

## COOPERATIVE DRIVING AT ISOLATED INTERSECTIONS BASED ON THE OPTIMAL MINIMIZATION OF THE MAXIMUM EXIT TIME

JIA WU \*, ABDELJALIL ABBAS-TURKI \*\*, FLORENT PERRONNET \*\*

\* School of Information and Software Engineering  
University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, 610054 Chengdu, China  
e-mail: jiauwu@uestc.edu.cn

\*\* Systems and Transport Laboratory  
University of Technology of Belfort-Montbéliard, Rue Thierry Mieg, 90010 Belfort cedex, France  
email: {abdeljalil.abbas-turki, florent.perronnet}@utbm.fr

Traditional traffic control systems based on traffic light have achieved a great success in reducing the average delay of vehicles or in improving the traffic capacity. The main idea of these systems is based on the optimization of the cycle time, the phase sequence, and the phase duration. The right-of-ways are assigned to vehicles of one or several movements for a specific time. With the emergence of cooperative driving, an innovative traffic control concept, Autonomous Intersection Management (AIM), has emerged. In the framework of AIM, the right-of-way is customized on the measurement of the vehicle state and the traffic control turns to determine the passing sequence of vehicles. Since each vehicle is considered individually, AIM faces a combinatorial optimization problem. This paper proposes a dynamic programming algorithm to find its optimal solution in polynomial time. Experimental results obtained by simulation show that the proper arrangement of the vehicle passing sequence can greatly improve traffic efficiency at intersections.

**Keywords:** cooperative driving, wireless communication, autonomous intersection management, dynamic programming.

### 1. Introduction

Efficient traffic control systems at intersections are in favor of enhancing the performance of urban traffic networks. Over the past half-century, a great deal of traffic control strategies concerning isolated intersections have emerged and have been widely used. From the earlier work of Webster (1958) to the recent works (Robertson, 1969; Allsop, 1971; 1976; Sims and Dobinson 1980; Hunt, 1982; Farges *et al.*, 1983; Gartner, 1983; Sen and Head, 1997; Lee and Hyung, 1999; Boillot *et al.*, 2000; Mirchandani and Head 2001; Mirchandani and Lukas 2001; Péter, 2012), these systems have achieved great success in reducing the average delay of vehicles and in improving the traffic throughput, especially through the emergence of adaptive traffic control systems (Hunt, 1982; Farges *et al.*, 1983; Sen and Head, 1997; Mirchandani and Head 2001; Mirchandani and Lukas 2001). However, the increased importance of environmental concerns and the limited economic and physical resources are among the most important reasons why these traditional traffic control systems

cannot continue being the answer to the ever-increasing transportation and mobility needs of modern societies. Safe, efficient, and less polluting transportation of persons and goods calls for an optimal utilization of the available infrastructure via novel traffic control measures. This trend is driven by the rapid developments in the areas of wireless communications and positioning.

Wireless technologies with positioning systems have received a great attention since they offer a tremendous potential for reinforcing the link between the vehicles (drivers), the traffic environment, and the control system (Uno *et al.*, 1999; Sakaguchi *et al.*, 2000). A new active research area, cooperative driving, has emerged (Tsugawa, 2002). The most basic application of cooperative driving at intersections is to provide information to the driver about the surrounding driving environment. A more advanced application is to elaborate the driving behavior in order to direct autonomous vehicles through an intersection (Arora *et al.*, 2012; Grünwald *et al.*, 2006; Li and Wang, 2006; Mehani and de La Fortelle, 2007; Dresner and Stone,

2008). This application is called Autonomous Intersection Management (AIM).

In the framework of AIM, each vehicle sends request of right-of-way by wireless communication and waits for receiving its right-of-way before crossing the intersection. The right-of-way is customized on the measurement of the vehicle state. The control of AIM determines the sequence of vehicles that cross the intersection instead of designing traffic light switching schemes. AIM is quite different from the traditional traffic control system and presents the following important novelties:

- Unlike the traditional control systems that decide on the duration of phases (the green light given to movements) or whether or not the phase is still kept for another period of time, AIM allows us to precisely decide which vehicle has the right-of-way and addresses the right-of-way to only admitted vehicles.
- The decision is performed according to the actually observed traffic instead of predicting the volume of the traffic from the past statements of the inductive loops.
- It uses cooperative driving techniques to help the driver prepare for or even avoid emergencies.
- It facilitates the deployment of priority policies between vehicles because each vehicle is considered individually.

The concept of AIM theoretically opens up the possibility of exploring a multitude of protocols to manage traffic efficiently at intersections. Li and Wang (2006), enumerated all feasible passing sequences of vehicles. Then trajectory planning algorithms were used to choose the most efficient sequence that guides the vehicles to cross the intersection as soon as possible. The main problem of this method is the complexity of generating passing sequences. As the authors pointed out, the time complexity exponentially increases with the number of vehicle and lanes. Dresner and Stone (2004), Mehani and de La Fortelle (2007) or David *et al.* (2012) introduced a simple control policy, First Come First Serve (FCFS), with low computational cost. Matteo and Sascha (2009) proposed a more complicated control policy, such as the longest-in-system. However, these policies are too simple to explore the potential capabilities of improving traffic efficiency in AIM.

Since researchers have reached the agreement that a proper arrangement of passing sequences will improve the traffic efficiency at intersections, in this paper we are interested in proposing a new traffic control strategy based on dynamic programming for finding an optimal sequence in polynomial time. This paper is organized as follows. First of all, it presents the basic configuration of

the AIM system. Then the traffic control at an intersection (to compute the vehicle passing sequence) is formulated as a combinational optimization problem. A dynamic programming algorithm is proposed to solve the control problem in polynomial time. Finally, the paper discusses the results of simulations before concluding.

## 2. System overview

We propose a new protocol of AIM, which is referred to as Cooperative Vehicle-Actuation Signalization (CVAS). In CVAS, the access to an intersection is ordered according to the requests received from the vehicles via the wireless communication. Each vehicle individually receives its right-of-way with respect to the sequence. Two main technologies are required:

- (a) discrete point positioning technology, which makes the intersection manager aware of the vehicle position,
- (b) wireless communication technology, which helps vehicles to communicate with the surrounding environment.

**2.1. Intersection configuration.** Consider a two-lane intersection as presented in Fig. 1. Each lane is one-way and split into two zones by the conflict zone: the storage zone and the exit one. Position markers are placed at the entrance of each zone. Thus, there are three position markers placed on each lane. When a vehicle detects the position marker 1, it sends a request for the right-of-way to the intersection manager. The position marker 2 inquires the approaching vehicle about its right-of-way. Vehicles without right-of-way cannot cross the conflict zone. After leaving the conflict zone, the position marker 3 detects the vehicle and the intersection manager takes back the vehicle's right-of-way.

Based on this configuration, we can picture the operation of the system (Fig. 2). When a vehicle arrives at the intersection (it passes over the position marker 1), the intersection manager sends the map of the intersection to this vehicle and locks its right-of-way. Then it computes the best sequence and accordingly distributes the appropriate right-of-ways to the first admitted vehicles of non-conflicting movements. When the last admitted vehicle passes over the position marker 3, it proceeds to free vehicles of the conflicting movements and so on. If some vehicles arrive before the end of the sequence, the intersection manager computes a new sequence without considering vehicles that have already received the right-of-way. The objective of this rule is to avoid an emergency braking to the drivers.

**2.2. Safety issues.** From the description given previously, the safety issue is mainly supported by a

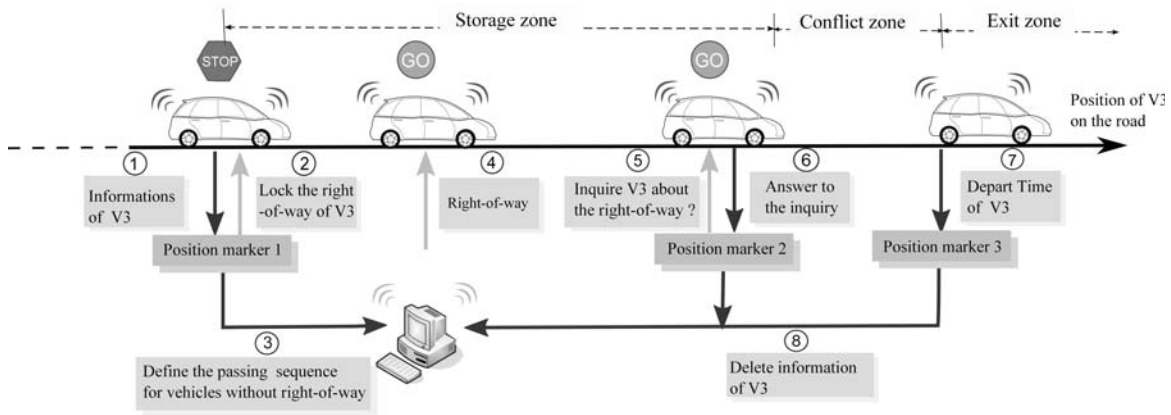


Fig. 2. Communication between a new coming vehicle and the intersection manager.

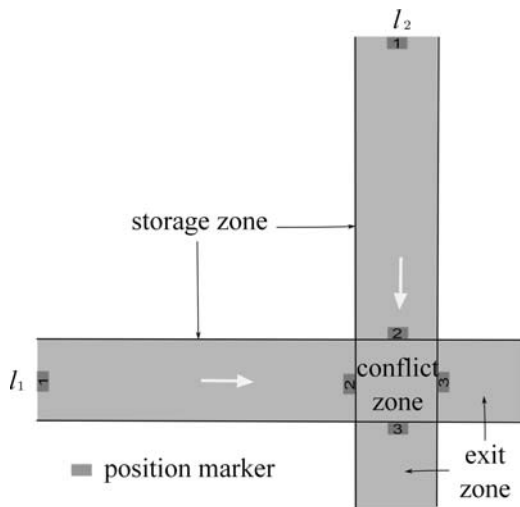


Fig. 1. Configuration of a 2-way intersection.

discrete point positioning system. In addition, the method of distributing the right-of-way is also very important to guarantee the safety at the conflict zone.

Two approaches of negotiating the right-of-way are conducted: the first one is a decentralized approach (Grünewald *et al.*, 2006), where vehicles communicate with each other to obtain the right-of-way; the second one is centralized, where the intersection manager receives the requests of vehicles and sends the right-of-way accordingly (Li and Wang, 2006; Mehani and de La Fortelle, 2007; Dresner and Stone, 2008). In the first approach, the main burden of caring for the safety falls on the performance of the communication system. However, wireless communication is not as reliable as expected. Grünewald *et al.* (2006) showed that the saturation of the bandwidth will cause collision. Hence, a centralized approach is necessary. The intersection manager centralizes the presence list and makes a selection of the vehicles that will receive the right-of-way.

Here, we do not want to go deep into the safety issue. The reader could find more detailed discussions in the work of Abbas-Turki *et al.* (2012), where a P-timed Petri net model (DiCesare *et al.*, 1994) is proposed to analyze the safety of the proposed system.

### 3. Problem formulation

Recall that the main concern of the paper in the context of CVAS is to explore the possibility of improving the traffic efficiency at intersections by means of a proper arrangement of the vehicle passing sequence. Since it is impossible to predict arrivals of all vehicles, a natural way is to find an optimal sequence for the vehicles present at the intersection. As soon as a new vehicle approaches the intersection, the controller recalculates an optimal sequence for vehicles without right-of-way. This control strategy is similar to the traditional rolling horizon technique, where the algorithm finds the optimal solution over each horizon, and the optimization process proceeds one horizon after another. However, since the optimization here is updated as soon as new vehicles approach, it is more flexible to adapt to the fluctuating demand situations. In order to pursue the optimization for the present vehicles, the control problem should be well formulated at first.

**3.1. Formulation of the control problem.** Given the presence list of vehicles by the position markers 1 and the vehicles, theoretical arrival times at the conflict zone, the control of CVAS determines a passing sequence of vehicles which minimizes the maximum exit time, under the constraint of safe headways. Here the exit time refers to the time when the last vehicle traverses the conflict zone. The formulation of the control in CVAS is presented in the following (the notation is defined in Table 1):

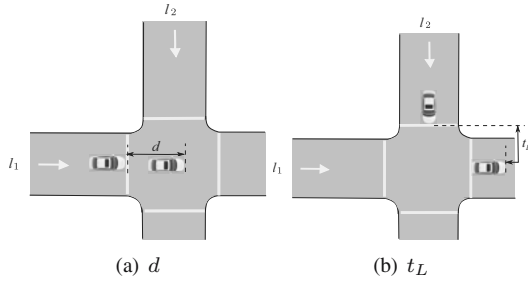


Fig. 3. Safe headways.

Given:  $r_{(i,1)}, \dots, r_{(i,n_i)}, i \in \{1, 2\}$   
 with  $r_{(i,q_i)} < r_{(i,q_i+1)}, q_i \in [1, n_i - 1]$   
 find  $\min\{e_{(1,n_1)}, e_{(2,n_2)}\}$   
 subjects to the constraints (1)  
 $|e_{(i,q_i)} - e_{(j,q_j)}| \geq \begin{cases} d, & i = j \wedge q_i = q_j + 1, \\ t_L, & i \neq j \end{cases}$   
 and  $e_{(i,q_i)} \geq r_{(i,q_i)}$ .

To ensure safety at an intersection, the vehicle passing sequence should be subject to two temporal constraints (Fig. 3):

- (a)  $d$ : the minimum safe headway time between two successive vehicles that move on the same lane;
- (b)  $t_L$ : the minimum time between two successive vehicles that access the conflict zone from conflicting movements.

A safe interval  $t_L$  is entailed between two vehicles from lanes  $l_1$  and  $l_2$ . Here  $t_L$  takes into account the start-up lost time (Kutz, 2004) and the time for the clearance of the conflict zone. The values of  $d$  and  $t_L$  depend on several parameters such as pavement surface conditions, profiles of the drivers, and weather conditions. One way to accurately estimate the values is to refer to empirical observations. The literature is rich in terms of values and formulas with adjustment factors for approximating these parameters (Bertolazzi et al., 2009; Chen et al., 2010). Since the scope of this

Table 1. Adopted notation.

Notation	Meaning
$n_i$	number of vehicles present on the lane $l_i$
$(i, q_i)$	$q_i$ -th arriving vehicle on the lane $l_i$
$r_{(i,q_i)}$	theoretical arrival time of vehicle $(i, q_i)$
$e_{(i,q_i)}$	time when vehicle $(i, q_i)$ accesses the conflict zone
$d$	safe headway respected by vehicles on the same lane
$t_L$	safe headway respected by vehicles of conflicting movements

Table 2. Input of problem (second).

Estimated arrival times		$d$	$t_L$
$l_1$	$r_{(1,1)} = 0$	$r_{(1,2)} = 7$	2
$l_2$	$r_{(2,1)} = 4$	$r_{(2,2)} = 7$	

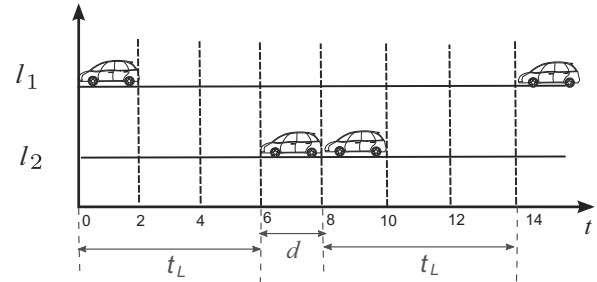


Fig. 4. Optimal vehicle passing sequence of the example given in Table 2.

paper does not cover the estimation of these values, we will use the widely admitted standards (Cohen, 1993).

The objective of control is to minimize the maximum exit time, which is an interesting approach to reduce the vehicle delay. As proved by Papageorgiou et al. (2003), the earlier the vehicles are able to exit the network (by appropriate use of the available control measures), the less time they will have spent in the network.

In order to clearly illustrate the control process, consider an example where four vehicles are present at a two-way intersection. There are two vehicles on  $l_1$  and  $l_2$ , respectively. No vehicles have yet the right-of-way. The infrastructure provides the data about the vehicles. Accordingly the theoretical arrival times are estimated. Table 2 summarizes all the required data to compute the optimal passing sequence. Figure 4 presents an optimal solution with the minimal exit time, i.e., 14 seconds.

**3.2. Complexity of the control problem.** It is noted that one of the major obstacles of the control is the problem complexity since control considers each vehicle individually. The following proposition gives us the number of all feasible sequences for an intersection with  $L$  lanes.

**Proposition 1.** An intersection of  $L$  lanes admits exactly

$$\frac{(\sum_{i=1}^L n_i)!}{\prod_{i=1}^L (n_i!)} \tag{2}$$

ordered sequences of vehicles.

*Proof.* There are  $(\sum_{i=1}^L n_i)!$  possible sequences. On each lane, the  $n_i$  vehicles are stringently ordered (for the sake of safety, vehicle-overtake is forbidden after they have crossed the position marker 1). Hence, we conclude that the number of all feasible sequences equals (2). ■

An optimal solution can be found by an exhaustive search. However, this applies only to scenario with a small number of vehicles and lanes. Since the description of (1) renders the control a combinational optimization problem, we can make use of some specialized theories to solve this problem in polynomial time. Potts and Kovalyov (2000) give a dynamic programming algorithm to solve the family scheduling problem of a single machine. The original problem is decomposed into sub-problems recursively until the sub-problems can be dealt with. In the next section, we will use this decomposition and propose a novel algorithm which is able to integrate estimated arrival times of vehicles.

#### 4. Control strategy

Dynamic programming is a method for solving complex problems by recursively breaking them down into simpler sub-problems (Bellman, 1957). It is applicable to an optimization problem which has two key ingredients (Thomas *et al.*, 2009):

- optimal sub-structure,
- overlapping sub-problems.

A problem exhibits an optimal sub-structure if an optimal solution to the problem contains within it optimal solutions to sub-problems. This property will produce the optimal solution constructed from optimal solutions to sub-problems. The overlapping sub-problems mean that the space of sub-problems is typically polynomial in the input size that a recursive algorithm for the problem solves the same sub-problems over and over, rather than generating new sub-problems. Hence, dynamic programming algorithms solve each sub-problem once and then store the solution in a table where it can be looked up when needed.

In the following, we will use the dynamic programming algorithm to find the optimal solution of the problem (1). To introduce the algorithm, we consider the problem of control at an intersection of two lanes (Fig. 1). It is noted that the proposed algorithm is adaptive to any layout of intersection.

**4.1. Traffic control algorithm.** Let  $[n_1, n_2]$  denote the problem (1). If  $n_1 > 0$  and  $n_2 > 0$ , the last vehicle in the sequence is either from  $l_1$  or  $l_2$ . We use  $[n_1, n_2, 1]$  and  $[n_1, n_2, 2]$  to denote these two possibilities, where  $[n_1, n_2, i]$  ( $i \in \{1, 2\}$ ) indicates the last vehicle in the sequence is from  $l_i$ .

For  $[n_1, n_2, 1]$ , before the last vehicle  $(1, n_1)$ , the  $(n_1 + n_2 - 1)$ -th vehicle is either from  $l_1$  or  $l_2$ . Hence, the vehicle sequence of  $[n_1, n_2, 1]$  is constructed either from sub-sequence of  $[n_1 - 1, n_2, 1]$  or that of  $[n_1 - 1, n_2, 2]$ . In other words, the problem  $[n_1, n_2, 1]$  can be

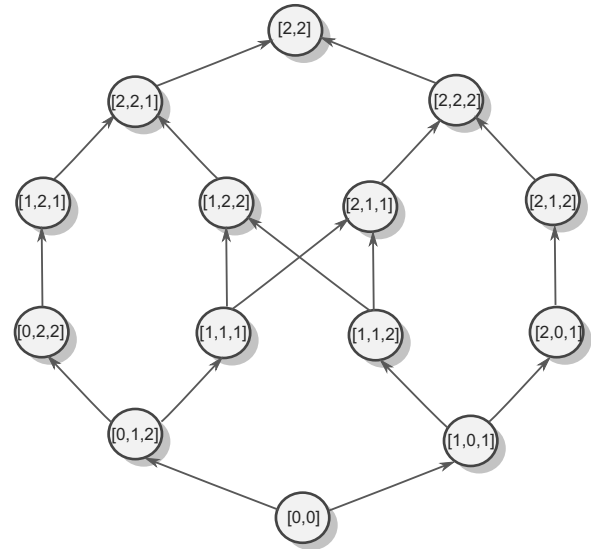


Fig. 5. Decomposition of  $[2,2]$ .

decomposed into two sub-problems  $[n_1 - 1, n_2, 1]$  and  $[n_1 - 1, n_2, 2]$ . Using symmetric reasoning,  $[n_1, n_2, 2]$  results in two sub-problems  $[n_1, n_2 - 1, 1]$  and  $[n_1, n_2 - 1, 2]$ . Each resulting sub-problem can be decomposed into two sub-problems in the same way after removing the last vehicle in the sequence. In general, the decomposition of  $[q_1, q_2, i]$  with  $q_i \leq n_i$  produces two sub-problems  $[m_1, m_2, 1]$  and  $[m_1, m_2, 2]$ , where

$$(m_1, m_2) = \begin{cases} (q_1 - 1, q_2) & \text{if } i = 1 \wedge q_1 \geq 1, \\ (q_1, q_2 - 1) & \text{if } i = 2 \wedge q_2 \geq 1. \end{cases} \quad (3)$$

The relation (3) shows that the decomposition of a problem brings at most two sub-problems by removing the last vehicle in the sequence. The process of decomposition can continue until it reaches the problem  $[0, 0]$ , which denotes that all vehicles are removed. By this decomposition, we can build a directed graph where each sub-problem is linked to its original problem. Figure 5 presents the decomposition of an instance  $[2, 2]$ .

Let  $G_{n_1, n_2} = (V, A)$  be a directed graph that results from the decomposition of  $[n_1, n_2]$ , where a node  $v \in V$  is a sub-problem and an arrow  $a \in A$  is a decision taken from one sub-problem to its original problem. In the following, we call the one relating to a sub-problem “child” and its original problem “parent”.

Let  $P_{n_1, n_2}$  be the set of paths from  $[0, 0]$  to  $[n_1, n_2]$ , and  $S_{n_1, n_2}$  be the set of all possible ordered sequences of vehicles. The cardinal of  $S_{n_1, n_2}$  is given by (2). It is noted that there exists an isomorphism between  $S_{n_1, n_2}$  and  $P_{n_1, n_2}$ . Table 3 gives an example of mapping between  $S_{2,2}$  and  $P_{2,2}$ .

According to the definition of an arrow  $a$ , the weight

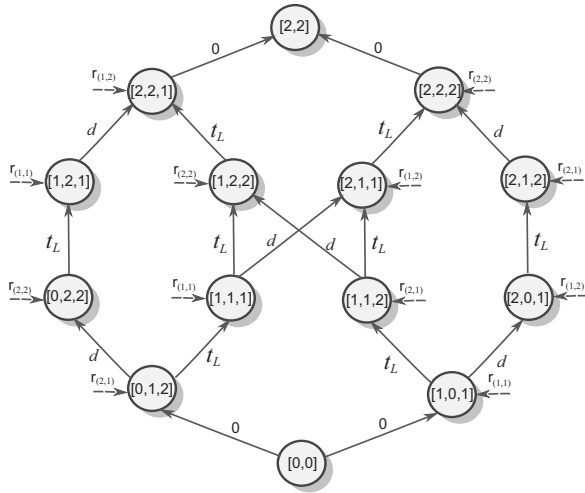


Fig. 6. Weighted graph  $G_{2,2}$ .

Table 3. Mapping between  $S_{2,2}$  and  $P_{2,2}$ : 1: the first arriving vehicle on  $l_1$ , i.e., (1,1), 2: the first arriving vehicle on  $l_2$ , i.e., (2,1), 3: the second arriving vehicle on  $l_1$ , i.e., (1,2), 4: the second arriving vehicle on  $l_2$ , i.e., (2,2).

$S_{2,2}$	$P_{2,2}$
1 > 2 > 4 > 3	[1,0,1]-[1,1,2]-[1,2,2]-[2,2,1]
1 > 2 > 3 > 4	[1,0,1]-[1,1,2]-[2,1,1]-[2,2,2]
1 > 3 > 2 > 4	[1,0,1]-[2,0,1]-[2,1,2]-[2,2,2]
2 > 4 > 1 > 3	[0,1,2]-[0,2,2]-[1,2,1]-[2,2,1]
2 > 1 > 4 > 3	[0,1,2]-[1,1,1]-[1,2,2]-[2,2,1]
2 > 1 > 3 > 4	[0,1,2]-[1,1,1]-[2,1,1]-[2,2,2]

of  $a$  is given by

$$w(a) = \begin{cases} d & \text{if } a \text{ connects } [m_1, m_2, i] \text{ to } [q_1, q_2, i], \\ t_L & \text{if } a \text{ connects } [m_1, m_2, i] \text{ to } [q_1, q_2, j], \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

with  $i \neq j$ . More precisely,  $w$  assigns weights to arrows in order to denote the safe headway between the last vehicle in the sequence of the sub-problem and the inserted vehicle. By assigning weights to arrows, it is possible to use the graph to compute the required time for freeing all vehicles. This period of time is obtained by computing the length of the corresponding path from  $[0, 0]$  to  $[n_1, n_2]$ . The length of the sub-path from  $[0, 0]$  to  $[q_1, q_2, i]$  corresponds to the exit time  $e_{(i, q_i)}$  of the last vehicle  $(i, q_i)$  in the sub-sequence. However, since a vehicle cannot access the conflict zone before the minimal value of the estimated arrival time,  $r_{(i, q_i)}$  is associated to each problem  $[q_1, q_2, i]$  to restrict the length of the path (Fig.6).

The calculation of the length of a path from  $[0, 0]$  to  $[n_1, n_2, i]$  ( $i \in \{1, 2\}$ ) is given by the following recursive

formula:

$$W(v) = \max(W(v') + w(a), r_v), \quad (5)$$

$$W([0, 0]) = 0,$$

where  $v, v' \in V$ ,  $v'$  is a child node of  $v$ ,  $a$  connects  $v$  and  $v'$ ,  $W(v)$  is the length of the path from  $[0, 0]$  to  $v$  and  $r_v$  is the estimated arrival time associated to  $v$ . One can note that  $W([n_1, n_2, i])$  gives the theoretical time for freeing all vehicles. The shortest path from  $[0, 0]$  to  $[n_1, n_2]$  corresponds to the optimal solution to the problem (1).

In the following, we will use dynamic programming to find the shortest path. Before introducing the algorithm, we will prove that the problem exhibits the two key ingredients.

**Theorem 1.** (Optimal sub-structure) *The optimal solution of  $[n_1, n_2, i]$  ( $i \in \{1, 2\}$ ) is calculated by the recursive formula*

$$W^*(v) = \max_{v' \in N_v} (W^*(v') + w(a), r_v), \quad (6)$$

where  $W^*(v)$  is the length of the shortest path from  $[0, 0]$  to  $v$ ,  $N_v$  is the set of children nodes that are directly connected to  $v$  and  $a$  is the arrow connecting  $v'$  to  $v$ .

*Proof.* Let  $P(v') = \{p_1(v'), \dots, p_m(v')\}$  be the set of all possible paths to reach a node  $v'$  from  $[0, 0]$ ,  $W_i(v')$  be the length of the path  $p_i(v')$ ,  $i \in [1, m]$  and  $W_i(v)$  be the length of the path from  $[0, 0]$  to  $v$  that covers the path  $p_i(v')$ .

To prove Theorem 1, it is sufficient to prove that if  $W_k(v') \geq W_l(v')$  then  $W_k(v) \geq W_l(v)$ , where  $k, l \in [1, m]$ .

From Eqn. (5), the recurrence (6) will always be true due to the following relation:

$$W_k(v) - W_l(v) = \begin{cases} W_k(v') - W_l(v') & \text{if } r_v \leq W_l(v') + w(a), \\ W_k(v') + w(a) - r_v & \text{if } W_l(v') + w(a) < r_v < W_k(v') + w(a), \\ 0 & \text{otherwise.} \end{cases}$$

Indeed, if  $W_k(v') - W_l(v') \geq 0$  then  $W_k(v) - W_l(v) \geq 0$ , where  $k, l \in [1, m]$ . ■

Theorem 1 proves that the decomposition of problem makes the problem exhibit an optimal sub-structure. It can be easily proved that the space of sub-problems is polynomial in the input size since the number of nodes  $[q_1, q_2, i]$  is bounded by  $\Theta(n^2)$ . Hence, the dynamic programming algorithm can be used to solve the problem (1).

From Theorem 1, the optimal solution of a problem  $[q_1, q_2, i]$ ,  $q_i \in [1, n_i]$ ,  $i \in \{1, 2\}$  is recursively calculated

in terms of the optimal solutions to sub-problems. Algorithm 1 calculates the values of  $W^*([q_1, q_2, i])$ . Table  $W^*$  records the optimal solution of all sub-problems. When a sub-problem is encountered again, simply look it up in the table. To help us keep track of how to construct an optimal solution, let  $L^*([q_1, q_2, i])$  be the lane number,  $L^*([q_1, q_2, i]) = 1$  or  $2$ . For a node  $[q_1, q_2, 1]$ ,  $L^*([q_1, q_2, 1])$  indicates that the  $(q_1 + q_2 - 1)$ -th vehicle in the optimal sequence is from a given lane.

The minimum exit time of  $[n_1, n_2]$  is given by

$$W^*([n_1, n_2]) = \min\{W^*([n_1, n_2, 1]), W^*([n_1, n_2, 2])\}. \quad (7)$$

Having computed the values of  $W^*([q_1, q_2, i])$  and  $L^*([q_1, q_2, i])$ , we need to construct the vehicle passing sequence by tracing the shortest path. Algorithm 2 finds out the optimal vehicle passing sequence.

Now, we will discuss the computational cost of the algorithm. The following proposition gives the answer.

**Proposition 2.** *The time complexity of the proposed algorithm is  $\mathcal{O}(n^2)$  and the space complexity is  $\Theta(n^2)$ , with  $n = n_1 + n_2$ .*

*Proof.* The time complexity of the algorithm depends on the product of two factors: the overall number of subproblems and how many choices we have for each subproblem. On the one hand, we have overall  $\Theta(n^2)$  sub-problems; on the other, we have at most two choices to examine each sub-problem. Hence, the time complexity is  $\mathcal{O}(n^2)$ .

For the space complexity, since the algorithm requires the  $\Theta(n^2)$  space to store the optimal values of sub-problems, it is equal to  $\Theta(n^2)$ . ■

**4.2. Example.** Figure 7 illustrates how the algorithm calculates sequences through the example presented in Table 2. The problem  $[2, 2]$  is recursively decomposed into simpler sub-problems. Weights are assigned to arcs, in order to take into account the minimum safe headway time ( $d = 2$  s) and the lost time ( $t_L = 6$  s). Besides, the estimated arrival time is associated to each node of the graph. The value at the top of each node gives the length of the shortest path from  $[0, 0]$ . This value is computed by using Algorithm 1. The time required for freeing the intersection is about 14 s. Thick dark arrows and dark grey nodes show the optimal sequence obtained by Algorithm 2. Based on the optimal sequence, the intersection manager first distributes the right-of-way to the first vehicle of  $l_1$ . After this vehicle has successfully crossed the intersection (passes over the position marker 3), it sends the right-of-ways to the two vehicles on the opposite lane  $l_2$ . When the last vehicle passes over the position marker 3, the second vehicle

**Algorithm 1.** Recursive-path  $(q_1, q_2, i)$ .

```

1:  $v = [q_1, q_2, i]$ 
2: if  $i = 1$  then
3:   if  $q_1 = 0$  then
4:      $W^*(v) = \infty$  {Illegal decomposition}
5:   else if  $q_1 = 1$  and  $q_2 = 0$  then
6:      $W^*(v) = r_v, L^*(v) = 1$ 
7:   else
8:      $v_1 = [q_1 - 1, q_2, 1]$ 
9:      $W^*(v_1) = \text{RECURSIVE-PATH}(q_1 - 1, q_2, 1)$ 
10:     $Temp1 = \max\{W^*(v_1) + a(v_1, v), r_v\}$ 
11:     $v_2 = [q_1 - 1, q_2, 2]$ 
12:     $W^*(v_2) = \text{RECURSIVE-PATH}(q_1 - 1, q_2, 2)$ 
13:     $Temp2 = \max\{W^*(v_2) + a(v_2, v), r_v\}$ 
14:    if  $Temp1 \leq Temp2$  then
15:       $W^*(v) = Temp1, L^*(v) = 1$ 
16:    else
17:       $W^*(v) = Temp2, L^*(v) = 2$ 
18:    end if
19:  end if
20: else if  $q_2 = 0$  then
21:    $W^*(v) = \infty$  {Illegal decomposition}
22: else if  $q_1 = 0$  and  $q_2 = 1$  then
23:    $W^*(v) = r_v, L^*(v) = 2$ 
24: else
25:    $v_1 = [q_1, q_2 - 1, 1]$ 
26:    $W^*(v_1) = \text{RECURSIVE-PATH}(q_1, q_2 - 1, 1)$ 
27:    $Temp1 = \max\{W^*(v_1) + a(v_1, v), r_v\}$ 
28:    $v_2 = [q_1, q_2 - 1, 2]$ 
29:    $W^*(v_2) = \text{RECURSIVE-PATH}(q_1, q_2 - 1, 2)$ 
30:    $Temp2 = \max\{W^*(v_2) + a(v_2, v), r_v\}$ 
31:   if  $Temp1 \leq Temp2$  then
32:      $W^*(v) = Temp1, L^*(v) = 1$ 
33:   else
34:      $W^*(v) = Temp2, L^*(v) = 2$ 
35:   end if
36: end if
37: return  $W^*(v)$ 

```

Table 4. Characteristic of vehicles and the intersection.

Vehicles		Intersection	
Length	4-5 m	Length of the storage zone	105 m
Reaction time	1 s	Lane width	3.5 m
Acceleration	2 m/s <sup>2</sup>	Weather condition	Dry road
Deceleration	-2 m/s <sup>2</sup>	Coefficient of friction	0.8
Maximal speed	15 m/s	Saturation flow rate	0.5 veh/s

**Algorithm 2.** Construct-sequence ( $W^*, L^*$ ).

```

1:  $q_1 = n_1$ 
2:  $q_2 = n_2$ 
3: if  $W^*([q_1, q_2, 1]) \leq W^*([q_1, q_2, 2])$  then
4:   The  $(q_1 + q_2)$ -th vehicle is :  $(1, q_1)$ 
5:    $i = L^*([q_1, q_2, 1])$ 
6:    $q_1 = q_1 - 1$ 
7: else
8:   The  $(q_1 + q_2)$ -th vehicle is :  $(2, q_2)$ 
9:    $i = L^*([q_1, q_2, 2])$ 
10:   $q_2 = q_2 - 1$ 
11: end if
12: while  $(q_1 + q_2) \geq 1$  do
13:   if  $i = 1$  then
14:     The  $(q_1 + q_2)$ -th vehicle is :  $(1, q_1)$ 
15:      $i = L^*([q_1, q_2, 1])$ 
16:      $q_1 = q_1 - 1$ 
17:   else
18:     The  $(q_1 + q_2)$ -th vehicle is :  $(2, q_2)$ 
19:      $i = L^*([q_1, q_2, 2])$ 
20:      $q_2 = q_2 - 1$ 
21:   end if
22: end while

```

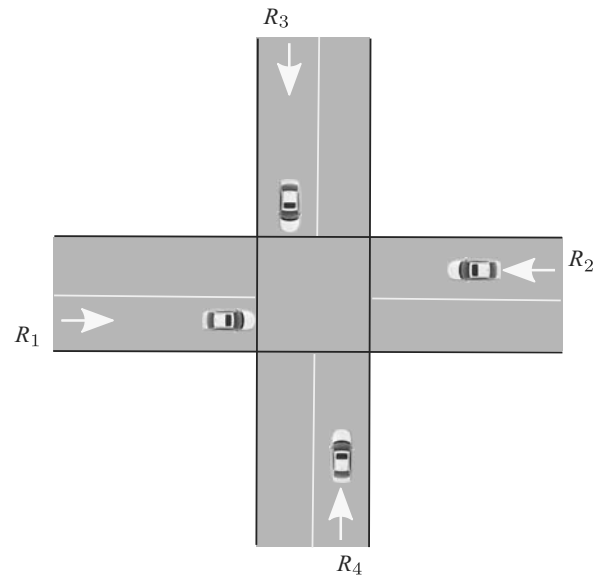


Fig. 9. 4-way intersection.

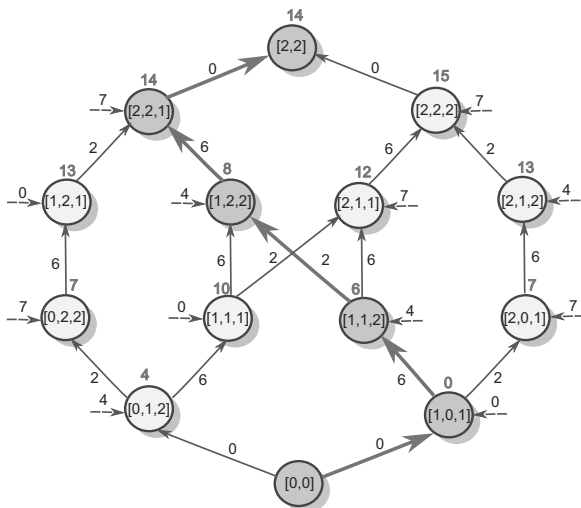


Fig. 7. Graph of decomposition of the example presented in Table 2.

of  $l_1$  receives the right-of-way and can then exit the intersection.

To validate the optimal solution, we have simulated the microscopic behavior of vehicles under all feasible sequences. The Gipps model (Gipps, 1981; 1986) is used as the simulation tool to describe the microscopic car-following behavior. The parameters of vehicles and the intersection are presented in Table 4. The behaviors of vehicles under all possible sequences are shown in Fig. 8. One can note from this example that, although the

calculation of the exit time is approximate, the optimal sequence can be found by the algorithm. Beside, the exit time can be reduced if the position marker 3 confirms the departure of the last admitted vehicle in practical applications. In other words, the right-of-ways are distributed more rapidly than expected because the system distributes the right-of-way according to data collected at the position marker 3.

**4.3. Extensions.** The proposed algorithm can be easily applied to isolated intersections with different layouts. Before applying the decomposition, vehicles are partitioned into different movements. In the same movement, since vehicles can pass through the intersection without conflict, only safe headway  $d$  is required for two successive vehicles on the same lane. For vehicles belonging to conflicting movements, the safe headway  $t_L$  is entailed to ensure the safety at the conflicting zone. After partitioning vehicles into movements, we can apply the same algorithm to find a vehicle passing sequence.

For example, let us consider a typical intersection with four ways as illustrated in Fig. 9. There are two conflicting movements, i.e., vehicles from roads  $R_1, R_2$  and vehicles from roads  $R_3, R_4$ . Let  $n_1$  and  $n_2$  denote the numbers of vehicles belonging to the two conflicting movements. After the partition, the problem becomes  $[n_1, n_2]$ , which can be decomposed in the same way as introduced before.

In general, if the number of the conflicting movements  $c$  is greater than 2, the problem  $[n_1, n_2, \dots, n_c]$  is decomposed into  $c$  sub-problems:  $[n_1, n_2, \dots, n_c, 1], \dots, [n_1, n_2, \dots, n_c, c]$ , where  $n_i$



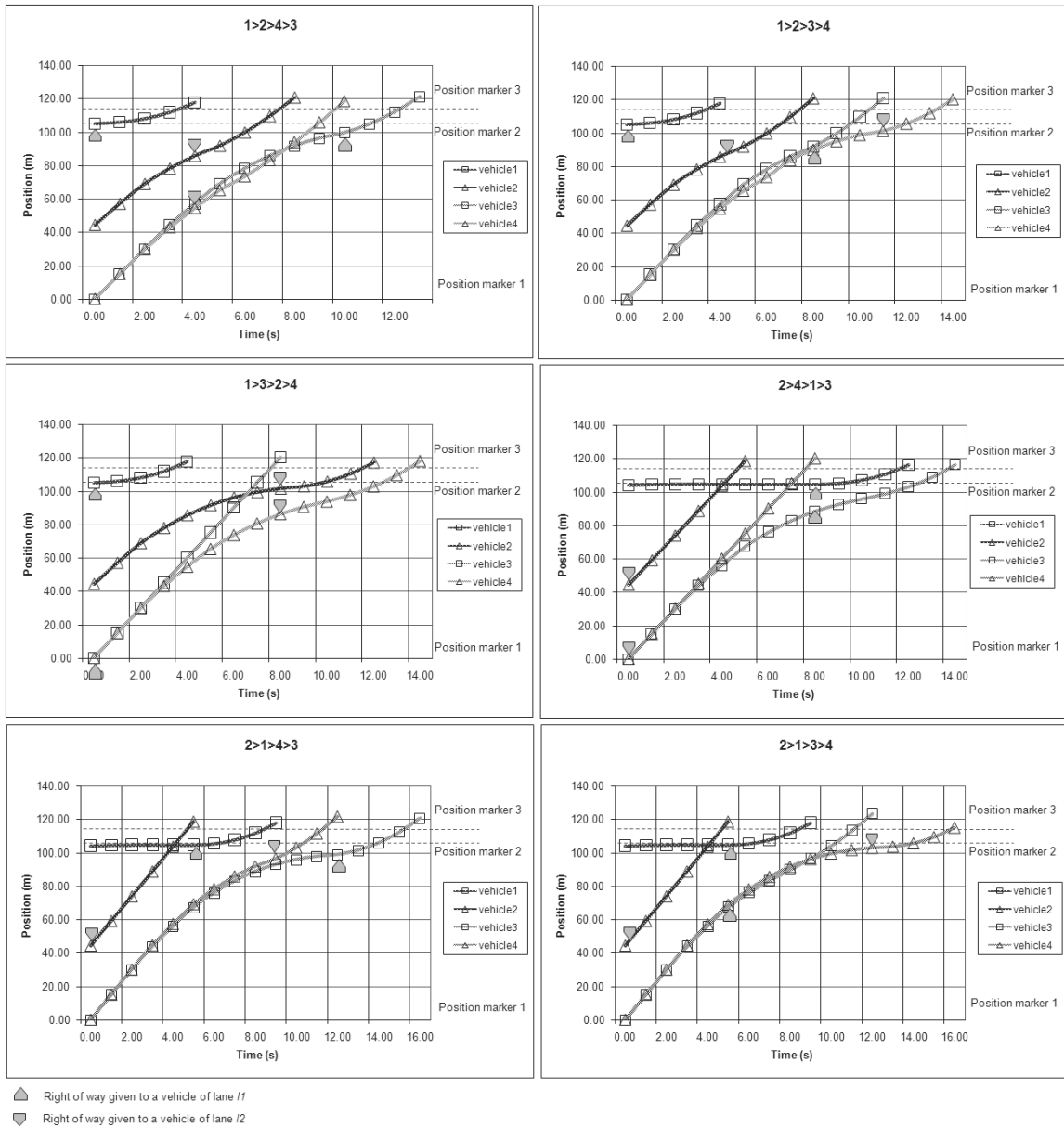


Fig. 8. Simulation of vehicle microscopic behavior under all feasible passing sequences (a number is assigned to each vehicle in order to distinguish vehicles, as shown in Table 3).

denotes the number of vehicles belonging to the conflicting movement  $i$  (here we have  $i \in [1, c]$ ). Each sub-problem can be decomposed into  $c$  subproblems after removing the last vehicle in the sequence. For example,  $[n_1, n_2, \dots, n_c, 1]$  results in  $[n_1 - 1, n_2, \dots, n_c, 1], \dots, [n_1 - 1, n_2, \dots, n_c, c]$ . Each resulting sub-problem can be decomposed into at most  $c$  sub-problems in the same way until the decomposition reaches

$$\underbrace{[0, \dots, 0]}_c$$

Figure 10 shows the decomposition of an instance [2,3,2].

It is easy to prove that the problem  $[n_1, n_2, \dots, n_c]$  still exhibits two key ingredients: optimal sub-structure and overlapping sub-problems, as shown in Section 4.1. The optimal solution can be calculated by Eqn. (6). The time complexity of the algorithm is  $\mathcal{O}(n^c)$  and the space complexity is  $\Theta(n^c)$ .

### 5. Efficiency study of CVAS

In this section, the traffic control effectiveness of CVAS is evaluated. The simulation is implemented in a traffic simulation environment where the car-following model

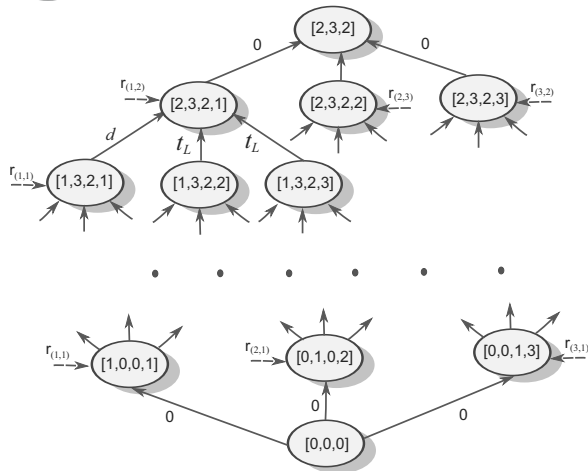


Fig. 10. Decomposition of [2,3,2].

(Gipps model) is used to simulate the behaviour of vehicles. The inputs of the simulation are given in Table 4. The intersection under consideration is an intersection of four lanes presented in Fig. 9.

The effectiveness of CVAS is compared with the following four traffic control systems:

1. Adaptive control system: a traditional traffic control method which is proven efficient in the current traffic system. Here, the method presented by Fang and Elefteriadou (2006) is used for comparison.
2. New traffic control system based on wireless communication proposed by Gradinescu *et al.* (2007): the controller keeps tracking the vehicles by the wireless communication. Thus the estimation of the traffic volume is more precise. The traffic light switching scheme is updated during each cycle. The famous Webster formula (Webster, 1958) is applied to calculate the cycle length and the green time according to the estimated demand.
3. New traffic control system proposed by Wunderlich *et al.* (2008): the control strategy is based on the Little formula (Little, 1961). The average delay experienced by vehicles in the network is directly proportional to the average queue size. The controller tries to minimize the queue size by giving the right-of-way to the approach with the maximal weighted queue length. The queue lengths are measured precisely by the advanced wireless communication and positioning technologies.
4. Autonomous intersection management proposed by Dresner and Stone (2004; 2006). This comparison aims to know whether the optimization of passing sequence will improve traffic efficiency in heavy traffic load. Hence, we compare our system with the AIM based on the FCFS control policy.

It is to be noted that although the new traffic systems (2) and (3) adopt the advanced technologies of wireless communication and of positioning systems, the traffic control is still based on traffic lights. In the following, we use the abbreviations Adaptive, Webster, LQF-MWM, and AIM (FCFS) to represent the above systems, respectively. The performances of all the systems are evaluated by the following criteria:

1. Evacuation time: the time to evacuate all the vehicles present in one hour.
2. Throughput: the ratio between the number of vehicles passed over the position marker 2 and the number of vehicles passed over the position marker 1 during the time of simulation.
3. Mean vehicle delay: this quantity is calculated by the difference between the estimated travel time in the absence of the intersection control and the travel time in the presence of the intersection control.
4. Mean queue length: it is defined as the cumulative number of vehicles stopped before the conflict zone divided by the simulation time.

The simulation runs sixty minutes and the simulation step is 1 s. The results of simulation are presented in Fig. 11. Each point in the figure represents an average of ten runs. The unit of flow rate *pcpspl* denotes passenger cars per second per lane.

First, let us compare CVAS with the traditional traffic control system. The results show that CVAS outperforms the Adaptive system. Besides, its performance is as stable as the system Adaptive. Turning to the comparison with other new traffic control systems, CVAS also presents the best performance. We observe that Webster exhibits the poorest performance since its inherent limitation of the control strategy: although the green time is updated every cycle, it cannot react quickly to the traffic. For LQF-MWM, it shows good performance at a low traffic load ( $\leq 0.35$  pcpspl); however, as the traffic load increases, its performance is deteriorated. That is due to the frequent change of right-of-way when each approach accumulates too many vehicles waiting for the right-of-way. The same conclusion can be drawn from AIM (FCFS). When the traffic load is low, i.e.,  $\leq 0.35$  pcpspl, it has the compatible performance with CVAS; as the traffic load increases, it shows a sharp decline in its performance. The comparison with AIM (FCFS) proves that a proper arrangement of the vehicle passing sequence will improve the traffic efficiency.

## 6. Discussion

The simulations show that a proper arrangement of vehicles passing sequence in the proposed system

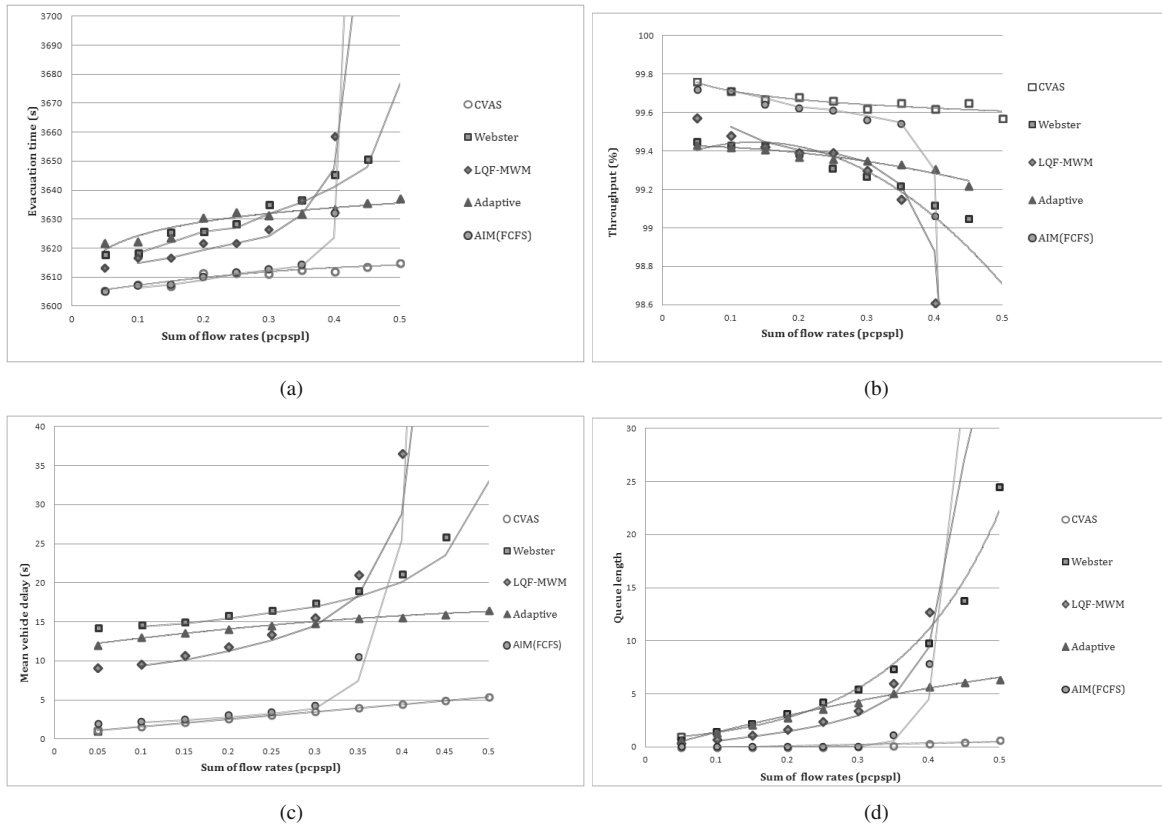


Fig. 11. Simulation results: evacuation time (s: second) (a), throughput (b), mean vehicle delay (s: second) (c), mean queue length (d).

has great capacity to improve the traffic efficiency, whereas there are several issues that deserve further investigations. The most important issue is the feasibility of the system. To study this, we realized the CVAS in a simple intersection with human driven vehicles. The right-of-way is notified to the driver by onboard signalization. The video of the simulation can be found at <http://www.youtube.com/watch?v=D6s8tXlGSLA>. During the tests, we did not observe any collision. This experiment proves the feasibility of the protocol of CVAS that we present here, at least for industrial applications.

In addition, the main obstacle of the system is that all vehicles must be equipped. We draw the reader's attention to the fact that AIM can be forthcoming since Europe (France, Germany, Switzerland, etc.) and the United States have launched clean car sharing programmes to reduce pollution in urban areas. Some programmes (e.g., PRAXITELE, Liselec, and Autolib Projects in France, City-Car in Switzerland, IntelliShare in the USA) have planned to limit the access of some zones of the city to green car sharing. In other words, some intersections of the urban area will be exclusively shared by fleets of equipped vehicles where vehicles negotiate their access to intersection. Hence, CVAS has a great potential to be applied in car sharing zones.

## 7. Conclusion

This paper introduces a theoretical system, cooperative vehicle actuation signalization, to manage the traffic at isolated intersections. This new traffic control system is capable of managing vehicles individually based on the technology of cooperative driving. To propose a proper vehicle passing sequence, the control of a system is rendered through a combinational optimization problem. A dynamic programming algorithm is proposed to solve the control problem in a polynomial-time complexity for an isolated intersection. Simulation results demonstrate significant improvements over current intersections. Besides, this system has the potential to be applied to any isolated intersection with more complicated layout. A real test is implemented to study the feasibility of the system. In the future, we will enrich the results and apply them to connected intersections.

## References

Abbas-Turki, A., Ahmane, M., Gao, F., Wu, J., El-Moudni, A. and Miraoui, A. (2012). On the conflict with dioid algebra: Autonomous intersection management, *Proceedings of the 7th IEEE Conference on Industrial Electronics and Applications, ICIEA'12, Singapore*, (CD-ROM).

- Allsop, R.E. (1971). Sigset: A computer program for calculating traffic capacity of signal-controlled road junctions, *Traffic Engineering & Control* (12): 58–60.
- Allsop, R.E. (1976). Sigcap: A computer program for assessing the traffic capacity of signal-controlled road junctions, *Traffic Engineering & Control* (17): 338–341.
- Arora, S., Raina, A.K. and Mittal, A.K. (2012). Collision avoidance among AGVs at junctions, *Proceedings of the IEEE the Intelligent Vehicles Symposium, ICIEA'12, Singapore*, pp. 585–589.
- Bellman, R. (1957). *Dynamic Programming*, Princeton University Press, Princeton, NJ.
- Bertolazzi, E., Biral, F., Lio, M.D., Saroldi, A. and Tango, F. (2009). Supporting drivers in keeping safe speed and safe distance: The SASPENCE Subproject within the European Framework Programme 6 Integrating Project PReVENT, *IEEE Transactions on Intelligent Transportation Systems* (99): 1–14.
- Boillot, F., Midenet, S. and Pierrelee, J.C. (2000). Real-life cronos evaluation, *Proceedings of the 10th International Conference on Road Transport Information and Control, London, UK*, (CD-ROM).
- Chen, X. Li, L. and Zhang, Y. (2010). A Markov model for headway/spacing distribution of road traffic, *IEEE Transactions on Intelligent Transportation Systems* 11(4): 773–785.
- Cohen, S. (1993). *Ingénierie du Trafic Routier. Eléments de thorie du trafic et applications*, Presses de l'Ecole National des Ponts et Chaussées, Princeton, NJ.
- David, F., Tsz-Chiu, A., Travis, W., Peter, S. and David, Y. (2012). Automated intersection control: Performance of a future innovation versus current traffic signal control, *Transportation Research Record* (2259): 223–232.
- DiCesare, F., Kulp, P.T., Gile, M. and List, G. (1994). The application of Petri nets to the modeling, analysis and control of intelligent urban traffic networks, in R. Valette (Ed.), *Application and Theory of Petri Nets*, Lecture Notes in Computer Science, Vol. 815, Springer, Berlin/Heidelberg, pp. 2–15.
- Dresner, K. and Stone, P. (2004). Multiagent traffic management: A reservation-based intersection control mechanism, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS-2004, New York, NY, USA*, pp. 530–537.
- Dresner, K. and Stone, P. (2006). Traffic intersections of the future, *Proceedings of the 21st National Conference on Artificial Intelligence, AAAI-2006, Boston, MA, USA*, pp. 1593–1596.
- Dresner, K. and Stone, P. (2008). A multiagent approach to autonomous intersection management, *Journal of Artificial Intelligence Research* 31(13): 591–656.
- Fang, F.C. and Elefteriadou, L. (2006). Development of an optimization methodology for adaptive traffic signal control at diamond interchanges, *Journal of Transportation Engineering* 132(8): 629–637.
- Gartner, N. (1983). OPAC: A demand-responsive strategy for traffic signal control, *Transportation Research Record* (906): 75–81.
- Gipps, P.G. (1981). A behavioural car-following model for computer simulation, *Transportation Research, Part B: Methodological* 15(2): 105–111.
- Gipps, P.G. (1986). MULTSIM: A model for simulating vehicular traffic on multi-lane arterial roads, *Mathematics and Computers in Simulation* 28(4): 291–295.
- Gradinescu, V., Gorgorin, C., Diaconescu, R., Cristea, V. and Iftode, L. (2007). Adaptive traffic lights using car-to-car communication, *Proceedings of the 65th IEEE Vehicular Technology Conference, VTC2007-Spring, Dublin, Ireland*, pp. 21–25.
- Grünewald, M., Rust, C. and Witkowski, U. (2006). Using mini robots for prototyping intersection management of vehicles, *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment AMIRE 2005, Awara-Spa, Fukui, Japan*, pp. 287–292.
- Henry, J.J., Farges, J.L., and Tuffal, J. (1983). The PRODYN real-time traffic algorithm, *Proceedings of the 4th IFAC Symposium on Transportation Systems, Baden Baden, Germany*, pp. 307–312.
- Hunt, P.B. (1982). The scoot on-line traffic signal optimisation technique, *Traffic Engineering & Control* (23): 190–192.
- Kutz, M. (2004). *Handbook of Transportation Engineering*, McGraw-Hill, Boca Raton, FL.
- Lee, J. and Hyung, L. (1999). Distributed and cooperative fuzzy controllers for traffic intersections group, *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews* 29(2): 263–271.
- Li, L. and Wang, F.Y. (2006). Cooperative driving at blind crossings using intervehicle communication, *IEEE Transactions on Vehicular Technology* 55(6): 1712–1724.
- Little, J.D.C. (1961). A proof for the queuing formula:  $L = \lambda w$ , *Operation Research* 9(3): 383–387.
- Matteo, V. and Sascha, O. (2009). Evaluating policies for reservation-based intersection control, *Proceedings of the 14th Portuguese Conference on Artificial Intelligence, EPIA'09, Aveiro, Portugal*, pp. 21–25.
- Mehani, O. and de La Fortelle, A. (2007). Trajectory planning in a crossroads for a fleet of driverless vehicles, *Computer Aided Systems Theory* (4739): 1159–1166.
- Mirchandani, P. and Lucas, D.E. (2001). RHODES-ITMS Tempe field test project: Implementation and field testing of RHODES, a real-time traffic adaptive control system, *Technical Report FHWA-AZ01-447*, Federal Highway Administration, Washington, DC.
- Mirchandani, P. and Head, L. (2001). A real-time traffic signal control system: Architecture, algorithms, and analysis, *Transportation Research, Part C: Emerging Technologies* 9(6): 415–432.
- Papageorgiou, M., Diakaki, P., Kotsialos, D. and Wang, Y.B. (2003). Review of road traffic control strategies, *Proceedings of the IEEE* 91(12): 2043–2067.

- Péter, T. (2012). Modeling nonlinear road traffic networks for junction control, *International Journal of Applied Mathematics and Computer Science* **22**(3): 723–732, DOI: 10.2478/v10006-012-0054-1.
- Potts, C.N. and Kovalyov, M.Y. (2000). Scheduling with batching: A review, *European Journal of Operational Research* **120**(2): 228–249.
- Robertson, D.I. (1969). Transyt method for area traffic control, *Traffic Engineering & Control* (10): 276–281.
- Sakaguchi, T., Uno, A., Kato, and Tsugawa, S. (2000). Cooperative driving of automated vehicles with inter-vehicle communications, *Proceedings of the IEEE Intelligent Vehicles Symposium, IV 2000, Dearborn, MI, USA*, pp. 516–521.
- Sen, S. and Head, K.L. (1997). Controlled optimization of phases at an intersection, *Transportation Science* **31**(1): 5–17.
- Sims, A.G. and Dobinson, K.W. (1980). The Sydney Coordinated Adaptive Traffic (SCAT) system philosophy and benefits, *IEEE Transactions on Vehicular Technology* **29**(2): 130–137.
- Thomas, H.C., Leiserson, C.E., Rivest, R.L. and Stein, C. (2009). *Introduction to Algorithms*, MIT Press, Cambridge, MA.
- Tsugawa, S. (2002). Inter-vehicle communications and their applications to intelligent vehicles: An overview, *Proceedings of the IEEE 2002 Intelligent Vehicle Symposium, Versailles, France*, pp. 564–569.
- Uno, A., Sakaguchi, T. and Tsugawa, S. (1999). A merging control algorithm based on inter-vehicle communication, *Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems 1999, Tokyo, Japan*, pp. 783–787.
- Webster, F.V. (1958). Road research technical paper, *Technical Report 39*, Road Research Laboratory, London.
- Wunderlich, R., Liu, C.B., Elhanany, I. and Urbanik, T. (2008). A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection, *IEEE Transactions on Intelligent Transportation Systems* **9**(3): 536–547.



**Jia Wu** received the Ph.D. degree in automatic control from the University of Technology of Belfort-Montbéliard in 2011. She is currently a lecturer at the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her main research interests include intelligent transportation system, system-engineering, and computational intelligence.



**Abdeljalil Abbas-Turki** received the Ph.D. degree in control and computer science from the University of Franche-Comté, France, in 2003. One year later, he joined the University of Technology of Belfort-Montbéliard as a lecturer. His research interests include discrete event dynamic system and combinatorial optimization applied to urban traffic modeling and control. He is involved in many BRT (bus rapid transit) projects, for studying the future behavior of the traffic through macroscopic and microscopic models. He is also involved in developing cooperative intersection management for encouraging new mobility systems as well as in time-tabling urban rail traffic.



**Florent Perronnet** received the Master's degree in computer science from the University of Franche-Comté, France, in 2011. Since 2012, he has been a Ph.D student at the University of Technology of Belfort-Montbéliard. His research interests include multi-agent simulations, discrete event systems, and wireless systems applied to traffic control. His work deals with the problem of urban congestion, in particular at intersections shared with public transport vehicles. The studied solutions use new technologies among communication systems and positioning systems in order to develop cooperative intersection management. The research field also includes the possible use of autonomous vehicles.

Received: 24 March 2013

Revised: 11 July 2013

Re-revised: 20 August 2013