LEARNING AND COORDINATION FOR AUTONOMOUS INTERSECTION CONTROL

Matteo Vasirani and Sascha Ossowski
Centre for Intelligent Information Technology, University Rey Juan Carlos, Móstoles, Madrid, Spain

Future urban road traffic management is an example of a socially relevant problem that can be modeled as a large-scale, open, distributed system, composed of many autonomous interacting agents, which need to be controlled in a decentralized manner. In this context, advanced, reservation-based, intersection control—where autonomous vehicles controlled entirely by agents interact with a coordination facility that controls an intersection, to avoid collisions and minimize delays—will be a possible scenario in the near future. In this article, we seize the opportunities for multiagent learning offered by such a scenario, studying i) how vehicles, when approaching a reservation-based intersection, can coordinate their actions in order to improve their crossing times, and therefore, speed up the traffic flow through the intersection, and ii) how a set of reservation-based intersections can cooperatively act over an entire network of intersections in order to minimize travel times.

INTRODUCTION

Future urban road traffic management systems will be large-scale, open, distributed systems, where autonomous, computer-equipped vehicles will interact with each other and with the intelligent infrastructure components. The application of hardware and software technology helps to address and alleviate transportation problems, although it enormously increases the complexity of the systems to optimize and keep under control.

This research was partially supported by the Spanish Ministry of Science and Innovation through projects “THOMAS” (TIN2006-14630-C03-02), “AT” (CONSOLIDER CSD2007-0022, INGENIO 2010) and “OVAMAH” (TIN2009-13899-C05-02).

Address correspondence to Matteo Vasirani, Centre for Intelligent, Information Technology, University Rey Juan Carlos, c/Tulipán s/n, Móstoles, Madrid 28933, Spain. E-mail: matteo.vasirani@urjc.es
In recent years, many agent-based techniques have been proposed to tackle traffic management problems (Bazzan 2008). Many approaches aim at optimizing the use of existing traffic infrastructures, by improving the control policies. For example, in (Junges and Bazzan 2008) a constraint optimization model for synchronizing traffic lights is presented, aiming at creating “green waves” in a particular direction. In Kuyer et al. (2008), the traffic lights learn in a coordinated way the best signal plans that lead to low average travel times. Still, in these approaches just the infrastructure is modeled in terms of agents, while drivers are only considered insofar as elements of the traffic flow through the road network that cannot be individually targeted by the control policy.

Other approaches conceive the drivers as the agents whose behavior is to be modeled. In this context, it is particularly interesting to study mechanisms that influence driver behavior so as to improve their local utility (e.g., to reduce travel time) and/or to enhance the global system performance (e.g., to reduce occurrence of congestion). Variable message signs and on-board driver information systems, for instance, can help agents avoid congested road sections (Hernandez, Ossowski, and Garcia-Serrano et al. 2002), but it can cause new problems when used by a large population of drivers (Bazzan and Klügl 2005; Yamashita, Izumir, and Kurumatani et al. 2005). However, an integration with the aforementioned approaches is difficult, since existing urban road traffic management infrastructures based on traffic light controlled intersections affect traffic flows but cannot act on individual vehicles.

On the other hand, air traffic control allows a more fine-grained control over the traffic flow. For example, in Tumer and Agogino (2007), air-traffic control agents manage the “fixes” that airplanes pass through, and learn the optimal delay to introduce between flights in order to minimize travel times and congestion. These infrastructure agents control the system in a distributed way, acting directly upon the individual entities (airplanes) that compose the (air) traffic flow, a possibility that, as we have argued, does not hold for today’s urban road traffic management.

Nevertheless, this is likely to change in a future not too far from now. The problem of advanced intersection control (Dresner and Stone 2008) is being discovered as a promising application field for multiagent systems applied to the traffic and transportation domain. In this context, vehicles, controlled by autonomous intelligent agents, interact with a coordination facility that controls the traffic flow of an intersection, with the aim of avoiding collisions and minimizing delays.

In this work, we evaluate two different coordination scenarios that arise in the reservation-based mechanism for intersection control, as proposed by Dresner and Stone (2008). In particular, we investigate i) how vehicles, when approaching a reservation-based intersection, can coordinate their...
actions in order to improve their crossing times, and therefore speed up the traffic flow through the intersection; and ii) how a set of reservation-based intersections can cooperatively act over an entire network of intersections in order to minimize travel times.

RESERVATION-BASED INTERSECTION CONTROL

In this work, we draw upon the model of intersections proposed by Dresner and Stone (2008), which aims at implementing a reservation-based mechanism to regulate intersections. In this work, an intersection is not regulated by traffic lights, rather by an intelligent agent that assigns reservations of space and time to the vehicles that want to cross the intersection. Due to the high infrastructure costs associated with this system, it is intended as a possible substitute of traffic light systems for high capacity intersections with several lanes. Such control mechanism has shown in simulation a lot of potential to lower delays and increase throughput of intersections if compared to traffic lights, including in mixed scenarios with human drivers and autonomous vehicles (Dresner and Stone 2007).

Their work assumes two types of agent:

- **Intersection manager agents** control the space of an intersection and schedule each individual vehicle’s crossing; and
- **Driver agents** autonomously operate their assigned vehicle.

When a vehicle reaches a minimum distance to the intersection, the driver agent start requesting the intersection manager agent (driver and intersection manager from now on for short) to reserve the necessary time-space slots to safely cross the intersection. The driver provides information such as the vehicle properties (vehicle ID, vehicle size, ...), as well as some properties of the proposed reservation (arrival time, arrival velocity, type of turn, arrival lane, arrival road segment, ...). The intersection manager, provided with such information, is able to simulate the vehicle crossing through the intersection and so determine whether or not a request is in conflict with the already confirmed reservations. If the request is confirmed by the intersection manager, the driver stores the reservation details and tries to meet them, otherwise it slows down and makes another request at a later time (see Figure 1).

If the driver detects changing traffic conditions and estimates that it will not be able to meet the reservation constraints, it can cancel the reservation and make a new one. Doing so is in its interest because a driver can only hold one reservation, and a confirmed reservation whose constraints are not possible to satisfy is useless.¹
Improving the Reservation-Based Mechanism

The reservation-based intersection control offers many opportunities to improve the mechanism by incorporating multiagent learning and coordination mechanisms in the agents of such scenario (Dresner and Stone 2006). We distinguish between two different settings: i) a single reservation-based intersection; and ii) a network of reservation-based intersections.

**Single Intersection**

If we focus on a single intersection, the very multiagent learning resides in the drivers. In the implementation of the reservation-based mechanism proposed by Dresner and Stone, a driver must estimate the arrival time at the intersection, the arrival velocity and the arrival lane without communication nor coordination with the other drivers; each agent makes its request on the basis of its actual velocity, and, if the request is rejected, the driver heuristically slows down and tries again. On the other hand, by letting the agents form teams and coordinate their actions, we provide them with more information that they use to make decisions.

This kind of coordination is somehow related with context-aware route planning of automated guided vehicles (AGVs) (Mohring et al. 2005; Kim and Tanchoco 1991; ter Mors, Zutt, and Witteveen 2007). Context-aware route planning consists in finding a collective optimal and feasible set of routes for several AGVs. In its general form, given a set of route requests,
a centralized route planner computes a feasible and possibly optimal set of routes for the agents, applying shortest path algorithms to find an initial route set and then repairing the initial set of routes in case of conflicts. However, this type of closed-world off-line planning is hard to implement in an open system such as urban road traffic.

**Network of Intersections**

If we focus on a network of intersections, the multiagent learning may reside in the drivers as well as in the set of intersection managers. In the current implementation of the reservation-based mechanism, the reservations are granted in a “first-come-first-served” fashion and without any associated cost. In this setting, a driver does not have incentives to prefer a particular intersection over another. In other words, when it starts to travel from its origin to its destination, it is likely to choose the route with the lowest (estimated) travel time. A possible improvement may derive by giving the intersection managers the possibility of applying a price to the reservations that they grant. In this way, they may cooperate and try to influence the driver behavior to better allocate the urban the road network.

**LEARNING AND COORDINATION WITH A SINGLE INTERSECTION**

In this part of the work, we study the performance, in terms of average travel time needed to cross an intersection, of a set of driver agents that cooperatively coordinate their reservation requests when approaching the intersection.

**Driver Agent Model**

A driver in the traffic domain is generally modeled as a selfish agent. Still sometimes purely selfish behavior is not always beneficial when the agents have to share a congestible resource (Christodoulou Koutsoupias 2005). In the case studied in this work, driver agents can behave in a selfish way, without communicating nor coordinating their actions when approaching the intersection, and figuring out a reservation request that will not be rejected. On the other hand, they are free to decide to coordinate their actions cooperatively, if such coordination benefits the agent. In the experiments, we will compare two different settings (selfish drivers vs. cooperative drivers), and we will show how cooperation benefits the drivers in case of medium traffic density, whilst it does not worsen the average travel time in case of low and high traffic density.
Attributes

We assume that a driver has a preferred speed, $v_p$, which cannot be greater than the speed limit. The driver tries to proceed at this speed when crossing the intersection, depending on the traffic conditions.

Constraints

When a driver approaches an intersection on a given lane, it cannot change it. In this way, if a front vehicle proceeds at a lower velocity, the driver is obliged to slow down. Furthermore, as demonstrated in (Dresner Stone 2008), it is not convenient that the drivers could turn from any lane, so in our model turning right (respectively left) is only possible from the rightmost (respectively leftmost) lane of a road.

Action Space

The actions that a driver can autonomously take are related to the speed at which it crosses the intersection. In particular, an agent could set its velocity to a value in the set $\{1, 2, \ldots, v_p\}$.

So, for the generic driver $i$, the variable $x_i$ that represents an action is the tuple defined as follows:

$$x_i = (ID, D, U, L, t_a, v_a),$$

(1)

where $ID$ is the driver’s identification number; $D \in \{N, S, E, W\}$ is the exit lane from the intersection; $U \in \{LEFT, RIGHT, STRAIGHT\}$ is the type of turn; $L$ is the lane occupied by the agent, $t_a$ is the arrival time at the intersection, and $v_a$ is the arrival speed at the intersection. The component $t_a$ of the tuple $x_i$ is implicitly set by the specific $v_a$ and the vehicle actual position, while the components $ID$, $D$, $L$ and $U$ are constant parameters.

Utility Function

The ultimate goal of the generic driver $i$ is minimizing the time to cross the intersection. This time depends not only on its speed while crossing the intersection, but also on the conflicts that may occur among different competing requests. Let $C$ be a set of drivers. The vector $x = [x_1, x_2 \ldots x_n]$ defines the joint action of this set of agents. A possible function $^2$ that rates “how good” is a joint action is:

$$G(x) = -[1 + P(x)] \cdot D(x),$$

(2)

where $P(x)$ is the number of conflicts among the reservation requests contained in the full joint action $x$; and $D(x)$ is the time spent by the agents to
cross the intersection. We remark that a generic joint action \( \mathbf{x} \) contains all the necessary information to simulate the crossing of the vehicles through the intersection, in the same way it is done by the intersection manager, so that it is also possible to calculate the travel time as well as the number of conflicts among them.

We define the utility function of the drivers on the basis of the theory of COllective INtelligence (COIN) (Wolpert and Tumer 2001). The aim of COIN is studying the properties that a utility function (or reward function) of a learning agent situated in a multiagent environment must meet. COIN introduced the concepts of factoredness and learnability of an agent private utility function. A private utility function \( g_i(\mathbf{x}) \) is meant to be factored if it is aligned with the global utility \( G(\mathbf{x}) \), i.e., if the private utility increases, the global utility does the same. Furthermore, it should be easily learnable, i.e., it should enable the agent to distinguish its contribution to the global utility from that of the other agents. For example, the Team Games Utility, \( TGU_i(\mathbf{x}) \equiv G(\mathbf{x}) \), is trivially factored, but is poorly learnable. If for example agent \( i \) takes an action that actually improves the global utility, while all the other agents take actions that worsen the global utility, the agent wrongly believes that its action was bad.

Wolpert and Tumer (2001) defined a fully factored and highly learnable utility function, the Difference Utility, defined as follows:

\[
DU_i(\mathbf{x}) = G(\mathbf{x}) - G(\mathbf{x} | x_i \leftarrow c_i),
\]

where \( \mathbf{x} \) if the joint action of the collective; \( G(\mathbf{x}) \) is the global utility derived from such a joint action; and \( G(\mathbf{x} | x_i \leftarrow c_i) \) is the joint action where all the components of \( \mathbf{x} \) affected by agent \( i \) are replaced with a constant factor \( c_i \). If this constant is the null action, the Difference Utility is equivalent to the global utility minus the global utility that would have arisen if the agent \( i \) had been removed from the system.

Such an utility function is aligned with the global utility; in fact, since the second term in Eq. (3) does not depend on the action taken by agent \( i \), any action that improves \( DU_i(\mathbf{x}) \) also improves the global utility \( G(\mathbf{x}) \). Furthermore, it is more learnable than the Team Games Utility, because removing agent \( i \) from the dynamics of the system provides a more informative signal to the agent about its contribution.

In our case, the driver computes \( DU_i(\mathbf{x}) \) as follows:

\[
DU_i(\mathbf{x}) = -[1 + P(\mathbf{x})] \cdot D(\mathbf{x}) + [1 + P(\hat{\mathbf{x}})] \cdot D(\hat{\mathbf{x}}),
\]

where \( \hat{\mathbf{x}} = [x_1 \ldots x_i \ldots x_{i+1} \ldots x_n] \).
Coordination Mechanism

We draw upon Probability Collectives (PC) (Wolpert 2006) as coordination mechanism for the distributed decision making. PC, developed within the COIN framework, replaces the search in the space of joint actions with the search in the space of probability distributions over those actions.

Formally, let $C$ be a collective of $n$ agents. Each agent can take an action by setting its action variable $x_i$, which can take on finite number of values from the set $\mathcal{X}_i$. So these $|\mathcal{X}_i|$ possible values constitute the action space of agent $i$. The variable describing the joint action is $x = [x_1, x_2, \ldots, x_n] \in \mathcal{X}$, with $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$.

The collective goal of such a set of agents is selecting the best joint action $x$ that expectedly optimizes the objective function. Such combinatorial problem can be transformed into a continuous optimisation problem, if each agent assigns a probability to each action it can take. Be $q_i(x_i)$ a probability distribution over agent $i$’s actions, the goal of PC is to induce a product distribution $q = \prod_i q_i(x_i)$ that is highly peaked around the $x$ that maximizes the objective function of the problem, and then obtaining the best joint action $x$ by sampling $q$.

The main result of PC is that the best estimation of the distribution $q_i$ that generates the highest expected utility values is the minimizer of the Maxent Lagrangian:

$$\mathcal{L}_i(q_i(x_i)) = E_q[g_i(x)] - T \cdot S(q_i(x_i)), \quad (5)$$

where $q_i(x_i)$ is the probability distribution over agent $i$’s actions; $g_i(x)$ is the agent private utility function [e.g., the Difference Utility defined in Eq. (3)], which maps a joint action into the real numbers; the term $E_q[g_i(x)]$ is the expected utility value for agent $i$, subjected to its action, and the actions of all the agents other than $i$; $S(q_i(x_i))$ is the Shannon entropy associated with the distribution $q_i(x_i)$, $S(q_i(x_i)) = -\sum x_i q_i(x_i) \ln [q_i(x_i)]$; $T$ is an inverse Lagrangian multiplier.

Since the Maxent Lagrangian is a real valued function of a real valued vector, it is possible to use gradient descent or Newton methods for its minimization. Using Newton methods, the following update rule is obtained:

$$q_i^{t+1}(x_i) = q_i^t(x_i) - \alpha \cdot q_i^t(x_i) \times \left\{ \frac{E_q[g_i| x_i] - E_q[g_i]}{T} + S(q_i^t(x_i)) + \ln [q_i^t(x_i)] \right\}, \quad (6)$$

where $E_q[g_i]$ is the expected utility; $E_q[g_i| x_i]$ is the expected utility associated with each of the agent $i$’s possible actions; and $\alpha$ is the update step.
Equation (6) is the update rule with which the agents should modify their distributions in order to jointly implement a step in the steepest descent direction of the Maxent Lagrangian.

Since at any time step \( t \), an agent might not know the other agents’ distributions, it would not be able to evaluate any expected value of \( g_i \), because they depend on the full probability distribution \( q \). Those expectation values can be estimated by repeated Monte Carlo sampling of the distribution \( q \) to produce a set of \((x; g_i(x))\) pairs. Each agent \( i \) then uses these pairs to estimate the values \( E_{q_i}[g_i|x_i] \), for example by uniform averaging the \( g_i \) values in the samples associated with each possible action.

**PC in Reservation-based Intersection Control**

Usually, the Lagrangian minimisation proceeds as follows: for a given \( T \), the agents jointly implement a step in the steepest descent direction of the Maxent Lagrangian using Eq. (6). Then \( T \) is slightly reduced, and the process continues, until a minimum value is reached. \( T \) is geometrically reduced as long as a driver approaches the point after which it starts requesting a reservation to the intersection manager, as described in the section Reservation–Based Intersection Control. When a driver arrives at that point, it evaluates the action with the highest probability, sets its speed accordingly and makes a reservation request with the given speed.

Algorithm 1 sketches the algorithmic structure of the driver program that implements PC in the reservation-based intersection scenario. The algorithm starts initializing \( T \) and the probability distribution \( q_i \) (line 01 and 02). The initial probability distribution \( q_i \) is initialized with the maximum entropy distribution, i.e., the uniform distribution over the action space \( v_i \). In this way, no assumptions are made about the desirability of each action, and all the actions are equiprobable. Furthermore, each agent allocates in memory a vector \( m \) to store the samples sent by the other agents to create the set of sampled joint actions (used to evaluate the conditional expected values).

The main loop controls the progressive reduction of \( T \) (line 09), until the drivers reach the minimum distance to the intersection (line 03). The minimization of \( L_i \) for a given \( T \) is accomplished by repeatedly determining all the conditional expected values \( E_q[g_i|x_i] \) (line 07), and then using these values to update the distribution (line 08). Such values are obtained by requesting samples to the drivers in the collective (line 05) and storing them as soon as they are received (line 11), in order to have an estimation of the entire distribution \( q \).

At the end of the algorithm, agent \( i \) chooses its best action by selecting the action with the highest probability, and then it stores the request that will be sent to the intersection manager (line 16). The requests of all the
agents that participate in the coordination phase are expected to maximize the objective function, i.e., they generate no conflicts and minimize the travel time to cross the intersection.

Algorithm 1 PC in reservation-based intersection control

\begin{algorithm}
\begin{algorithmic}
\STATE $T \leftarrow 1$
\STATE $q_i \leftarrow \text{uniformDistribution}$
\STATE $m \leftarrow \text{allocateMCSamplesVector}$
\WHILE {minimum distance not reached}
\STATE requestMCSamples
\IF {m not empty}
\STATE $ce \leftarrow \text{evalConditionalExpectations}(m)$
\STATE $q_i \leftarrow \text{updateQ}(ce)$
\ENDIF
\STATE $T \leftarrow \text{update T}$
\STATE $m \leftarrow \text{storeIncomingMCSamples}$
\ENDWHILE
\STATE $\hat{x}_i \leftarrow \text{mostProbableAction}$
\STATE $v_a \leftarrow \hat{x}_i, v_a$
\STATE store request $R = (ID, D, L, U, t_a, v_a)$
\end{algorithmic}
\end{algorithm}

After the execution of algorithm 1, the coordination phase ends and each driver switches to the “normal” behaviour of the reservation-based mechanism. It sends the reservation request that came out from the coordination phase to the intersection manager, until it receives a confirmation or a refuse message. In the first case, the driver stores the reservation details and tries to meet them. Otherwise, it decreases its speed and makes another request in the next step.

Experimental Results

In this section we present the results of the experiments conducted with a microscopic simulator of an intersection with 4 approaches, each of them with 6 lanes (see Figure 2). The metric we used to evaluate the efficiency of the intersection was the average travel time of the set of generated vehicles. During the simulation, a total of 100 vehicles are generated using a Poisson distribution $f(k, \lambda) = \lambda^k e^{-\lambda} / k!$, where $\lambda$ is the number of expected vehicles that are spawned in a given interval. In all the experiments, the $\lambda$ parameter is kept fixed, while we progressively reduce the interval, simulating in this way increasing traffic densities.

Each spawned vehicle has a preferred speed, whose value is generated randomly using a Gaussian distribution with $\mu = 3$ and $\sigma^2 = 1$, and the speed limit was set to 10.
Table 1 shows the average travel time for two different configurations. In one configuration, each driver communicates exclusively with the intersection manager and makes reservation requests solely on the basis of its knowledge (NO COORD); in the other configuration, the drivers implement the coordination mechanism before starting making reservation requests (COORD).

If the traffic density is low, the average travel time of the two configurations is approximately the same. This is reasonable, since with low traffic density few reservation requests are rejected, so no previous coordination is needed. Similarly, with high traffic density the average travel time tends to be the same for the two configurations. Again this is reasonable, because the intersection tends to be saturated by vehicles stopped at the intersection, waiting for its reservation request to be confirmed. On the other hand, in case of medium traffic density, the coordination between drivers reduces the average travel time by 6.84%, due to a lower number of refused reservations (see Figure 3).

<table>
<thead>
<tr>
<th>Density</th>
<th>COORD</th>
<th>Stddev</th>
<th>NO COORD</th>
<th>Stddev</th>
<th>%Δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low density</td>
<td>143.95</td>
<td>5.3</td>
<td>143.9</td>
<td>4.9</td>
<td>+0.04%</td>
</tr>
<tr>
<td>Medium density</td>
<td>228.16</td>
<td>3.1</td>
<td>244.91</td>
<td>3.0</td>
<td>−6.84%</td>
</tr>
<tr>
<td>High density</td>
<td>269.34</td>
<td>4.3</td>
<td>269.19</td>
<td>4.5</td>
<td>+0.05%</td>
</tr>
</tbody>
</table>

FIGURE 2 Intersection simulator.
The slight reduction of the average travel time for the medium traffic density denotes that the driver action space is quite reduced, since a driver can only set the speed at which it intends to cross the intersection. For example, if there is a confirmed reservation of a very slow vehicle, which occupies the intersection for many time slots, it is reasonable to think that there is no way for an approaching driver to make a request that will not be rejected, no matter the speed (and the arrival time) it proposes. This reduced action space allows to cope with the real-time constraints, making the coordination problem more tractable. Nevertheless, a possible improvement could derive from giving the agents the possibility of changing its lane when approaching the intersection.

Furthermore, the global utility function $G(x)$ of Eq. (2) does not take into consideration external factors (i.e., noise). In the domain of the intersection control, for a given $x = [x_1 \ x_2 \ldots \ x_n]$, a driver is only able to evaluate the number of conflicts that occurs among the $n$ agents and their travel times, by simulating the crossing of each agent $i$ through the intersection. If for example the intersection is saturated due to a crash, or it has been reserved by very slow vehicles, the collective is not able to react to these events and adjust its collective behavior, since it does not have such information.

A way to circumvent this problem is modifying the structure of the global utility as a 2-players game between the collective and the external world. At each time step, the collective sets its joint action $x$, while the world plays $y$. Then the global objective $G(z)$ is calculated, as a function of the full vector $z = [x \ y]$. The vector $y$ represents what is not directly under control of the collective, and could contain information about the confirmed reservations that the intersection manager holds in its database.
LEARNING AND COORDINATION WITH A NETWORK OF INTERSECTIONS

The study presented in the previous section aims at improving the efficiency of a single intersection, allowing the drivers to communicate and coordinate their actions when approaching a reservation-based intersection. If we focus on a network of reservation-based intersections, there are more options to improve the efficiency of the system as a whole. In the current implementation of the reservation-based mechanism, the reservations are granted to the first requester without any associated cost, so that a driver that wants to go from its origin to its destination in the road network is likely to choose the shortest path. This is a particular case of a congestion game (Christodoulou and Koutsoupias 2005), where such greedy behavior could generate highly suboptimal performance of the system.

A way to shape the drivers decision making and to give the intersection managers a lever to affect the driver decision making is letting them sell the time-space slots at the intersection, thus generating incentives to prefer or to avoid routes that pass through certain intersections (if people usually are very patient in tolerating traffic jams, they are not so willing to waste their money).

In this section, we draw upon market-based control methods (Clearwater 1996) as a paradigm for the design of the traffic control mechanism at the network level. We model the system as a computational economy, where drivers trade with the infrastructure agents in a virtual marketplace, purchasing reservations to cross intersections when commuting through the city. We design the market rules with the aim of aligning the global profit with the social welfare (i.e., average travel time), in a way that, in situations of similar traffic load, an increase of the infrastructure’s profits usually implies a decrease of the drivers’ average travel time.

Trading Model

The modeling of the market rules aims at aligning the objective pursued by the intersection managers, i.e., maximizing the global profit, with the social welfare of the drivers, i.e., lower travel times. The idea is that by striving for selling as much reservations as possible, the intersection managers tend to make their intersections more efficient, generating in this way less congestions and lower average travel times. The trading interactions between a driver and an intersection manager that define our marketplace are:

**Purchasing a Reservation when Approaching the Intersection**

In order to purchase a reservation, when a driver is in the proximities of an intersection manager, it provides it with the necessary data to simulate its
crossing through the intersection (Dresner and Stone 2008). If the request cannot be satisfied, due to conflicts with the already confirmed reservations, the intersection manager refuses the reservation request. Otherwise, it notifies the driver with the reservation fare. The driver then transfers a percentage of the fare in terms of reservation fee, while the intersection manager commits itself to maintain the contracted fare for the driver. When the driver actually arrives at the intersection, it pays the remaining amount and crosses safely.

Notice that this protocol presumes that drivers hold a reservation in order to safely cross the intersection. Although nothing can physically impede the driver to cross the intersection without a reservation, we rely on the assumption that a driver is risk averse and does not take the risk of causing an accident by crossing the intersection without holding a confirmed and up-to-date reservation. This assumption also holds for today’s intersections regulated by traffic lights or stop signs (Dresner and Stone 2008). Furthermore, a driver has no incentives for risking an accident for monetary purposes, avoiding to pay for a reservation, since it has the possibility of having a reservation for free if it stops at the intersection (see following).

*Receiving a Reservation when Stopped at the Intersection*

If a driver reaches the edge of the intersection and it does not hold a valid reservation, it must stop. In this case, it is entitled to receive a reservation, when available, for free. The intersection manager may give priority to buyers; still it has incentives for eventually granting such reservation because it is likely that the bigger the number of stopped vehicles, the less capacity there is for payer drivers to cross.

*Withdrawing a Reservation*

When a driver is holding a reservation, it tries to meet the reservation constraints, specially the arrival time. If it realizes that these cannot be met, due for example to changing traffic conditions, it can withdraw the reservation. In this case, the driver will lose the reservation fee paid in advance.

*Intersection Manager Agent Model*

Since intersection managers are part of the infrastructure, we can design them as cooperative global utility maximizers. An intersection manager, as a seller, is free to set its desired fare, within a certain price range, for the reservations at the intersection that it manages. Still, the ultimate objective of each intersection manager is maximizing the global profits raised by the whole team of intersection managers.
**Action Space**

The action space $Z_j$ of an intersection manager is formed by the prices that it can apply to the reservations that it manages. More formally:

$$Z_j = [p_{j_{\text{min}}}, p_{j_{\text{max}}}],$$

where $p_{j_{\text{min}}}$ and $p_{j_{\text{max}}}$ are the minimum and maximum allowed price for a reservation fare.

**Profit Function**

An intersection manager is characterized by its profit function $U_j$, defined as the difference between revenue $R_j$ and cost $C_j$. More formally:

$$U_j = R_j - C_j = \sum z_j - C_{\text{max}} \cdot e^{-\sum_i d_i}.$$

The revenue $R_j$ is calculated as the money earned with the reservations that have been sold over time at price $z_j$. The cost function $C_j$ is a function of the number of drivers that have purchased a reservation through time. The cost function has a maximum if no drivers have purchased a reservation, and tends to zero with the increase of drivers (i.e., the costs are amortized). Such profit function has a twofold objective: from one side it intends to penalize unused intersections, by a mean of low revenue and high cost; from the other side, also congested intersections will be penalized, since vehicles stopped at the intersection do not generate any revenue (recall that a vehicle stopped at the intersection is entitled to receive a reservation for free).

**Driver Agent Model**

The deliberation process of a driver is shaped by the fact that it must purchase the necessary reservations to cross the intersections that it encounters during its trip. This process includes taking decisions such as selecting with which seller it wants to trade or allocating a certain amount of money for each intersection it must cross.

We model the drivers as individually rational utility maximizers, which are willing to minimize travel time and monetary cost of a route. We model the utility function $B_i$ of a driver as:

$$B_i = -[\rho \cdot TT + (1 - \rho) \cdot K],$$
where $\rho$ is a trade-off factor that weights the relative importance of travel time, $TT$; and costs, $K$. Tuning the parameter $\rho$, we can model different driver profiles, from drivers more concerned about travel times ($\rho = 1$) that are willing to pay any price to cross their more preferred intersections, such as business drivers, to drivers that are more concerned about monetary costs of routes ($\rho = 0$), such as leisure drivers (Vasirani and Ossowski 2009).

The cost component $K$ is simply the total price of the intersections that lay on the route. We assume that drivers can be provided with the current prices of the intersections in the network. This can be done, for instance, by a price propagation scheme through the intersection network.\(^5\)

Although the travel time function $TT$ is unknown in general, it is usually monotonically increasing with the density (vehicles per km) of the links of the route. Therefore, a driver agent uses an optimistic estimation $TT^{\text{est}}$ of the travel time, calculated as:

$$TT^{\text{est}} = \sum_k \frac{||s_k||}{v_k^{\text{max}}},$$  \hspace{1cm} (10)

where $||s_k||$ is the length of a section of the driver route; and $v_k^{\text{max}}$ is the maximum allowed speed on section $k$.

We simulate the driver decision making over time as follows: when a driver intends to travel through the road network, firstly it builds its route, then it starts traveling following that route, and when it is approaching the first intersection on its path it tries to purchase a reservation with the forthcoming intersection manager. The driver re-calculate its route after crossing each intersection, so reacting to the market fluctuations.

**Environment Model**

We assume that the intersection managers are able to communicate with each other, and drivers can communicate with the intersection manager of the forthcoming intersection on its route, and also with the neighbors of such intersection manager. Drivers can be provided with the current prices of the intersections in the network. Finally, we assume that a trusted payment system is available, such in today’s toll roads, allowing drivers to securely transfer money to intersection managers when required.

**Coordination Mechanism**

Every intersection manager must decide which fare to apply to its reservations in order to maximize the profit. The decision is not trivial, because
if the fare is too high, the intersection is likely to be avoided by the drivers, generating low profits. On the other hand, if the fare is too low, the intersection is likely to be congested, resulting again in low profits. Furthermore, a fare cannot be defined high or low in absolute terms, but is strictly dependent on the fares applied by the other (possibly neighboring) intersection managers.

The entire collective of intersection managers is so faced with two tasks: 1) discovering the effect of a specific price vector; and 2) coordinating their fares in order to maximize the global profit (we assume that the intersection managers act as a team and aim at jointly maximizing the profit). Given Eq. (8), the global objective of the team of intersection managers can be expressed as a function of the joint action \( z \), \( G(z) = \sum_j U_j \), where \( z \) is the price vector of the team of intersection managers, \( z = [z_1, z_2, \ldots, z_n] \). We remark that the functional form of \( G \) as a function of \( z \) is not known, since it is not known which profit \( U_j \) is generated by a specific \( z \).

We use Q-learning with immediate rewards and \( \epsilon \)-greedy action selection (Watkins 1989) to learn which fare scheme leads to the best global profit \( G \). After having taken action \( z_j \) (i.e., setting the price of the reservation fare), the learning agent receives a reward that rates that action, then it updates its action-value function estimation as follows:

\[
Q_{t+1}(z_j) = Q_t(z_j) + \alpha \cdot [r(z) - Q_t(z_j)],
\]

where \( \alpha \) is the learning rate and \( r(z) \) is the reward, which depends on the full joint action \( z \). Each intersection manager selects a random action with small probability \( \epsilon \), and the greedy action (i.e., the action with highest Q-value) with probability \( 1 - \epsilon \).

For the reward structure, we rely again on the work of Wolpert and Tumer (2001), which aims at defining the characteristics of reward functions in multiagent domains, so that the distributed maximisation of such rewards leads to good overall system performance. The main result of COIN is that difference rewards of the form:

\[
D_j(z) = G(z) - G(z|z_j \leftarrow c_j),
\]

are both sensitive to the agent actions and aligned with the overall system reward. Here, with the notation \( z|z_j \leftarrow c_j \) we refer to a vector where all the components of \( z \) affected by agent \( j \) are replaced by the constant \( c_j \).

Since it is not possible to calculate the term \( G(z|z_j \leftarrow c_j) \) when the functional form of \( G \) is not known, we follow (Tumer and Agogino 2007) for the estimation of the difference reward. If \( G(z) = G_j(f(z)) \), where \( G_j \) has a
known functional form, and if \( f(z) = \sum_j f_j(z_j) \), where each \( f_j \) is an unknown function, the difference reward can be estimated as:

\[
D_j^{\text{est}}(z) = \mathbb{E} \left[ \sum_j U_j | z_j \right] - \mathbb{E} \left[ \sum_j U_j \right].
\]  
(13)

In other words, the difference reward can be estimated as the expected global profit when agent \( j \) applies fare \( z_j \) minus the expected global profit.

**Experimental Results**

To evaluate the approach here presented we implemented a hybrid mesoscopic-microscopic simulator. The traffic flow on the roads is modeled at mesoscopic level, where the dynamics of a vehicle are governed by the average traffic density on the link it traverses rather than the behavior of other vehicles in the immediate neighborhood as in microscopic models (Schwerdtfeger 1984). Since the mesoscopic model does not offer the necessary level of detail to model a reservation-based intersection, when a vehicle enters an intersection its dynamics switch into a microscopic, cellular-based simulator, whose update rules follow the Nagel-Schreckenberg model\(^6\) (Nagel and Schreckenberg 1992).

The cell size is set to 5 meters, and for simplicity, we assume that vehicles cross the intersection at a constant speed, so that any additional tuning of parameters, such as slowdown probability or acceleration/deceleration factors, is not necessary.

We instantiated the simulation environment using the main roads network of metropolitan Madrid’s area (see Figure 4, where each big dark vertex is an intersection). For the intersection manager model, the minimum and maximum prices \( p_j^{\text{min}} \) and \( p_j^{\text{max}} \) were set to 1 and 10, respectively, for all the intersection managers [Eq. (7)], while the maximum cost \( C_{\text{max}} \) was set to 1 [Eq. (8)]. Regarding the learning algorithm, we selected by trial-and-error \( \alpha = 0.5 \) and \( \epsilon = 0.1 \) [Eq. (11)], since such values imply fast convergence and enough exploration. In order to guarantee initial exploration, the Q-values are initialized optimistically with the maximum global profit (see below). For the driver model, the trade-off factor \( \rho \) of the utility function was set to 0.5 [see Eq. (9)]. In other words, the decision making of the driver is equally affected by the estimated travel time of a route and the fares applied by the intersection managers laying on that route. We simulated 2000 drivers (the double of the optimal flow in the simulated road network) commuting along the North-South axis (from A and B to C and D), generated in an interval of 15 minutes.
Figure 5 plots how the global profit evolves during the learning. The maximum global profit, used as baseline, is the global profit that would have been obtained if all the intersection managers sold a reservation at the maximum allowed price to all the drivers (i.e., if every driver crosses
all the intersections). The intersection managers are able to jointly find a fare scheme that generates high profits. Nevertheless, it is interesting to see the effect of this profit maximization on the average travel time.

Figure 6 plots the average travel time of the set of 2000 drivers. The lowest limit is represented by the average travel time at free flow, i.e., when there is no traffic. The highest limit is represented by the average travel time of the 2000 drivers if we remove the market mechanism. In this case, the drivers always select the shortest path, generating congestions and, consequently, higher travel times. On the other hand, if the intersection managers try to maximize profit, they indirectly influence the driver decision making and so they better allocate the road network resource, generating lower travel times over time.

This fact is remarked also by the average distance covered by the drivers (see Figure 7). Without the market, the average covered distance corresponds with the shortest route between origins and destinations. On the other hand, when the agents must participate in the virtual market, the maximization of the global profit performed by the intersection managers makes the drivers spread through the network, causing higher covered distances. Nevertheless, an increase of the average covered distance of about 15% (Figure 7) generates a decrease of the average travel time of about 40% (Figure 6).

This is a significant result, because shows that the objective pursued by the intersection managers is aligned with the social welfare. Furthermore, the maximization of the global profit is a much more tractable problem with respect to the direct maximization of the social welfare. To optimize

**FIGURE 6** Average travel time.
the global profit of a team of agents, each agent needs to evaluate $G(z)$ for a specific, observed, $z$. This can be done in a decentralized way, using for example a gossip-based protocol (Jelasity, Montresor, and Babaoglu 2005) to aggregate the local profit values into a global value. On the other hand, to directly optimize the social welfare, the intersection managers ought to receive a feedback with the travel time of all the driver agents, a requirement that is pretty unrealistic.

**CONCLUSION**

This work showed that the reservation-based intersection control problem offers many opportunities for multiagent learning (Dresner and Stone 2006). Focusing on a single reservation-based intersection, we applied a coordination mechanism based on Probability Collectives to make the drivers learn to coordinate their action. The experiments showed that if the traffic density is low, the coordination mechanism does not generate benefits in terms of lower travel times, since few reservation requests are rejected. Similarly, with high-traffic density, the intersection tends to be saturated by vehicles stopped at the intersection, waiting for its reservation request to be confirmed, so that again any previous coordination is not fruitful. On the other hand, in case of medium traffic density, the coordination between drivers reduces the average travel time up to 6.84\%, due to a lower number of refused reservations.
More promising improvements have been obtained with a network of reservation-based intersections. We designed regulations of our marketplace, minimizing the possibilities for strategic manipulation, and fostering the alignment between the profit maximization, and the social welfare maximization. We implemented a learning mechanism for intersection managers, so as to coordinate their pricing policies and maximize the global profit. We analyzed the system performance using a reasonable model of individually rational driver behaviour, based on the notion of route choice. We showed that, in general, the market is modeled in such a way that an increase in the global profit of the intersection manager team generates a reduction of the drivers average travel time.

This work can be extended in many ways. Different reward functions as well as different demand patterns (i.e., different driver profiles, different distributions of drivers, etc.) may be evaluated. Furthermore, in the above experiments the price of a reservation requested at time $t$ was unique, while it is reasonable to assume that the value of a reservation is inversely proportional to how far in advance it is purchased. The impact of the road network topology on the system performance can be taken into account, as well as studying the system behavior in a co-learning setting, where both the drivers and the intersection managers learn from their experience.

In the present work, there was no explicit negotiation between drivers and intersection managers. A possible extension is letting the drivers and intersection managers reach an agreement on an acceptable price for both agents. Similarly, intersection managers could form coalitions, aiming at improving the coalition’s profit rather than the global one. Finally, the market can be enriched allow intersection managers to offer a range of different pricing instruments (e.g., off-peak pricing, daily subscriptions, and fares based on usage of the intersection).

NOTES

1. As remarked by Dresner and Stone, the reservation-based mechanism relies on the assumption that a driver does not have incentives to cross an intersection without a reservation, or with an out-of-date reservation (e.g., whose arrival time does not coincide with the real arrival time). Although nothing can “physically” impede the driver to cross the intersection, a driver is assumed to be a risk averse entity that does not take the risk of causing a crash, by crossing an intersection without a reservation. We remark that this assumption also holds for intersections regulated by traffic lights.

2. It is possible to formulate other objective functions that take in consideration a different relationship between conflicts and time, as well as including other aspects, such as congestion or lane changes.

3. Without loss of generality, the global utility function is considered as a “cost” to be minimized, by simply flipping the sign of the utility value.

4. To this respect, we assume that the road infrastructure provides intersection managers with a way to actually confirm that a vehicle is stopped at the intersection (e.g., with cameras), in order to avoid that the mechanism is exploited by strategic drivers.
5. See for example (Jelasity, Montresor, and Babaoglu 2005), where a gossip-based, adaptive protocol for extremely large and highly dynamic networks is presented. Such protocol has been successfully tested on a network distributed over five continents, whose number of nodes dynamically oscillated between 2500 and 6000.

6. We chose the Nagel-Schreckenberg model for its computational efficiency and also because it fits very well the intersection model introduced by Dresner and Stone. In fact an intersection is modeled as a matrix of cells, and a vehicle, depending on the granularity, occupies one or more cells, exactly as in the cellular-based model of traffic flow of Nagel and Schreckenberg.

REFERENCES


