Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 104349, 13 pages
http://dx.doi.org/10.1155/2014/104349

*Research Article*

# Multiagent Based Decentralized Traffic Light Control for Large Urban Transportation System

**Yang Xu, Yulin Zhang, and Ming Liu**

*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Yang Xu; xuyang@uestc.edu.cn

Received 2 February 2014; Revised 27 May 2014; Accepted 19 June 2014; Published 15 July 2014

Academic Editor: X. Zhang



Intelligent traffic control is an important issue of the modern transportation system. However, in large-scale urban transportation systems, traditional centralized coordination methods suffer bottlenecks in both communication and computation. Decentralized control is hard if there is very limited observation to the whole network as evidences to support joint traffic coordination decisions. In this paper, we proposed a novel decentralized, multiagent based approach for massive traffic lights coordination to promote the large-scale green transportation. Considering that only the traffic from the adjacent intersections may affect the state of a given intersection one time ahead, the key of our approach is using the observations of a local intersection and its neighbors as evidences to support the traffic light coordination decisions. Therefore, we can model the interactions as decentralized agents coordinating with a decision theoretical model. Within a local intersection, constraint optimizing agents are designed to efficiently search for joint activities of the lights. Since this approach involves only local intersection cooperation, it is well scalable and easily implemented with small communication overhead. In the last section, we present our software design on this approach and based on our simulation, this approach is feasible to a large urban transportation system.

## 1. Introduction

Building intelligent transportation techniques in large urban transportation systems is appealing to reduce fuel consumptions and transportation delays [1]. A key is to enable joint coordination between intersections so that continuous traffic can flow over several intersections in main directions with the least delays [2].

To achieve this, extensive researches have been carried out. Traditional centralized coordination schemas, such as auction [3] and resource optimization [4], although have proven their performance to coordinate a number of intersections, they are infeasible to the large urban transportation system due to their computation and communication bottleneck shown in related literatures [5]. Existing works in decentralized traffic control suffer many limitations due to the nature of partial observability to the whole transportation network as well as its high dynamic traffic patterns. Historical data based approaches, such as reinforcement learning [6] and pattern discovery models [7], rely on the statistic pattern inferred to predict incoming traffics. However, they are incapable of managing dynamic traffic patterns dissimilar to the historical data, such as traffic accident or road maintenance. Some other works use dynamic programming to react to the dynamic real-time traffic according to the sensor readings, but the models are based on some impractical assumptions. For example, Robertson used a decentralized control schema but assumed that the sensor information within the whole area was easily obtained from centralized severs [8]. This may suffer a communication bottleneck in a huge urban area. Shenoda assumed that the coming vehicles follow a Poisson distribution to build a decentralized coordination algorithm [9], which is not always the case, for example, domain of emergency response in disasters. Xie et al. used fixed traffic signal phases to solve the conflicted traffic flows [10], which is not flexible to optimize concurrent traffic flows within the intersections.

In this paper, we proposed a decentralized traffic control approach to enable green transportation in a large urban area.

This approach is built based on a two-level hierarchical agent-based architecture toward robust and flexible coordination. In the top level, decentralized agents are modeled to coordinate with the neighbor intersections. In the bottom level, local agents within an intersection work cooperatively with a constraint optimization model. To enable the intelligent transportation, the key is that agents should make decisions based on the prospection of their local traffic state. Although it is infeasible for these agents to get the global state of the whole network to infer its next local state, we observed that only the traffic in their locally adjacent intersections can affect their coordination decision. Therefore, by closely interacting with their neighbors and sharing their local traffic states, each intersection may be able to gain a complete view of the states necessary to achieve decentralized control. Since this approach requires only the local state of neighbor intersections to cooperate, it is well scalable and easily implemented with small communication overhead.

In our algorithm design, between intersections, we setup a decision theoretical model for decentralized coordination. Since the next-time traffic is solely determined by the current state of the local intersection and its actions, it is a Markov decision process (MDP) and each agent takes the states of adjacent intersections into consideration for their decision model. To solve the uncertainties in state transition functions, we built heuristics either from statistic data or from reinforcement learning so that middle agents are able to jointly choose their best actions to minimize traffic delays. In addition, local agents within an intersection handle the local traffic. The decision process within a local intersection is modeled as a constraint optimization problem (COP) where maximum traffic flow should get through but conflicted traffics should be avoided.

In order to implement our software design, the key is to build each agent and implement the coordination by the interactions of the agents. In the top level, information agent is modeled for each intersection to maintain the local state that it needs to make decisions. In the bottom level, control agents within an intersection work cooperatively to solve the conflicts of different traffic flows. Middle agents are also built to coordinate all these control agents with the COP model. By building these two-level agents, the coordination between intersections is mainly achieved by the information agents, and the conflicts within each intersection are able to be solved by the middle agent and control agents. The system is implemented by RETSINA platform. In addition, it is simulated and the illustrated results proved the feasibility of our approach.

## 2. State of the Art

Many algorithms are designed to optimize the traffic of large urban transportation system in a decentralized manner. In order to achieve decentralized urban traffic control, the most straightforward approach is to generate optimal coordinated plans for fixed-time operation, such as TRANSYT [11]. However, due to the high dynamics in large urban transportation, this approach is hard to be adaptive to its real-time traffic.

Historical data are widely used to generate adaptive algorithms. Pattern discovery models are developed by categorizing the historical traffic into different patterns and assigning an optimal traffic light control plan for each pattern. PCA and SVM methods are applied for feature extraction, training, and classification of network-level traffic patterns [12] so that sensors within the intersection can detect real-time traffic pattern and choose the predefined plan. But the traffic patterns known as a priori may not cover all the patterns in real domains. An alternative is to incrementally build new patterns by assuming that the traffic is relatively infrequently changing [13]. Reinforcement learning is a set of techniques that is always applied in this domain. $Q$-learning is applied to learn the control policy for single intersection from historical data [14]. Multiagent reinforcement learning is applied for all the agents to learn the control concurrently, but this approach is not scalable since the reward function is too large to be enumerated [6]. In addition, all these approaches are incapable of managing dynamic traffic patterns dissimilar to the historical data, such as traffic accident or road maintenance.

In order to effectively respond to real-time traffic, close interactions between intersections are carried out to obtain the network-level traffic information and synchronize all the interactions over the network. Distributed constraint optimization problem (DCOP) is applied to formalize the synchronization of different intersections [15]. However, this approach is based on a centralized mediator to refer for all intersections in a given mediation session, which suffers communication and computation bottleneck when the network scales up. ADOPT and DPOP models are also carried out with either huge communication overload or retarded system response time [5].

Some other approaches relax these interactions and synchronize only within local intersections to obtain a good policy. For example, Phase-by-phase system is developed to optimize traffic for local intersections by predicting the traffic flow merely from their neighbors, but this research is based on the assumption that the vehicles to an intersection follow a Poisson process, which is not always the case in the real domain [9]. Xie et al. model the urban traffic control as a synchronously operating scheduling problem [10]. Each intersection agent estimates its future traffic from its upstream neighbors to obtain a myopic projection if the traffic flow is characterized as a cluster sequence, which is inaccurate in real domain with unpredictable vehicle behaviors.

## 3. Problem Description

A given traffic network can be modeled as an undirected graph $G(V, E)$ shown in Figure 1, where $V$ is the set of intersections and $E$ is the set of roads between intersections. For any two intersections $v_i$ and $v_j$, $\langle v_i, v_j \rangle \in E$ represents that there is a road that vehicles can get through between $v_i$ and $v_j$, and $v_i$ and $v_j$ are neighbors. Specifically, $n(v_i)$ is defined as all the neighbors of the intersection $v_i$, and $|n(v_i)|$ is the number of roads connected to intersection $v_i$. For example,

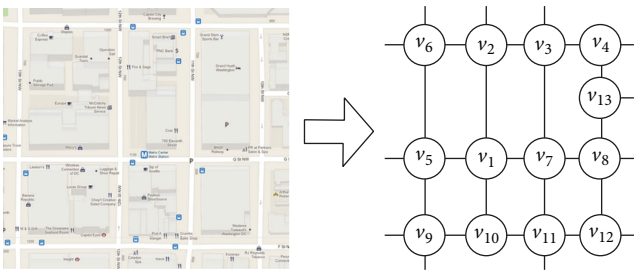



Figure 1: Models of intersections in a given urban area.

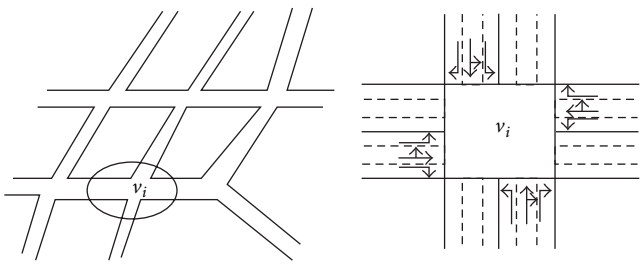



Figure 2: Model of intersections in a transportation network.

in Figure 1, $|n(v_1)| = 4$ and $|n(v_4)| = 3$. For each road $\langle v_i, v_j \rangle$, there are $|n(v_i)| - 1$ turning directions toward $v_i$, and each of them is called a lane. In Figure 1, there are 3 lanes of each road connecting to $v_1$ and 2 lanes each to $v_{13}$. The total number of lanes in $v_i$ can be calculated as $|n(v_i)| \times (|n(v_i)| - 1)$. To be easily referred, the lane from intersection $v_j$ to intersection $v_k$ going through $v_i$ is written as $Ln_i^j(k)$, $v_k \in \{n(v_i) - v_j\}$.

In each intersection, there are a number of vehicles stopping and waiting to go through. We assume that the number of vehicles waiting at lane $Ln_i^j(k)$ at time $t$, written as $Tf_i^j(k,t)$, can be detected by video cameras installed at each intersection. Because it is unable to detect the waiting traffic one time ahead, $Tf_i^j(k,t+1)$ is unobservable.

As shown in Figure 2, in an intersection, each traffic light takes an action to control a specific lane. Specifically, at time $t$ the action to control the vehicles on the lane directed from $v_j$ to $v_k$ at $v_i$ is written as $£_i^j(k,t) \in \{green, red\}$. Please note that *yellow* is not assigned to $£_i^j(k,t)$ because *yellow* is a fixed interim status necessary from *green* to *red* and is not an independent action.

At each horizon, we model the transition function under the action of each traffic light $£_i^j(k,t)$ as the waiting vehicles transferred from $Tf_i^j(k,t)$ to $Tf_i^j(k,t+1)$. It denotes that after each lane takes an action, waiting traffic gets through the intersection and new traffic arrives, and the waiting traffic is transferred to $Tf_i^j(k,t+1)$.

Since green intelligent control is an intention that a series of traffic lights are coordinated to allow continuous traffic flow over several intersections in main directions [1], its control should allow more vehicles to go through the intersections with the least delay. To maximize the moving vehicles, the sum of vehicles in the waiting queues should be minimized. Therefore, the key is to minimize waiting vehicles in the next time step other than the myopic optimization in the current time. We define the utility function toward the intelligent control for the global transportation network as

$$\mathrm{EU}(G,t) = \sum_{v_i \in V} \sum_{v_j \in n(v_i)} \sum_{v_k \in \{n(v_i) - v_j\}} Tf_i^j(k,t+1). \quad (1)$$

In this formula, the expected utilities of the cooperative transportation network $G$ at time $t$ (defined as $\mathrm{EU}(G,t)$) are the sum of the waiting vehicles in front of intersections at time step $t+1$.

Based on the expected utility function, the goal of cooperative traffic light control over the network $G$ is to find an optimal joint policy $\pi^*$ for traffic lights coordinating all the intersections so that the expected utility could be minimized. Consider

$$\pi^*(t) = \underset{Jt\_act(G,t)}{\operatorname{argmin}} \mathrm{EU}(G,t), \quad (2)$$

where $Jt\_act(G,t)$ in the transportation network $G$ at time $t$ consists of all the independent activities of each lane over every intersection at that time. It can be written as

$$Jt_{act(G,t)} = \bigcup_{v_i \in V} \bigcup_{v_j \in n(v_i)} \bigcup_{v_k \in \{n(v_i) - v_j\}} £_i^j(k,t). \quad (3)$$

Since $Tf_i^j(k,t+1)$ is not observable, finding the optimal policy $\pi^*$ is intrinsically unsolvable in a large-scale transportation system [16].

## 4. Multiagent Based Traffic Control System

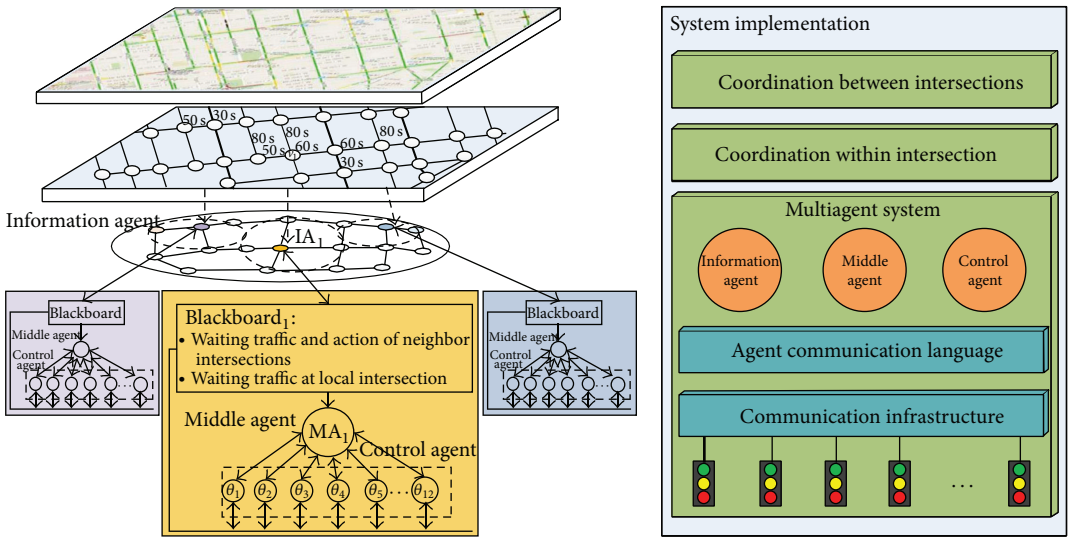
To solve the problem above, the key is to make the decisions to carry out the activities based on the prospection of local intersection state one time ahead. Although neither observation of the local state one time ahead nor accessing the global states of whole transportation network for inference is feasible, the estimation to the next local state of each intersection can still be made according to the following key characters in physical transportation networks.

(i) The state of the network at time step $t+1$ is solely determined by its state at time step $t$.

(ii) For any intersection $v_i$, its traffic flow is solely determined by the traffic flow arriving from its adjacent intersections.

Therefore, only the traffic in local intersection and the neighbors can affect its next coordination decision. If close interaction between adjacent intersections can be made to share their local traffic states and inform neighbors in advance of the actions that they are going to carry out, we can build a decentralized traffic coordination to solve this urban traffic optimization problem in a myopic view.

*4.1. Decentralized Multiagent System Design.* According to this, our multiagent system design is illustrated as Figure 3(a). Based on the analysis in Section 3, the urban traffic network




(a) Multiagent traffic light coordination architecture design (b) System implementation

Figure 3: Decentralized multiagent based traffic light coordination architecture and implementation.

is modeled as a graph. In the graph, each intersection is represented as a node and the road between them has been abstracted as a link. In the next level, since only the traffic of the local intersection and its adjacent intersections can affect its next coordination decision, each intersection has to share their states with their neighbors. For each intersection, we setup an information agent to share the states of the traffic of their lanes with the neighbors as well as receiving and publishing the adjacent intersection states to be used for the intersection's local decisions.

In the bottom level, control agents are responsible to solve their joint optimized decisions within the intersection. To achieve this, the multiagent system consists of three parts. As the first part, a blackboard is used to gather the information provided by its information agent, which are (1) current states of waiting traffic and traffic control actions from each adjacent intersection and (2) the states of waiting traffic of its own intersection. As the second part, each lane within the intersection is represented by a control agent to make decisions for its own lane and negotiate with the other control agent to solve the conflicts of different traffic flows. As the third part, a middle agent is built to coordinate local control agents to reach optimized joint decisions.

The system implementation of our approach is illustrated by Figure 3(b). We use the RETSINA platform to implement the multiagent system with the three basic agents as well as their agent communication language. In addition, we build algorithms for the agents to achieve green intelligent traffic coordination. Between intersections, since an intersection's decision is only determined by the states of its local neighbors, we model the decision as a Markov decision process. Within each intersection, the decision process of the control agents is modeled as a constraint optimization problem (COP). In the rest of this section, we will introduce in detail the agent and algorithm designs.

*4.2. Information Agent.* In order to coordinate with neighbor intersections and gain the information required for decision, information agent $IA_i$ is built for intersection $v_i$ and we have $IA = \{IA_1, IA_2, \ldots IA_i, \ldots\}$, $|IA| = |V|$. Inherited from the transportation network, the logical network of the information agents follows the same connection of $G$ and agent $IA_i$ has a set of neighbor information agents $n(IA_i) = \{IA_j \mid v_j \in n(v_i)\}$, which represents the information agents of $v_i$'s adjacent intersections.

At time $t$, agent $IA_i$ is able to gain the local state $Tf(v_i, t)$ of intersection $v_i$:

$$Tf(v_i, t) = \bigcup_{v_j \in n(v_i)} \bigcup_{v_k \in \{n(v_i) - v_j\}} Tf_i^j(k, t). \tag{4}$$

To make a rational decision toward decentralized traffic optimization, the intersection should know the traffic flows released from neighbor intersections before they arrive. In this case, the agent $IA_i$ has to gain a complete view of the local state of $v_i$ and the actions of their neighbors' incoming action in advance. Hence the state of $v_i$ is defined as follows:

$$S_{v_i}(t) = Tf(v_i, t) \bigcup_{v_j \in n(v_i)} Tf(v_j, t) \bigcup_{v_j \in n(v_i)} \pounds_j(t), \tag{5}$$

where $\pounds_j(t)$ is the joint action of all the traffic lights, $\pounds_j(t) = \bigcup_{v_l \in n(v_j)} \bigcup_{v_m \in \{n(v_j) - v_l\}} \pounds_j^l(m, t)$, and $\pounds_j^l(m, t)$ is the action for the traffic light controlling lane $Ln_j^l(m)$ at time $t$. As Figure 4 shows, the local state obtained by the information agent is published on the blackboard for the agents within the intersection.

In formula (5), the state of intersection $v_i$ is composed of the waiting traffic at the local intersection, those of its adjacent intersections as well as their actions to be carried

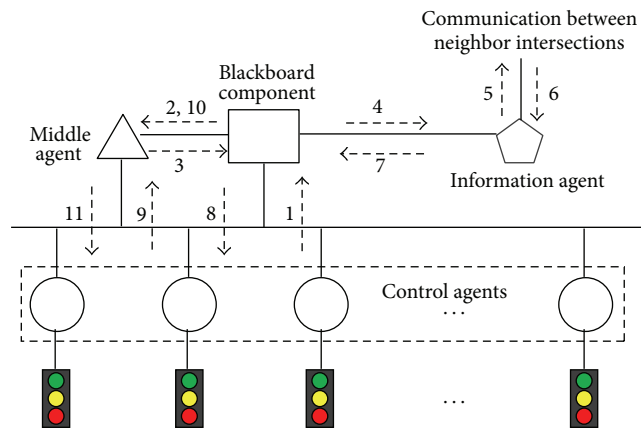
out at $t$. Since the actions of the neighbors to be carried out at $t$ should be shared in advance to gain the complete view of $S_{v_i}(t)$, $IA_i$ as a neighbor of the other agents should have to share its action in advance as well. To break this deadlock, a practical protocol has been designed to estimate its action with $£_i^+(t)$. The details of the decentralized coordination model of $v_i$ will be presented in Section 5.1. In addition, another key of this model is how to infer $v_i$'s next-time local state $Tf(v_i, t+1)$ from its current local state, which will be addressed in Section 5.3.

*4.3. Middle Agent and Control Agent.* Within each intersection, there are conflicts between different traffic lights. Although it is possible for $v_i$ to make a decision for all the traffic lights within the intersection based on the fixed conflict-free traffic signal phases, it is not flexible to deal with unpredictable traffic patterns. Therefore, we build a middle agent and a set of control agents for each intersection. As Figure 4 illustrates, a control agent $\theta_i^j(k)$ is built for each traffic light to monitor and control traffic on lane $Ln_i^j(k)$. Before each round of decision, it monitors the waiting traffic on the lane and publishes it on the blackboard. After the local state is obtained by the information agent and published on the blackboard, all the control agents in the intersection work together to eliminate the conflict actions and provide local optimized activities $£oc_i(t)$ for the middle agent. The middle agent $\mathrm{MA}_i$ for intersection $v_i$ is responsible to initialize and coordinate all these control agents within the intersection and also choose the best joint action from the local optimized activities $£oc_i(t)$.

*4.4. Interactions between Agents.* With the two-level multiagent system, their decentralized coordination is able to be achieved through the information processing process shown in Figure 4. Firstly, the control agent monitors the waiting traffic at the lane and publishes it on the blackboard. Then the middle agent is able to estimate its current action $£_i^+(t)$. In order to gain the local state, the information agent shares its waiting traffic and expected action with neighbors. After the local state is obtained and published on the blackboard, the control agents are able to cooperatively generate the conflict-free local optimized joint actions $£oc_i(t)$ and the middle agent can choose the best action from $£oc_i(t)$.

## 5. Intelligent Traffic Control Algorithm

Based on our multiagent system design, we build algorithms for these agents to generate the optimal action $£oc_i^*(t)$ according to state $S_{v_i}(t)$ and solve the constraints within the intersection. An overview of our traffic control algorithm is illustrated in Figure 5. For each intersection, before making the decision, it has to share its state of local waiting traffic and expected actions. Since the intersection's next time state can be maintained with its information agent, a Markov decision process can be built for the decentralized traffic control as a conflict-free localized optimal action set. Secondly, we model this coordination within the intersection as a constraint optimization problem (COP). In order to solve these models,




(1) Current waiting traffic at each lane $Tf_i^j(k,t)$
(2) Current waiting traffic at the intersection $Tf(v_i,t)$
(3) Estimated action $£_i^+(t)$
(4, 5) Waiting traffic and estimated action $\langle Tf(v_i,t), £_j^+(t)\rangle$
(6) Information $\langle Tf(v_j,t), £_j^+(t)\rangle$ from neighbor $v_j$
(7, 8, 10) Local state $S_{v_i}(t)$
(9) Local optimized action $£oc_i(t)$
(11) Optimal action $£oc_i^*(t)$

Figure 4: Information processing between agents.

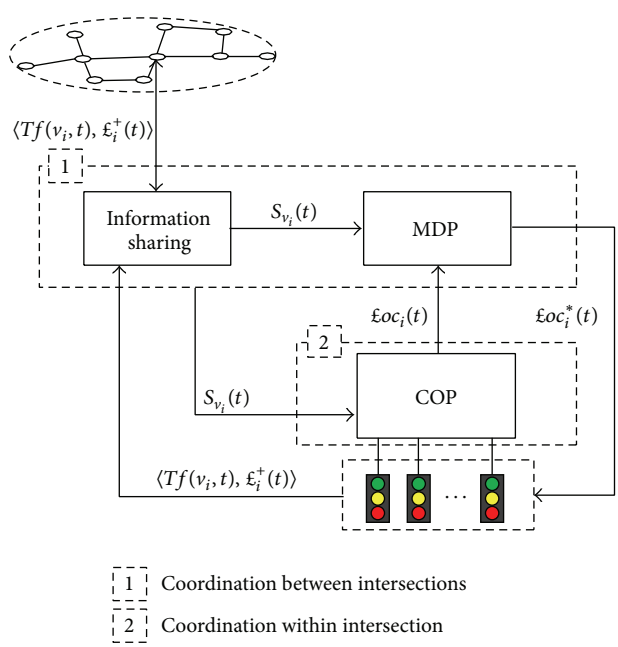



Figure 5: Coordination model of local intersection.

the key is to figure out the transition from the current local state $Tf_i^j(k,t)$ to its next local traffic state $Tf_i^j(k,t+1)$ one time ahead. Finally, we propose a heuristic algorithm for the local state transition function.

*5.1. Coordination between Intersections.* In this section, we present the details on how agents in different intersections

coordinate with each other, and what information is shared, so as to make their decentralized actions toward optimized decentralized traffic control. The key is that agents build their own local states to infer the states of their local intersections one time ahead. For a specific middle agent $\mathrm{MA}_i$, its decentralized control process is built as a decision theoretical model $\langle S_{v_i}, £oc_i, T, Tf(v_i)\rangle$.

(i) State: $S_{v_i}(t)$ is the intersection $v_i$'s local state at $t$. It is built from the information received from their neighbors as well as the observation to the local traffic state.

(ii) Action: as explained in Section 3, agent $\mathrm{MA}_i$ is only able to choose the action from the available joint activities set $£oc_i(t)$ worked out by its local agents $\Theta_i$.

(iii) Transition function $T : S \rightarrow Tf(v_i)$ defines the transition from state $S_{v_i}(t)$ to the local traffic state of $v_i$ at $t+1$ by the action.

(iv) Utility function is defined as $Tf(v_i, t+1)$.

$\mathrm{MA}_i$ is to find its optimal policy $\pi^{\Delta}$:

$$\pi^{\Delta} = \underset{£oc_i(t)}{\operatorname{argmin}} Tf\left(v_i, t+1\right). \tag{6}$$

A key challenge of this model is that information agent $IA_i$ has to share its current joint action of all the traffic lights in the interaction with its neighbors one time ahead. And these joint actions of the neighbors are critical to build $S_{v_i}(t)$, which in turn influences their own decision. Therefore, the agent has to make its decision based on sharing its decision result one time ahead and this looping process produces a deadlock.

To break this deadlock, from formula (5), we observe that a heuristic protocol can be designed because adjacent intersections' joint actions only contribute a small portion to the local state. Therefore, an estimated joint action $£_i^+(t)$ to $£_i(t)$ can be defined to solve this deadlock, and formula (5) can be represented as

$$S_{v_i}(t) = Tf\left(v_i, t\right) \bigcup_{v_j \in n(v_i)} Tf\left(v_i, t\right) \bigcup_{v_j \in n(v_i)} £_j^+(t). \tag{7}$$

$£_i^+(t)$ can be estimated in many ways; a practical way is to estimate $£_i^+(t)$ one time ahead according to the local state $Tf(v_i, t)$. Therefore, we design a practical algorithm as

$$Tf\left(v_i, t\right) \longrightarrow £_i^+(t), \tag{8}$$

where the action is determined by the number of waiting traffics of each lane in the intersection. Although it may be imprecise to estimate $£_i^+(t)$ based on historical states, considering that the traffic moves continuously, $£_i^+(t)$ cannot be significantly varied from $£_i(t)$.

*5.2. Coordination within Intersection.* In this section, we present how the control agents $\Theta_i$ in an intersection $v_i$ are coordinated by middle agent $\mathrm{MA}_i$ to build the local conflict-free joint activities set $£oc_i(t)$. In order to solve the constraints

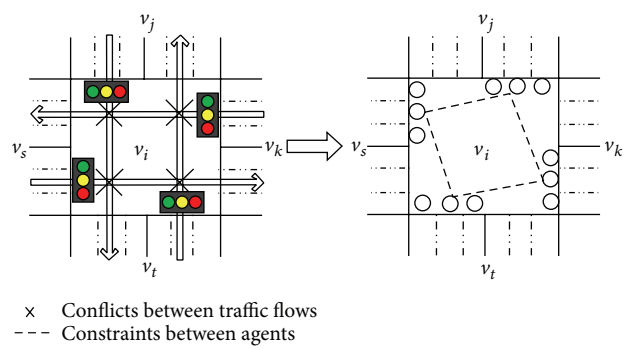



Figure 6: Traffic conflictions in a four-way intersection.

within the intersection, the agents have two policies: fixed policy and dynamic policy. In the fixed policy, $£oc_i(t)$ contains all the conflict-free joint actions, which are predefined. In the dynamic policy, the middle agent gives an order to the control agents and each agent proposes its preferred action according to the order.

In this paper, we focus on the dynamic policy design, and the middle agent has to work out a set of orders for the control agents as their social conventions and start the negotiation from one control agent at each round to get a local joint action. There are also two ways to generate each of the order: random ordering and heavy traffic lane first. Random ordering initializes the order of control agents randomly. On the other way, the control agent with more waiting traffic is given a higher priority.

Following a given order, all the control agents have to coordinate to work out the local optimized conflict-free joint policy. The coordination process of all the agents in $\Theta_i$ is built as a constraint optimization model $\langle £_i, \{green, red\}, C_i\rangle$.

(i) $£_i$ defines the variables of COP, where each variable is the traffic light action of $£_i^j(k,t)$ worked out by $\theta_i^j(k)$.

(ii) Each $£_i^j(k,t)$ is only chosen from a binary set $\{green, red\}$.

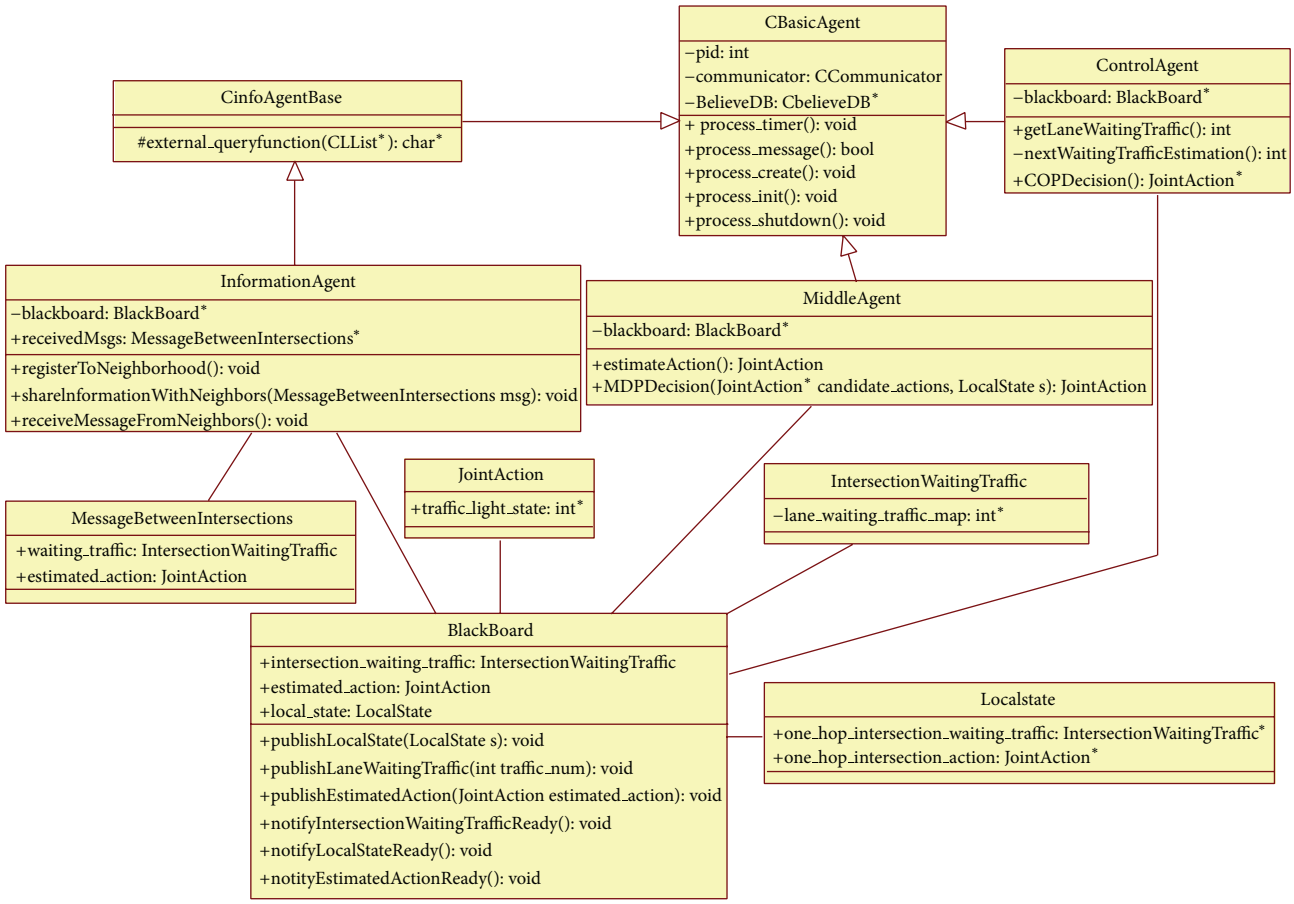
(iii) $C_i$ is the binary constraint set predefined for $v_i$ by domain. For example, typical traffic conflictions in a four-way intersection can be illustrated in Figure 6, where any variables connected with a dash line cannot be *green* at the same time.

The utility function for each assignment of variable $\theta_i^j(k)$ can be formulated to help agent $\theta_i^j(k)$ to locally minimize the waiting traffic of its lane:

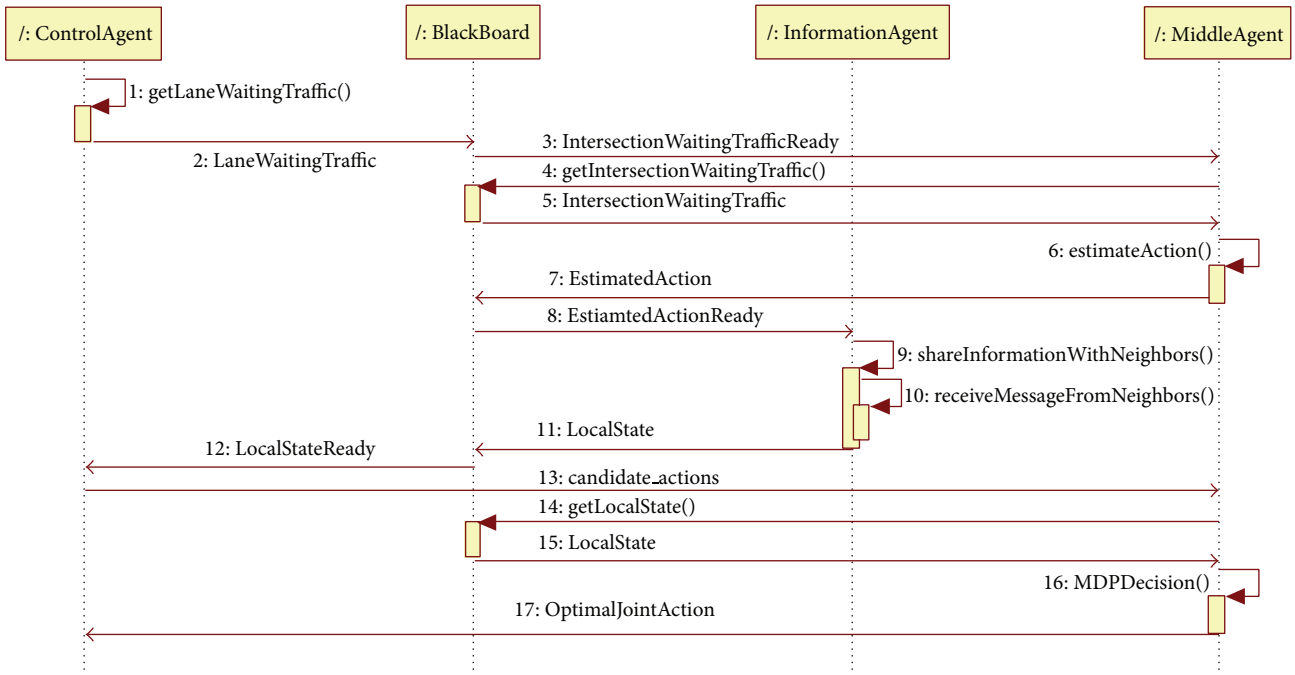
$$\mathrm{EU}\left(Tf_i^j(k), £_i^j(k,t)\right) = Tf_i^j(k, t+1). \tag{9}$$

For each agent $\theta_i^j(k)$, its optimization policy is

$$\pi^{\Delta\Delta} = \underset{£_i^j(k,t)}{\operatorname{argmin}} Tf_i^j(k, t+1). \tag{10}$$




(a) Class diagram




(b) Sequence diagram

Figure 7: Software engineering illustration of the multiagent system design.

(1) $order_i = M_i.generateOrder(\Theta_i)$;
(2) $M_i.SetupSearches(£oc_i)$;
(3) **for all** $£oc_i[m] \in £oc_i$ **do**
(4) $\quad order_i.starter = RandomChoose(\Theta_i)$;
(5) $\quad$ **for all** $\theta_i^j(k) \leftarrow order_i.next()$ **do**
(6) $\quad\quad £oc_i^j(k,t) \leftarrow Lottery(\mathrm{Tf}_i^j(k,t+1), \mathrm{Tf}(v_k,t))$;
(7) $\quad$ **end for**;
(8) **end for**;
(9) **return** $£oc_i$;

Algorithm 1: Decision process of $\Theta_i$ within intersection $v_i$.

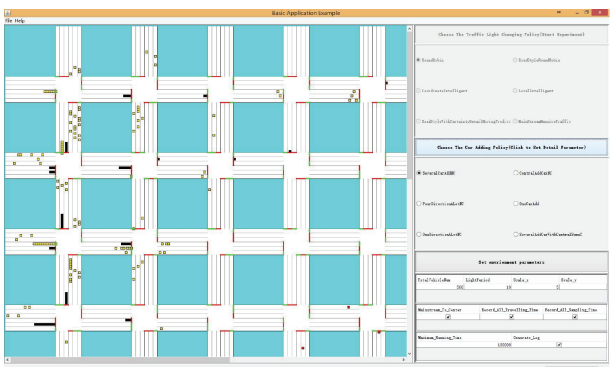


Figure 8: The screen shot of the urban traffic simulator.

According to this policy, we build a myopic heuristic mapping function as

$$£_i^j(k,t) \times Tf_i^j(k+1) \times \sum_{v_l \in n(v_k)-v_i} Tf_k^i(l,t) \longrightarrow [0,1]. \quad (11)$$

In this equation, if $\theta_i^j(k)$ is more likely to help to reduce $Tf_i^j(k+1)$, the agent $\theta_i^j(k)$ is more likely to set $£_i^j(k,t)$ as *green*. The pending downstream traffic $Tf_k^i(l,t)$ to the adjacent intersections should also be considered. Because if there is heavy traffic in the next intersection to go through, discharging more traffic there is not helpful and the lane is less likely to be set as *green*. In addition, to avoid any lane with very little waiting traffic being set as continuous *red*, we define that the continuous red phrase should not exceed $R_{\max}$.

The decision process of local agents $\Theta_i$ is described in Algorithm 1. In this algorithm, $\mathrm{MA}_i$ initiates an order for the control agents in $\Theta_i$ (line 1) and sets up a set of searches for $£oc_i$ (line 2). For each search $£oc_i[m]$, it randomly starts from an agent (line 4), and each agent $\theta_i^j(k)$ sequentially chooses its action $£oc_i^j(k,t+1)$ according to formula (11) (line 6). Since agents' decisions are based on the probabilistic model, the optimal joint activities are not guaranteed in the searches. Therefore, we set the size of $£oc_i$ bigger than the number of lanes in $v_i$ to increase the chance of optimization. However, it also increases the computation complexity for $\Theta_i$.

*5.3. Heuristic Transition Function.* Although solving the scalable MDP for massive traffic lights control is mathematically feasible, the uncertainties on the state transition function resulted by unpredictable traffic in heavy traffic network will make the computation hard. There are three key factors:

(i) the unpredictable amount of traffic going through under a given green light;

(ii) the uncertainty of line choosing on the adjacent intersection when vehicles passed through a given intersection;

(iii) the unpredictable arriving time of given traffic arrived at the next intersection which depends on the congestion and road conditions as well as their distances between intersections.

These factors may vary significantly under different traffic conditions. For simplicity and clarity of our model, we make the following two assumptions. Firstly, we assume that the traffic flow getting through the intersection follows the exponential queue discharge flow rate model [17]. In this case, during a green-light cycle $\Delta_i$, the maximum number of vehicles getting through an intersection is denoted as $h(\Delta_i)$. Therefore, at each time $t$, the number of vehicles getting through lane $Ln_i^j(k)$ is denoted as $Td_i^j(k,t)$, which could be estimated as If $£_i^j(k,t) = green$, during the green-light cycle $\Delta_i$

$$Td_i^j(k,t) = \begin{cases} Tf_i^j(k,t), & \text{if } Tf_i^j(k,t) < h(\Delta_i) \\ h(\Delta_i), & \text{otherwise,} \end{cases} \quad (12)$$

otherwise, $£_i^j(k,t) = red$ and $Td_i^j(k,t) = 0$.

Secondly, the probability for each vehicle to choose the lane could be estimated from historical statistics. In this paper, we assume that vehicles will evenly choose the lanes after it gets through an intersection. Thus, the probability $P_i^j(k)$ is

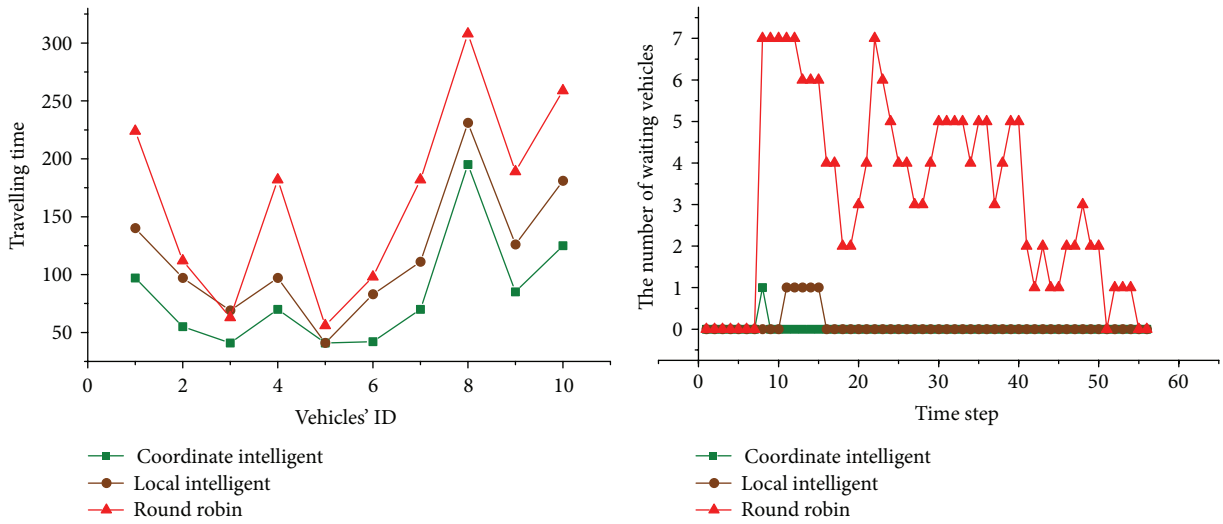
$$\forall v_k \in n(v_i), v_j \in n(v_i) - v_k, \quad P_i^j(k) = \frac{1}{|n(v_i)| - 1}. \quad (13)$$

With the assumptions above, we can establish the transition from $Tf_i^j(k,t)$ to $Tf_i^j(k,t+1)$. Observing that the number of vehicles on a given lane is determined by vehicles' choice of lanes and the number of vehicles released by adjacent intersections in the last cycle, we will have

$$P_i^j(k) \times \sum_{v_l \in n(v_j)-v_i} Td_j^l(i,t). \quad (14)$$

According to the real-time traffic condition, not all vehicles released from its adjacent intersections can arrive at the intersection at the end of the green phase. The number of vehicles arriving at intersection $v_i$ on lane $Ln_i^j(k)$ is denoted as $Tc_i^j(k,t)$. If the weight of next road is $w_{i,j}$, a function to predict the arriving ones within the traffic light period $\Delta_i$ can be proposed as

$$Tc_i^j(k,t) = g\left(P_i^j(k) \times \sum_{v_l \in n(v_j)-v_i} Td_j^l(i,t), w_{i,j}\right). \quad (15)$$




(a) The travelling time of all the vehicles

(b) The number of waiting vehicles at each time

Figure 9: Results of green-wave effect.

(1) **for** $\mathrm{Ln}_i^j(k) \in v_i$ **do**
(2) **if** $\mathrm{Tf}_i^j(k,t) < h(\Delta_i)$ **then**
(3) $\mathrm{Td}_i^j(k,t) = \mathrm{Tf}_i^j(k,t)$;
(4) **else**
(5) $\mathrm{Td}_i^j(k,t) = h(\Delta_i)$;
(6) **end if**
(7) $P_i^j(k) = \dfrac{1}{|n(v_i)| - 1}$
(8) *calculate* $\mathrm{Tc}_i^j(k,t)$
(9) *calculate* $\mathrm{Tf}_i^j(k,t+1)$;
(10) **end for**
(11) **return** $\mathrm{Tf}(v_i, t+1)$;

Algorithm 2: Intersection $v_i$'s transition process from current state to the local state one time ahead.

With the number of vehicles to be released and the ones to arrive at the intersection, the transition function to update the waiting vehicles at next time $Tf_i^j(k,t+1)$ can be computed as

$$Tf_i^j(k,t+1) = Tf_i^j(k,t) + Tc_i^j(k,t) - Td_i^j(k,t). \quad (16)$$

Algorithm 2 presents the process of state transition of intersection $v_i$ from state $t$ to the local state one time ahead $t+1$. For each lane $Ln_i^j(k)$, it firstly estimates the number of vehicles that can get through this intersection when the traffic light is set *green* according to formula (12) (line 2–6). Next, it calculates the probability of each line that the vehicles will choose (line 7). After the number of vehicles released from adjacent intersections and their lane choosing probability are figured out, the number of vehicles arriving at this lane at next time is able to be calculated according to formula (15) (line 8). Finally, according to the transition function (16), $Tf_i^j(k,t+1)$ could be solved (line 9). When all lanes' one time ahead states are estimated, the local state at $t+1$ is worked out (line 11).

## 6. Software Design

In this section, we present the multiagent system design as well as the information processing process. Our two-level agents are built based on RETSINA [18] for its advantage of the programming platform and multiagent coordination mechanism. RETSINA is developed by Robotics Institute of Carnegie Mellon University. It implements all the basic types of agents and the agent communication language (ACL) as well as the agent management service. In addition, RETSINA also provides a peer to peer interaction mechanism for the multiagent systems in a distributed infrastructure. It is implemented with C and C++, which could be easily encoded in embedded traffic control devices. In our multiagent system design, there are four key components.

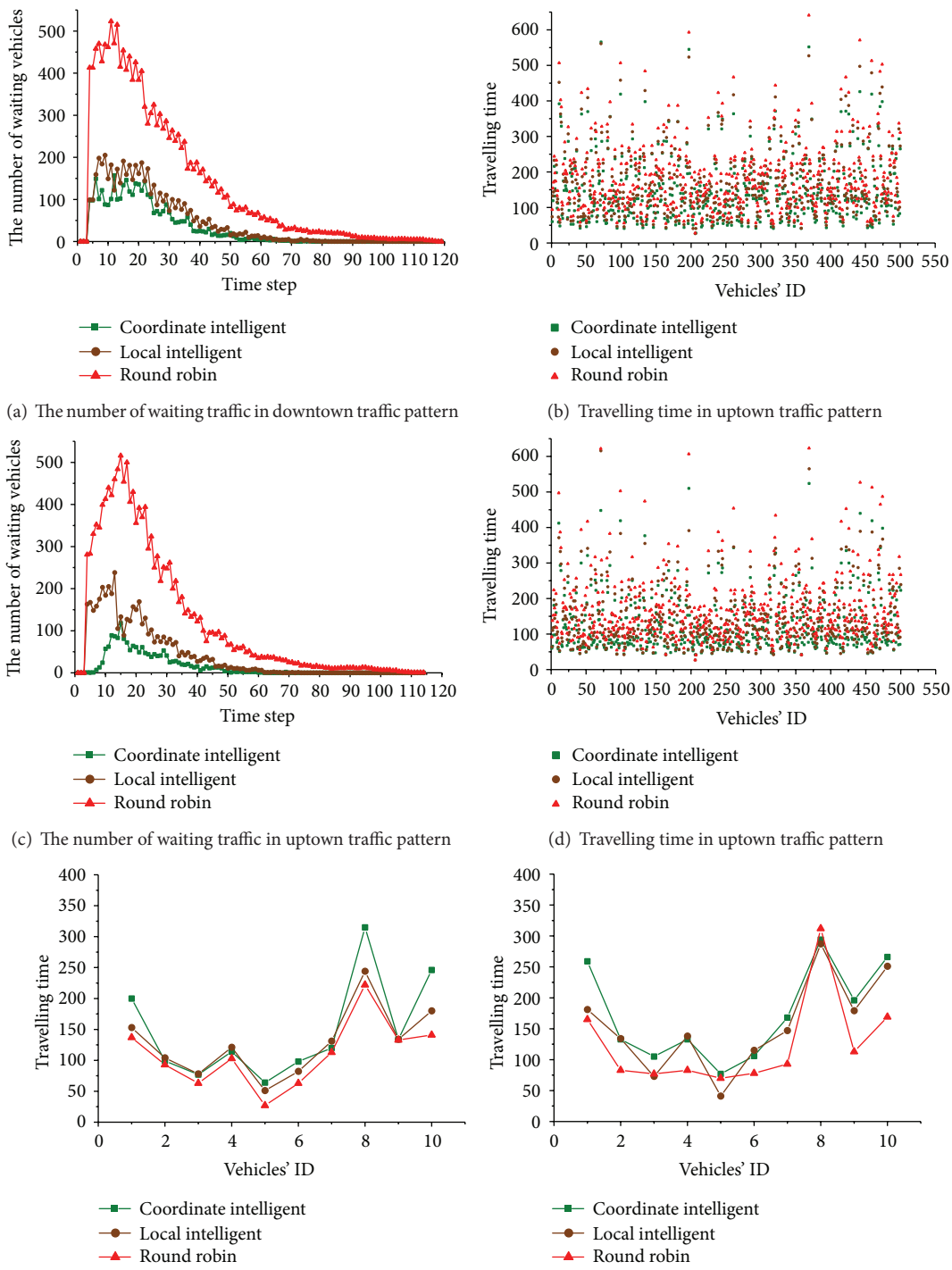
Information agent responds to share the local states with the other information agents of the adjacent intersections. It is implemented based on the RETSINA information agent, which carries out the specific task to communicate with other information agents by using ACL.

Control agent is customized to make decisions for each lane that it represents. It is based on the RETSINA task agent to carry out the information process described in Algorithm 2.

Middle agent is also built based on RETSINA basic task agent. It responds to generate the order for the control agents within the intersection and choose the best policy to achieve joint decentralized control.

Blackboard is a single instance for each intersection in the multiagent system. It is used to provide information publish service for all agents within the intersection.

The multiagent traffic control system designed is illustrated in Figure 7. In Figure 7(a), information agent inherits




(a) The number of waiting traffic in downtown traffic pattern

(b) Travelling time in uptown traffic pattern

(c) The number of waiting traffic in uptown traffic pattern

(d) Travelling time in uptown traffic pattern

(e) Travelling time of vehicles going against mainstream downtown traffic pattern

(f) Travelling time of vehicles going against mainstream in uptown traffic pattern

Figure 10: Results in two typical traffic patterns.

from RETSINA information agent. Both middle agent and control agent inherit from RETSINA task agent. All these agents have access to the blackboard component and the methods related to the cooperative decision. The interaction process in Figure 4 is presented as the sequence diagram illustrated in Figure 7(b). All the agents have their own threads and lifetime, and they interact with each other asynchronously. With the help of blackboard component, all these agents are able to publish and get the information critical to joint intelligent traffic coordination.

## 7. Simulation and Results

In order to manifest the feasibility of our approach, we build an abstract traffic simulator for evaluation. The screen shot of the urban traffic simulator is shown in Figure 8. In this simulator, we build a grid network with a number of intersections, where each intersection connects four adjacent intersections. Vehicles are simulated to go through the network but they have to wait in front of the intersections until the light of their lanes are green.

In our simulation, we load different control schemas to control the traffic so that we can compare the performance with our design. In our experiment, the route of each vehicle is randomly produced and we choose three different schemas to control the traffic: our design labeled as *coordinate intelligent*, local real-time traffic control policy labeled as *local intelligent,* and the traditional round-robin policy labeled as *round robin*. The local real-time traffic control policy (*local intelligent*) is based on the literature [15], whose objective is to release maximum waiting vehicles merely based on the local state of the interaction. We hypothesized that without the prospection one time ahead to build the intelligent control, *local intelligent* should perform worse than our approach. *Round robin* control policy assigns uniform time slices for all the conflict-free traffic signal phases, which are initialized in deployment. It schedules all these conflict-free traffic signal phases in periodic sequential order [19]. Since *round robin* schedules the traffic in a fixed control manner, ignoring the real-time traffic demand, it should perform the worst in unbalanced traffic. The results are compared with two criterions:

(i) the average traveling time for each vehicle to go through the transportation network;

(ii) the average number of waiting vehicles in front of intersections.

*7.1. Traffic Control with Green-Wave Effect.* When the traffic is sparse, the green-waved effect is the most straightforward way to test intelligent traffic control performance [20]. Therefore, we initialized a $5 \times 5$ grid transportation network and 10 vehicles are randomly generated from the margin of this network to get through under the three control schemas. The experimental results are shown in Figure 9. When there are only few vehicles in the network, in *Round Robin* schema, the vehicle not catching up green light has to wait although no vehicle goes the "green" lane. Because of one time ahead intelligent control, *coordinate intelligent* is easier to create the green wave effect that allows the coming vehicles to get through the intersection without stops. Therefore, both the waiting time and the number of waiting vehicles in the *coordinate intelligent* schema are the least in both Figures 9(a) and 9(b).

*7.2. Traffic Control in Different Traffic Patterns.* In urban transportation system, there are two typical traffic patterns [21]. The downtown rush hour emerges in the morning when a lot of vehicles are driven towards the town for work, while the uptown rush hour emerges in the afternoon when massive vehicles go the other way.

We test our approach under these two typical traffic patterns. In this experiment, we initialized a $5 \times 5$ grid network. In the downtown traffic pattern, we simulated 500 vehicles, which are evenly generated from the margin of the network, driving toward downtown within the initial 10 time steps, and as shown in Figure 10(a), the number of waiting vehicles reaches a peak between time steps 10 and 20 as they overload the intersections in the grid. Each traffic control schema is applied to route traffic to their destination and decrease the waiting traffics. Traditional *round robin* way, which only works well in handling balanced traffic, has a poor performance in responding to the unbalanced traffic. As expected, with the one time ahead intelligent control, our approach performs best to have the mainstream traffic flows get through the intersections quickly. Both the average traveling time of those vehicles and the number of waiting vehicles in this schema stay the least as shown in Figure 10(b). In the next section, we also simulated 500 vehicles in an uptown traffic pattern, evenly generated from downtown, driving to spread out of the network. Similar to Figures 10(a) and 10(b), our schema works best in Figures 10(c) and 10(d).

Since the intelligent traffic control should always try to have the mainstream traffic flows go through the intersections with high priorities, we test whether this is the case in our design. In the two traffic patterns, we put 10 vehicles to be driven against the mainstream. Figure 10(e) shows the average traveling time of the 10 vehicles to go uptown while there are 500 vehicles going downtown. Figure 10(f) shows the result of the 10 vehicles to go downtown while the 500 vehicles are going uptown. As expected, to evacuate the heavy traffic, both *local intelligent* and *coordinate intelligent* schemas have to give higher priorities to the mainstream and sacrifice the minority from the other direction. Therefore, as Figures 10(e) and 10(f) indicate, it costs more traveling time for the 10 vehicles to get through the network. Besides, the results also show that the evacuation ability of *coordinate intelligent* is higher than *local intelligent*.

*7.3. Traffic Control in Different Network Scales.* In order to test the scalability of our intelligent control, we perform experiments in different scales of transportation networks. We initialized 500 vehicles in the network with two traffic patterns described in Section 7.2. As shown in Table 1, when the scale of the network increases, it takes the vehicles more time to travel through this network. Due to the intelligent traffic control of our approach, it outperforms the other schemas.

*7.4. Traffic Control in Different Traffics.* In this section, we test our approach with different number of traffics. We initialized a $5 \times 5$ grid network with 500 to 2500 vehicles to go through. As shown in Table 2, in two typical traffic patterns, heavy traffic is more likely to cause congestions with longer average traveling time for each vehicle. However, our approach performed best.

Table 1: The average traveling time of different traffic controls in different network scales.

| | Scale | Coordinate intelligent | Local intelligent | Round robin |
|---|---|---|---|---|
| Downtown traffic pattern | $5 \times 5$ | 150.3 | 155.7 | 177.1 |
| | $6 \times 6$ | 165.5 | 182.1 | 210.1 |
| | $7 \times 7$ | 183.9 | 195.0 | 237.9 |
| | $8 \times 8$ | 216.5 | 219.0 | 255.0 |
| | $9 \times 9$ | 219.5 | 233.4 | 295.1 |
| | $10 \times 10$ | 241.8 | 252.8 | 327.0 |
| Uptown traffic pattern | $5 \times 5$ | 121.7 | 146.9 | 166.6 |
| | $6 \times 6$ | 177.0 | 206.4 | 236.0 |
| | $7 \times 7$ | 195.5 | 237.7 | 269.6 |
| | $8 \times 8$ | 257.7 | 298.9 | 336.7 |
| | $9 \times 9$ | 288.1 | 334.2 | 394.2 |
| | $10 \times 10$ | 359.6 | 401.2 | 478.2 |

Table 2: The average traveling time of different traffic controls with different number of vehicles.

| | Vehicle number | Coordinate intelligent | Local intelligent | Round robin |
|---|---|---|---|---|
| Downtown traffic pattern | 500 | 149.7 | 160.3 | 175.7 |
| | 1000 | 179.7 | 186.6 | 208.3 |
| | 1500 | 222.6 | 230.2 | 240 |
| | 2000 | 243.1 | 257.1 | 270.1 |
| | 2500 | 241.4 | 281.1 | 293.9 |
| Uptown traffic pattern | 500 | 128.7 | 142.5 | 167.3 |
| | 1000 | 138.8 | 181.9 | 221.2 |
| | 1500 | 164.1 | 219.9 | 277.9 |
| | 2000 | 180.4 | 240.9 | 328.7 |
| | 2500 | 201.8 | 280.9 | 382.6 |

## 8. Conclusion and Future Work

In this paper, we presented a multiagent based decentralized traffic light coordination approach for large urban transportation system. In order to improve the control efficiency, we use the prospection of local state one time ahead to make rational decision and build a two-level multiagent architecture and intelligent traffic control algorithms to coordinate these agents. Experiments manifest that our approach is feasible and scalable to improve the decentralized traffic control efficiency.

Although we are capable of dealing with some of the challenges, we leave many of the others in the future. Firstly, in our model we primarily considered video cameras as input sensors; however, more sensors should be considered as valuable inputs. Although those sensors are helpful to refine the model, as a challenge, they may also bring heavy computation. Secondly, traffic flow estimation methods should be polished to improve the efficiency. Moreover, deployment in real domain is the key to evaluate our approach.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research has been sponsored in part by National Natural Science Foundation of China nos. 61370151 and 61202211, National Science and Technology Support Program of China 2012BAI22B05, and Huawei Research Foundation YB2013120141.

## References

[1] M. Ángel Sotelo, J. W. C. van Lint, U. Nunes et al., "Introduction to the special issue on emergent cooperative technologies in intelligent transportation systems," *IEEE Transactions on Intelligent Transporation Systems*, vol. 13, no. 1, pp. 1–5, 2012.

[2] A. Warberg, J. Larsen, and R. Jorgensen, "Green wave traffic optimization—a survey," DTU Technical Report, 2008.

[3] S. Heiko and B. Klemens, "Agent-based traffic control using auctions," in *Cooperative Information Agents XI*, pp. 119–133, 2007.

[4] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network- Wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.

[5] R. Junges and A. L. C. Bazzan, "Evaluating the performance of DCOP algorithms in a real world, dynamic problem," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 599–606, ACM, 2008.

[6] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[7] B. C. Silva, D. Oliveira, A. L. C. Bazzan, and E. W. Basso, "Adaptive traffic control with reinforcement learning," in *Proceedings of the 4th Workshop on Agents in Traffic and Transportation*, pp. 80–86, ACM, 2006.

[8] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time—the SCOOT method," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 11–15, 1991.

[9] M. Shenoda and R. Machemehl, "Development of a phaseby-phase, arrival-based," Tech. Rep., SWUTC, 2006.

[10] X.- F. Xie, S. F. Smith, and G. J. Barlow, "Schedule-driven coordination for real-time traffic network control," in *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS '12)*, pp. 323–331, June 2012.

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

[12] J. Ren, X. Ou, Y. Zhang, and D. Hu, "Research on networklevel traffic pattern recognition," in *Proceedings of the IEEE Intelligent Transportation Systems*, pp. 500–504, 2002.

[13] B. C. da Silva, E. W. Basso, A. L. C. Bazzan, and P. M. Engel, "Dealing with non-stationary environments using context detection," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 217–224, June 2006.

[14] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[15] D. de Oliveira, A. L. C. Bazzan, and V. Lesser, "Using cooperative mediation to coordinate traffic lights: a case study," in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 463–470, ACM, Utrecht, The Netherlands, July 2005.

[16] D. Greenwood, B. Burdiliak, I. Trencansky, and H. Armbruster, "GreenWave distributed traffic intersection control," in *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1413–1414, ACM, 2009.

[17] R. Akçelik and M. Besley, "Queue discharge flow and speed models for signalised intersections," in *Proceedings of the 15th International Symposium on Transportation and Traffic Theory*, pp. 99–118, 2002.

[18] K. Sycara, M. Paolucci, M. Van Velsen, and J. Giampapa, "The retsina mas infrastructure," in *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 29–48, ACM, 2003.

[19] Y. Zhang, Y. Xu, T. Sun, and P. Liu, "Green-waved cooperative coordination algorithm for decentralized traffic control," in *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT '12)*, pp. 75–79, Macau, China, December 2012.

[20] A. Warberg, J. Larsen, and R. M. Jorgensen, "Green wave traffic optimization—a survey," Tech. Rep., Technical University of Denmark, 2008.

[21] S. Guo, J. Tan, J. Duan et al., "Characteristics of atmospheric non-methane hydrocarbons during haze episode in Beijing, China," *Environmental Monitoring and Assessment*, vol. 184, no. 12, pp. 7235–7246, 2012.