of incoming vehicles from north and south with the sum of incoming vehicles from other directions. There are 40 vehicles from north and south and 60 from west and east, indicating a higher traffic volume in direction EW/WE. Thus the relationships to be considered by B4 are with B3 and B5. The relationships with B3 and B5 fit "Situation III" (B4 is not giving priority to the direction with the highest traffic volume). Thus, from Algorithm 2:

$$f(B4, B5) = f(B4, B3) = 2 * \frac{30}{100}$$

The B4 global value is then:

$$F_i = 2 * \frac{30}{100} + 2 * \frac{30}{100} = 1.2$$

Since B4 has a global cost higher than the optimal value of the subsystem ($F_i = 0$) it conducts either an *active* or a *passive* mediation session. As B4 is the highest priority agent in its list, it begins an active session with the agents A4, B3, B5, and C4. Thus, at this point, m'_i for B4 is set to active.

For simplicity, let us consider that all agents not in Figure 1 (right) are with the NS/SN priority and all have 30 vehicles coming from EW/WE and 10 vehicles coming from NS/SN direction. The first attempt to solve the problem is to make a local change in the subsystem, with B4 changing to plan EW/WE. This would yield a non-zero cost:

Table 1: Preferences Table

Node	Priority Options	Conflict	Cost
A4	NS/SN	A3	0.86
		A5	0.86
	EW/WE	A3	0.43
		A5	0.43
B3	NS/SN	B2	0.75
		B4	0.75
	EW/WE	B2	0.375
		B4	0.375
B5	NS/SN	B4	1.2
	EW/WE	B4	0.6
C4	NS/SN	C3	0.75
		C5	0.75
	EW/WE	C3	0.375
		C5	0.375

$f(B4, B5) = 0.3$ because B5 would continue with the NS/SN plan and thus the $f(B4, B5)$ cost would fit "Situation II".

Since the local change does not lead to an optimal state, B4 sends "evaluate?" messages to A4, B3, B5, and C4 and constructs a table with the information returned by agents in the session (Table 1). In this table the values come from those agents returning information about each conflict generated by setting its d_i tentatively to the two values of coordination. For instance, if A4 would change to NS/SN plan, this would lead to a conflict with agents A3 and A5, with both $f(A4, A5)$ and $f(A4, A3)$ fitting "Situation III" with a 0.86 cost ($f(A4, A5) == f(A4, A5) = 2 * \frac{30}{70}$). This information is used by the mediator agent to minimize the cost of the conflicts with agents out of the session.

At the next step, the mediator agent (B4) performs a branch-and-bound search and finds $F_i = 0$ for its good_list. Using this solution, B4 changes its own direction, as well as B5 and C4, to EW/WE, yielding $f(B4, B5) = f(B4, C5) = 0$ ("Situation I"). A pre-existing conflicting situation regarding C4 with C3 and C5 remains since C3 and C5 are not in B4 agent_view list. However, this situation has now a lower cost (originally it was 0.75 and it is now 0.375, as we can see in Table 1). Note in the Table 1 that the conflicts of C4 are with agents outside the mediation, so B4 can only change the C5 value. These conflicts with agents external to the mediation process causes B4 to add C3 and C5 to its agent_view and good_list.

In a similar way, B4 is included into C3 and C5 agent_view and good_list because B4 has neighborhood relationship with C4. These additions create new "artificial" interactions between the agents and only happens if the mediator agent is conducting an active mediation and has the highest priority. After the B4 mediation, the OptApo process continues until each of the agents have justified their costs and found a global solution with less violation of constraints.

4. SCENARIO AND EXPERIMENTS

4.1 Scenario

We use the scenario depicted in Figure 1, representing a traffic network which is a 5x5 Manhattan-like grid, with a traffic light (agent) in each junction. There are 25 nodes and 40 edges or sections (i.e. 40 functional interactions between the variables). Each of these sections has a capacity of 30 vehicles per simulation cycle (in each traffic direction). The actual number of vehicles inserted in the sources, located in each border node, is given by a Gaussian distribution with mean μ and standard deviation σ.

Vehicles do not change direction during the simulation and upon arriving at the borders of the network they are removed from it. For instance, a vehicle inserted at the node A1 with the the South direction, will be removed at the node E1.

Traffic lights normally have a set of signal plans (for different traffic conditions and/or time of the day). We consider here only two plans, each allowing more green time to a given direction. These signal plans have two phases, one allowing green time to direction north-south(NS)/south-north(SN) and other to east-west(EW)/west-east(WE). All signal plans have cycle time of 90 seconds and phases of 60 and 30 seconds. Therefore, the smallest unit of time we consider in the simulation is one-third of the cycle time (30 seconds). The graphs shown in this section all depict this unit as one time step. Speed and topology constraints are so that 10 vehicles can pass the junction in 30 seconds.

At the beginning of the simulation, agents A1, A3, A5, B2, B4, C1, C3, C5, D2, D4, E1, E3, and E5 have d = NS/SN while others have d =EW/WE (Figure 1). This initial configuration makes all agents start with neighbors with different plans, thus there is no coordination at all. Unless stated, at the beginning of all simulation cases, the values of the variables are as these.

We have simulated four cases:

Case I the pattern of traffic volume is generated once and does not change;

Case II a change in the traffic volume is artificially inserted by us at the time step 400 to test the restart of the mediations;

Case III two traffic changes are inserted;

Case IV situation with fixed coordination (no mediation).

4.2 Results

In the case I, there are no changes in the traffic volume during the whole simulation time, i.e. the sources generate vehicles according to a unique Gaussian distribution ($\mu = 8$ and $\sigma = 2$). In this simulation, we expect agents to start the mediation and eventually reach a configuration of minimum cost.

In Figure 2 we plot the global cost (sum of the costs f of each edge or relationship) and the number of mediations (inset plot). The simulation starts with a high cost (25) and during the first 120 time steps the mediation goes on and the cost gets lower and lower, except for some peaks. These peaks happen because vehicles stop at red lights, thus increasing the queues (and hence costs). At green light, the queue decreases (but not necessarily empties). This pattern occurs due to the cycle time of the signal plan.

The mediation sets variables values (plans) for each agent so that the cost stabilizes around 12. Remember that this is a sum over all nodes and that zero cost is not possible because each node cannot simultaneously coordinate with *all* neighbors. After the stabilization, only minor changes occur but these oscillations are due to the Gaussian and do not cause new mediation processes to start because the

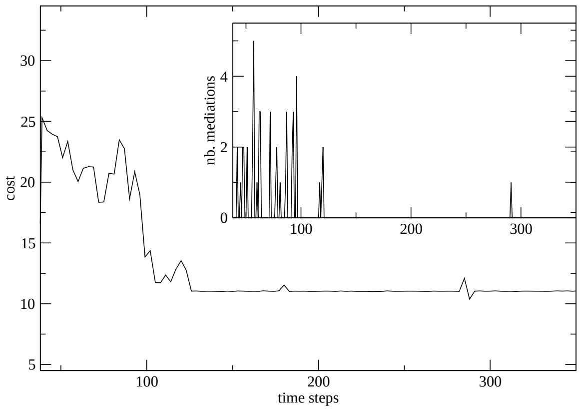

Figure 2: Scenario with constant traffic volume. Cost along time steps. Inset plot: number of mediations.

traffic volume does not change in a way that would require a change in plans (see Algorithm 1).

After the mediation, we notice blocks of coordination, as shown in Figure 3. This situation is much different from the starting one depicted in Figure 1: nodes B1, C1, D1; A3, B3, C3, D3; and A5, B5, C5 all have plans coordinating NS/SN, while nodes A1, A2; and E1, E2, E3, E4 are coordinating in direction EW/WE.

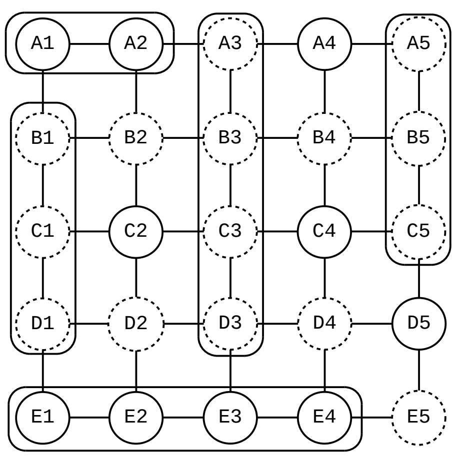

Figure 3: Groups formed after the mediation process.

The inset plot in Figure 2 shows that few agents conduct active mediations. This is important because the number of steps necessary to reach an stable state depends on how many agents need to mediate and if the mediators have enough information about the system (the more information gathering is necessary the longer the mediation).

In the case II, we artificially insert a change in traffic volume at time 400. As shown in Figure 5 (left), the changes are at the sources located in nodes A3, A4, E3, and E4, inserting more vehicles in the NS/SN direction. The new gaussian distribution in all these nodes is $\mu = 25$ and $\sigma = 1$. The other sources (other borders) are set to $\mu = 3$ and $\sigma = 1$. Given this, it is expected that something similar to case I happen in the beginning of the simulation (when the

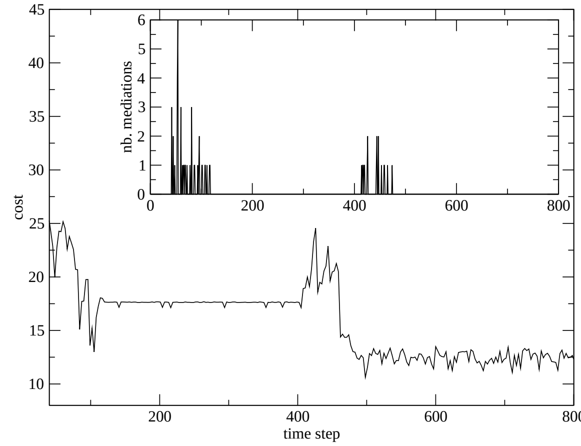

Figure 4: Situation with one change in the traffic volume (at time step 400).

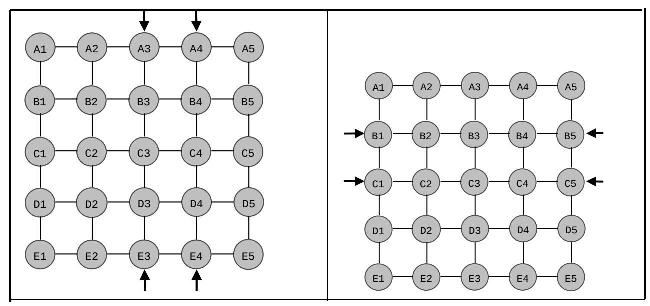

Figure 5: Change in Sources for Case II (left) and Case III (right).

distribution is the same as in case I). After step 400, with the change, we expect that costs increase again and a new mediation starts in order to reduce costs.

Figure 4 shows that the agents have indeed found a solution for the initial traffic pattern (stabilizing cost around $F = 18$). After the volume changes at time step 400, the cost increases, causing the agents to restart the mediation. At time 500 a new solution is found, yielding a global cost of around 12. The new solution sets the variables for agents A3, B3, ..., E3 and A4, B4, ..., E4 to the value NS/SN as expected since the traffic volume has increased in nodes A3, A4, E3, and E4. This shows that the mediation has worked and that agents are coordinating in a better form.

In the case III, two changes in the traffic volume occur: at step 400 and 700. The former occurs the same way as in the previous scenario. The second is depicted in Figure 5 (right): sources at nodes B1, C1, B5 and C5 now have $\mu = 25$ and $\sigma = 1$. In both cases all other sources are set to $\mu = 3$ and $\sigma = 1$. Here we expect coordination to built up among the second and third row of agents.

Figure 6 shows the effect of these two changes in traffic volume. Before step 400 we note the same stabilization pattern as in previous cases. After both changes, the mediation sessions restart and we can see that the agents reach a solution which sets costs to levels lower than the one in the initial configuration (complete miscoordination). After step 800 the coordination indeed involves agents in B1 to B5 and

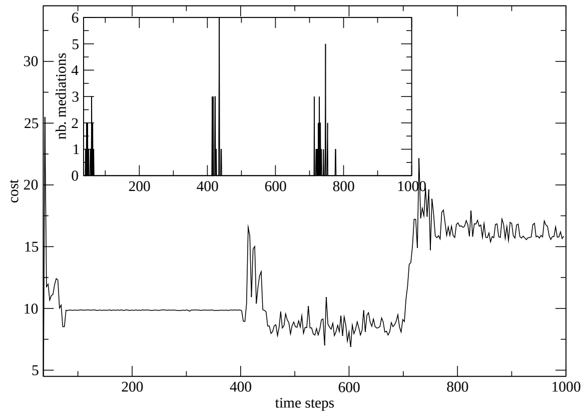

Figure 6: Situation with two changes in the traffic volume (at time steps 400 and 700).

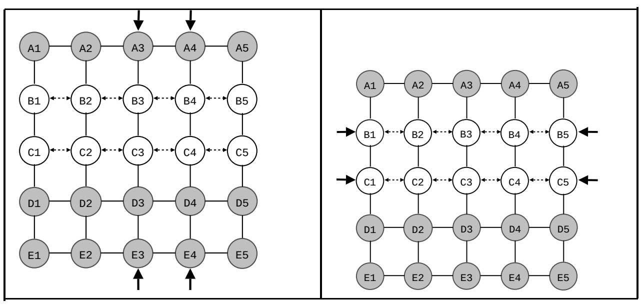

Figure 7: Case with fixed coordination: change in sources for Case 1 (left) and Case 2 (right).

C1 to C5.

Finally, in case IV we have performed a simulation without any mediation, which is shown in Figure 8. During the first 400 time steps the cost is high due to the starting situation (same as all previous cases, but here no agent mediate so that individual costs remain high).

Let us assume that the traffic lights are pre-programmed to run a fixed coordination involving agents B1 to B5 and C1 to C5 (Figure 7 (left)). However the traffic volume increases in an unexpected way at nodes A3, A4, E3, and E4 ($\mu = 25$ and $\sigma = 1$). Note that the costs increase even more because the fixed coordination in direction EW/WE is not appropriate for this particular situation.

Fixed coordinations perform well only in the case they fit the traffic volume. This is also shown in the same figure, after step 700. Now, the volume of traffic is high in nodes B1, C1, B5, and C5 and are in accordance with the fixed coordination (Figure 7 (right)), thus reducing the costs. To show that this reduction is worse than all cases with mediation, Figure 8 depicts two cases: case IV (no mediation, fixed coordination) and case III (mediation, adaptive coordination). We can see that even with a fixed coordination in the streets with higher traffic flow, this situation is not as good as the coordination reached by the agents using our approach: the cost when there is a mediation is always lower than the cost when the coordination is fixed.

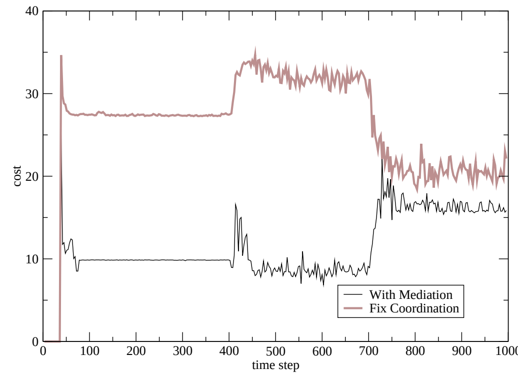

Figure 8: Case with fixed coordination.

5. CONCLUSIONS

Centralized approaches to traffic signal control cannot cope with the increasing complexity of urban traffic networks. A trend towards decentralized control was already pointed out by traffic experts in the 1980's and traffic responsive systems for coordination of traffic lights have been implemented. However, these are not flexible enough since they do not allow changes in the topological design of the coordination. For instance, it is not possible to break a coordinated group so that one intersection can start coordinating in another direction with other neighbors.

The present work is a step in the long term effort of testing different approaches inspired by multiagent systems for coordination of traffic lights. Techniques based on evolutionary game-theory proved successful to the extent that it was possible to maintain the coordination of traffic signals in a decentralized fashion and to permit the emergence of cooperation among individually-motivated agents in dynamic environment under communication constraints. This approach assumes that the expectations of agents concerning their local intentions converge to a given pattern upon receiving a feedback from the environment, this being one possible shortcoming when this feedback is only partially or not available. Thus, as future direction in [1] it was proposed that this could be replace by an efficient form of communication, in which the intentions are somehow included in the negotiation. Therefore the mediation based solution proposed in [6] is tried in the present paper.

Using the OptAPO algorithm, a cooperative mediation is performed by the agent with more information about the subsystem (determined by a relationship graph). Within this subsystem, the mediator proposes changes which minimize the costs. An important issue is that even if the good_list increases, the mediators sub-problem is not likely to hold the complete network. Also, the number of messages will not increase exponentially because, to calculate the costs, agents with different plans are not always in a conflicting situation.

Up to now the OptAPO algorithm was tested in a static environment (graph coloring) so that once a solution is found the process ends. In the traffic scenario, the mediation not only reduces the cost but also can deal with the changing traffic pattern as shown in the experiments. The mediation takes some time especially in the beginning when the agents views are not enlarged. However, already during the mediation, the costs go on decreasing so that the adaptation occurs concurrently with the traffic changes.

As pointed out in [6], in OptAPO there are mechanisms to preserve the identity and/or private values of the agents (increasing the autonomy of them). However, in these cases, the mediator will not gain as much information as it would be desired, affecting the optimality of the algorithm. Autonomy is indeed a central issue in the traffic scenario. When cooperation cannot be taken for granted (after all, agents may have high incentives not to disclose their actual traffic states) it remains to be investigated how this affects the optimality. However, this also motivates us to propose new forms of mediation like for instance giving the most constraint or interested agent the priority to mediate, eventually including costs for the mediation itself.

Other important issues are to study the balance between mediation cost in terms of communication *versus* time necessary for agents to reach the coordination in the game-theoretic approach (which normally takes some time because agents have to learn via reinforcement learning).

Acknowledgments

The first and second authors are partially supported by CAPES and CNPq respectively. We also thank the anonymous reviewers for their valuable suggestions.

6. REFERENCES

[1] A. L. C. Bazzan. Evolution of coordination as a metaphor for learning in multi–agent systems. In G. Weiss, editor, *DAI Meets Machine Learning*, number 1221 in LNAI, pages 117–136. Springer–Verlag, Berlin Heidelberg New York, 1997.

[2] A. L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multiagent Systems*, 10(1):131–164, March 2005.

[3] K. Dresner and P. Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *The Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, July 2004.

[4] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT - a traffic responsive method of coordinating signals. TRRL Lab. Report 1014, Transport and Road Research Laboratory, Berkshire, 1981.

[5] F. Klügl, C. Oechlein, and F. Puppe. Sesam. the multi–agent simulation environment, 2001. <http://www.simsesam.de>.

[6] R. Mailler and V. Lesser. Solving distributed constraint optimization problems using cooperative mediation. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 438–445. IEEE Computer Society, 2004.

[7] D. Oliveira, P. Ferreira, and A. Bazzan. Reducing traffic jams with a swarm-based approach for selection of signal plans. In *Proceedings of Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence - ANTS 2004*, volume 3172 of *LNCS*, pages 416–417, Berlin, Germany, 2004. Springer-Verlag.

[8] TRANSYT-7F. *TRANSYT-7F User's Manual*. Transportation Research Center, University of Florida, 1988.

[9] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *International Conference on Distributed Computing Systems*, pages 614–621, 1992.