Appl Intell (2012) 37:207–222
DOI 10.1007/s10489-011-0322-z

Cooperative driving: an ant colony system for autonomous intersection management

Jia Wu · Abdeljalil Abbas-Turki · Abdellah El Moudni

Published online: 12 October 2011

Abstract Autonomous intersection management (AIM) is an innovative concept for directing vehicles through the intersections. AIM assumes that the vehicles negotiate the right-of-way. This assumption makes the problem of the intersection management significantly different from the usually studied ones such as the optimization of the cycle time, splits, and offsets. The main difficulty is to define a strategy that improves the traffic efficiency. Indeed, due to the fact that each vehicle is considered individually, AIM faces a combinatorial optimization problem that needs quick and efficient solutions for a real time application. This paper proposes a strategy that evacuates vehicles as soon as possible for each sequence of vehicle arrivals. The dynamic programming (DP) that gives the optimal solution is shown to be greedy. A combinatorial explosion is observed if the number of lanes rises. After evaluating the time complexity of the DP, the paper proposes an ant colony system (ACS) to solve the control problem for large number of vehicles and lanes. The complete investigation shows that the proposed ACS algorithm is robust and efficient. Experimental results obtained by the simulation of different traffic scenarios show that the AIM based on ACS outperforms the traditional traffic lights and other recent traffic control strategies.

Keywords Ant colony system · Cooperative driving · Autonomous intersection management · Wireless communication

1 Introduction

Over the past decade, wireless technologies with positioning systems have received a great attention [1–5] since they offer a tremendous potential for reinforcing the link between the vehicles (the drivers), the traffic environment, and the control system. A new active research area, cooperative driving, has appeared. Cooperative driving was first introduced in the early 1990s [3] for the lane changing and merging in platoon. Then, it has been further investigated in the fields like the freeway on ramp system [2], guiding vehicle through un-signalized intersection [6, 7], which uses another appellation "Autonomous Intersection Management" (AIM).

More precisely, by means of V2I (Vehicle-to-Infrastructure) and positioning technology, the traffic controller is continuously aware of the number of vehicles, their positions, and their speeds. The traffic control algorithms can be based on new data [8–10], even be completely changed [7, 11–14] for un-signalized intersection. AIM explicitly determines the sequences of vehicles or robots for a real-time application. A sequence is a stringent order of distributing the right-of-way to vehicles or to robots. The right-of-way is sent to the robot or displayed to the driver by means of the onboard signalization. Each vehicle that approaches the intersection is managed separately and waits for receiving its right-of-way before crossing the intersection. Vehicle informs the others (vehicles or infrastructure) about its arrival

Electronic supplementary material The online version of this article (doi:10.1007/s10489-011-0322-z) contains supplementary material, which is available to authorized users.

J. Wu (✉) · A. Abbas-Turki · A. El Moudni
Laboratoire Systèmes et Transports, Université de Technologie de Belfort-Montbéliard, Rue Thierry Mieg, 90010 Belfort cedex, France
e-mail: jia.wu@utbm.fr

A. Abbas-Turki
e-mail: abdeljalil.abbas-turki@utbm.fr

A. El Moudni
e-mail: abdellah.el-moudni@utbm.fr

time, its access time, and its exit time. Other information such as the movement, the braking, and the speed of vehicles could be collected to distribute the right-of-way. AIM requires vehicles equipped with at least a positioning system and a wireless communication system which is able to communicate with the infrastructure.

The AIM is quite different from the traditional traffic control system and presents the following important novelties:

- Unlike the traditional control systems that decide the duration of phases (the green light given to movements) [15] or decide whether or not the phase is still kept for another period of time [16, 17], AIM allows 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 for deciding an optimized sequence.
- It facilitates the deployment of priority policies between vehicles because each vehicle is considered individually.

Most researches of AIM are dedicated to the safety issues, i.e. avoid vehicle collision at intersection [12–14]. Recently, more and more researches address the traffic efficiency—the optimization of the passing sequence [7, 11, 18–20]. It is noted that one of the major difficulties of the latter is the traffic control complexity since each vehicle is considered individually. Here we enumerate two typical works to the best of our knowledge. One notable work is the one presented in [7]. The authors have enumerated all feasible schedules (all allowable passing sequences of vehicles). Then trajectory planning algorithms were used to choose the most efficient schedule that guides the vehicles to cross the intersection as soon as possible. The idea of this work is quite novel since it not only considers the safety of driving, but also takes into account the traffic efficiency. However, as the author pointed out that the main problem of this method is the complexity of cooperative driving planning: it exponentially increases with the number of vehicles and lanes. In [19] and [20], the authors have proposed a dynamic programming and a branch and bound algorithm, respectively. Both algorithms optimize the sequence without enumerating all possible solutions. However, the complexity exponentially increases with the number of lanes.

Another notable work is realized by Kurt and Peter [11, 21]. They have introduced an intelligent agent, called intersection manager, which assigns reservations of space and time to each autonomous vehicle, operated by a driver agent. The intersection control policy is based on FCFS (First Come First Serve) policy with a low computational cost. More recently (2010), a mixed reality platform on which an autonomous vehicle can interact with multiple virtual vehicles is implemented [22], in order to show feasibility of AIM. The experience shows that AIM outperforms traffic signals and stop signs in low traffic load ($\leq$140 vehicles/hour/lane). Nevertheless, a query is raised about this simple control policy: Is it still efficient to control traffic in heavy traffic load?

All the above researches show that the proper arrangement of passing sequence will improve the traffic efficiency at intersections. However, the solution space of sorting vehicles is very huge. This paper proposes a heuristic based on an ant colony system algorithm, for quickly obtaining efficient sequences. The algorithm is compared to AIM based on FCFS control policy and to three other recent traffic control strategies that use traffic lights. The first one is an adaptive traffic light control that uses inductive loops for evaluating the traffic volume. The two other strategies use wireless communication and positioning technologies to optimize the green light duration.

This paper is organized as follows: First, a basic configuration of AIM system is introduced. Then the system behavior is analyzed. Based on the analysis, the traffic control at intersection is formulated as a combinational optimization problem. The problem complexity is discussed. Although an algorithm of dynamic programming can find an optimal solution [19], it cannot fulfill the real-time requirement as the problem complexity grows. Therefore, we draw an analogy between the dealt problem and a traveling salesman problem (TSP), with respect to all the constraints. Then, an ant colony system algorithm that has efficiently solved TSP [23] is adopted to solve the problem more quickly with ensuring the accuracy. The main idea of this algorithm works to search an approximate solution of combinatorial optimization problems by a set of ants which cooperate through exchanging information via pheromone. Finally, the paper discusses the results of simulations before concluding.

2 System overview

The AIM system gets the following important ambitions:

- The accesses to the intersection are ordered (sequence) 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.
- Each vehicle as well as the infrastructure instantaneously participates to the decision making process.
- Regardless technological and practical limitations, such as the road capacity or the restriction of the wireless communication, the traffic management at intersections is able to consider an unbounded number of vehicles without a preprogrammed schedule.

In the following, we will discuss the architecture of the studied protocol of AIM through the negotiation architecture and the signalization strategy.

2.1 Negotiation network

To the best of our knowledge, there are two possible architectures of the negotiation network. The first one is decentralized. Each vehicle directly negotiates with the other vehicles to obtain the right-of-way. The second one is centralized. The intersection manager receives the requests of vehicles and sends the right-of-way accordingly.

The decentralized prototype presented by Grünewald, Rust and Witkowski [24] has shown that collisions would occur if a vehicle had missed some messages broadcasted by other vehicles. This is due to the fact that each robot autonomously decides the right-of-way for itself according to its presence list which can be inconsistent. A centralized architecture overcomes this problem, if the decision is taken by a referee (intersection manager) that owns a common presence list. This architecture contributes to avoid some collision, even if the list is incomplete. Indeed, all missed vehicles are not able to traverse the intersection until they are included into the presence list of the intersection manager.

However, this rule raises another problem. Deadlock is possible if there are no supplemental positioning technologies (e.g., position markers) in order to make the intersection manager aware of the order of approaching vehicles. As proposed in [24], robots detect position markers (landmarks) at particular points of the infrastructure, in order to avoid deadlocks.

2.2 Signalization strategy

There are at least two strategies that allow a vehicle to go through the intersection. The simpler one informs a particular vehicle that the conflict zone is free. A more sophisticated one specifies the time and the velocity of the reservation given to each vehicle. The second strategy deserves more attention. This strategy states that the vehicle loses the reservation if it determines that it will not arrive at the intersection at the time and/or the velocity specified in the reservation. In this case, the vehicle may be unable to stop in front of the conflict zone due to the physical parameters of the situation. Hence, a vehicle must estimate whether it will respect the reservation specification before a given time and position. Otherwise, collision is possible because vehicle is not as controllable as one can expect [22]. This safety issue is well-known in the traffic engineering literature (see dilemma zone [25]). One solution to this problem is to consider a buffer time, as stated in [22].

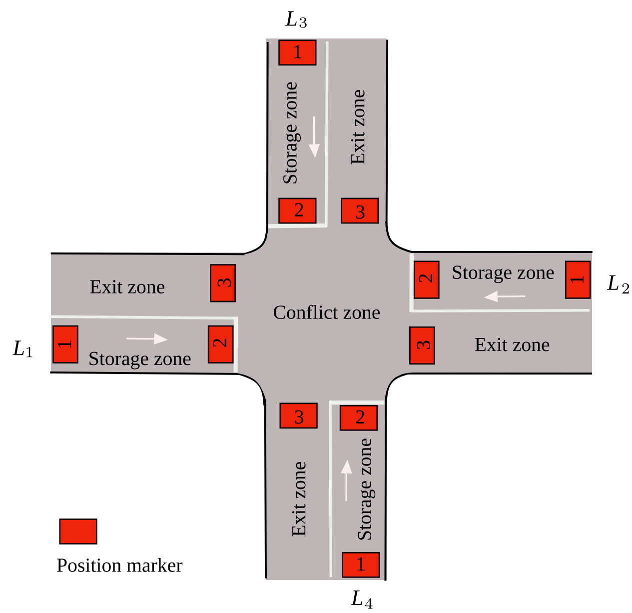


Fig. 1 Elementary configuration of the infrastructure

2.3 Protocol of AIM

Based on the above description, we propose a basic protocol of AIM system. The protocol is a centralized architecture of discrete point positioning and signaling technology. The main objective of the first technology is to guarantee that an authenticated vehicle is currently passing over a particular point of the infrastructure. Consider a 4-lane intersection as presented in Fig. 1, where each lane is divided into two zones by the conflict zone: the storage zone and the exit zone. Position markers are placed at the entry of each zone. Thus, there are three position markers on each lane. This configuration could confirm the arrival of a vehicle (by the position marker 1), its access to the conflict zone (by the position marker 2), and its departure (by the position marker 3).

The signaling technology uses existing DSRC (Dedicated Short Range Communications) [26] to transmit and receive text-based signals. The main objective of this technology is to successfully transmit the right-of-ways. The communication between vehicles and the intersection manager is mainly divided into three stages according to the movement of vehicle: the request of the right-of-way, the delivery of the right-of-way, and the removal of the right-of-way. The signaling technology combines with the technology of discrete point positioning to authenticate vehicles and to send the appropriate data according to their positions.

Based on this protocol, 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 se-

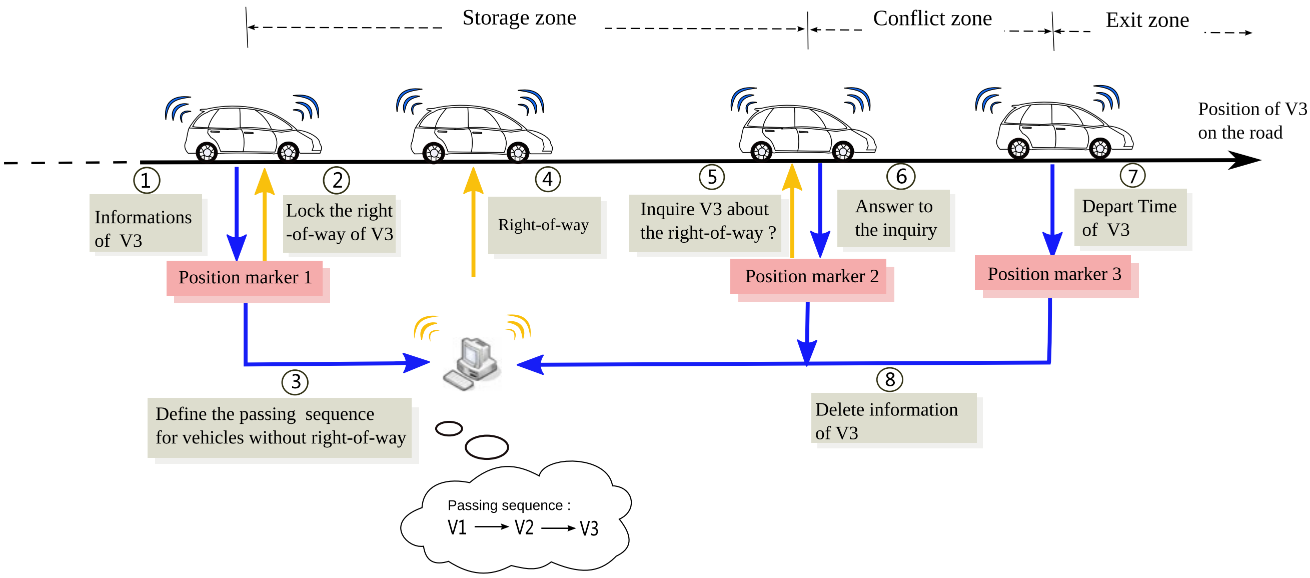


Fig. 2 Communication between a new coming vehicle V3 and the intersection manager

quence 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.

3 Problem formulation

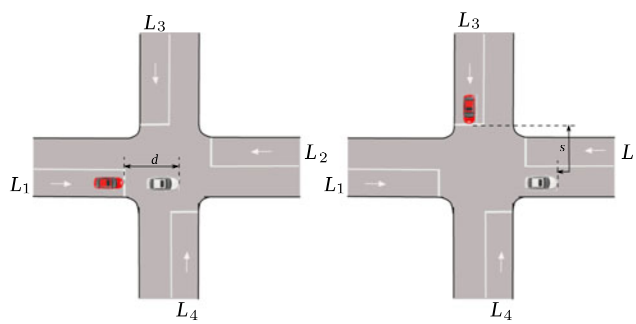
The main concern of AIM is to define the passing sequence based on vehicles' information. Before proposing a solution to this problem, traffic pattern of the proposed system is analyzed at first. Then, the formulation of the control problem is given based on the analysis.

3.1 Traffic pattern at an isolated intersection controlled by AIM

Since the concept of AIM is quite different from traditional traffic system which relies on the estimation of traffic flow rate, we need to focus on the discrete events observed by position markers:

1. Vehicle arrivals at the intersection;
2. Vehicle accesses to the conflict zone;
3. Vehicle departures.

Events (1) are the input data of the control problem and other events are used to ensure the safety. To decide the passing sequence, the system needs only an estimation of the instants of accesses to the conflict zone. Thus, we are interested in approximating the access times in consideration of the following three parameters:



(a) Safe headway between vehicles on the same lane

(b) Safe headway between vehicles of conflicting movements

Fig. 3 Safe headways

1. τ: the lower bound of the time taken to cover the distance between the position marker 1 and 2, which is used to predict the theoretical arrival time of vehicle;
2. d: the minimum safe headway time between two successive vehicles that move on the same lane (Fig. 3(a));
3. s: the minimum time between two successive vehicles that access to the conflict zone from conflicting movements (Fig. 3(b)).

To ensure the safety in the conflict zone, right-of-ways are assigned to vehicles of non-conflicting (compatible) movements. For example, a safe interval s is entailed between two vehicles from lanes L_1 and L_3 (Fig. 4). s takes into account the start-up lost time [27] and the time for clear-

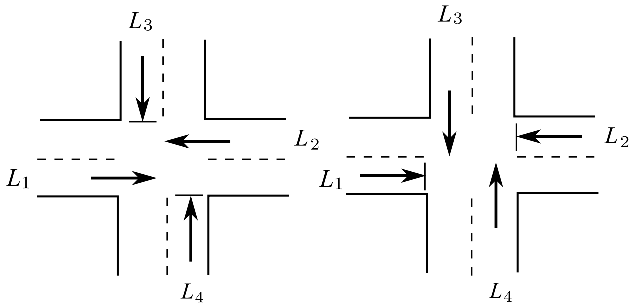


Fig. 4 Two conflicting movements of a 4-lane intersection shown in Fig. 1, following the National Electrical Manufacturers Association convention. For sake of readability, the turn-left and turn-right movements are not presented

ance of the conflict zone. In the reminder, we will call s and d safe headway.

τ, d and s depend on several parameters such as pavement surface conditions, profiles of the drivers, and weather conditions. One way to accurately estimate them is to refer to empirical observations. Literature is rich in terms of values and formulas with adjustment factors for approximating these parameters [28, 29]. Since the scope of this paper is not the estimation of these values, we will use the widely admitted standards [30].

3.2 Formulation of the control problem

Now, let us concentrate on the control of AIM. The main question is: how can the system ensure smoother traffic flow based on the observations of position markers? One answer to this question is to optimize the sequence of accesses to the conflict zone at each arrival of vehicles.

More precisely, given the presence list of vehicles by position markers 1 and theoretical arrival times at the conflict zone, the control of AIM involves a passing sequence of vehicles which minimizes the maximum exit time (the time when the last vehicle traverses the conflict zone), on respecting the safe headways.

The formulation of the control in AIM is presented in the following (Notations are defined in Table 1):

$$
\begin{aligned}
&\text{Given: } r_{(i,1)}, \ldots, r_{(i,n_i)}, i \in [1, L] \\
&\text{with } r_{(i,q_i)} < r_{(i,q_i+1)}, q_i \in [1, n_i - 1] \\
&\text{Find } \min\left\{\max_{i\in[1,L]} \{e_{(i,n_i)}\}\right\} \\
&\text{Subjects to the constraints:} \\
&|e_{(i,q_i)} - e_{(j,q_j)}| \geq \begin{cases} d, & i = j \wedge q_i = q_j + 1, \\ s, & i \hat{\neq} j, \end{cases} \\
&e_{(i,q_i)} \geq r_{(i,q_i)}
\end{aligned}
\tag{1}
$$

where $\hat{\neq}$ denotes two conflicting movements coming from lanes L_i and L_j.

Table 1 Definition of notation

Notation	Meaning
L	Total number of lanes
n_i	Number of vehicles present on the lane L_i
(i, q_i)	The q_ith arrival vehicle on the lane L_i
$r_{(i,q_i)}$	The theoretical arrival time of vehicle (i, q_i)
$e_{(i,q_i)}$	The exit time of vehicle (i, q_i). The exit time refers to the time when a vehicle accesses to the conflict zone
d	Safe headway respected by vehicles on the same lane
s	Safe headway respected by vehicles of conflicting movements

Table 2 Parameters of an example

	Theoretical arrival times			d	s
L_1	$r_{(1,1)} = 0$	$r_{(1,2)} = 3$	$r_{(1,3)} = 8$	2	6
L_2	$r_{(2,1)} = 1$	$r_{(2,2)} = 5$	$r_{(2,3)} = 10$		
L_3	$r_{(3,1)} = 4$	$r_{(3,2)} = 7$			
L_4	$r_{(4,1)} = 6$				

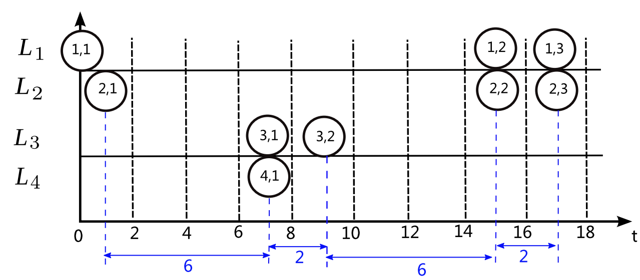


Fig. 5 An optimal vehicles passing sequence of the example given in Table 2

The minimization of the maximum exit time is an interesting approach to reduce the vehicle delay. Indeed, as the authors prove in [31], 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.

To better explain the control process, consider an example where nine vehicles are present at a 4-lane intersection in Fig. 1. There are three vehicles present on lanes L_1, L_2 respectively, two on L_3 and one on L_4. All the vehicles have not yet the right-of-way. The infrastructure provides the date at which these vehicles will be theoretically ready to cross the conflict zone, according to the observations of the position marker 1 and the parameters τ, d and s. Table 2 summarizes all the required data to compute the best passing sequence. Figure 5 presents an optimal sequence with the minimal exit time, i.e. 17 seconds.

3.3 Complexity

One of the major obstacles to achieve AIM is the problem complexity because the control considers each vehicle individually. The following proposition gives us the number of all feasible sequences.

Proposition 1 *An intersection of L-lane admits exactly*

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

ordered sequences of vehicles.

Proof There are

$$\left(\sum_{i=1}^{L} n_i\right)!$$

possible sequences. As on each lane, the n_i vehicles are stringently ordered (vehicle-overtake is forbidden after they have crossed the position marker 1), we conclude that the number of all feasible sequences equals to (2). □

We observe that the complexity of problem sharply increases with the number of vehicles and lanes. Hence, it is not possible to find the optimal solution by an exhaustive search. Hence, we could make use of some precious theories in combinational optimization. In [19], a dynamic programming algorithm is used to successfully solve the control problem in a simple intersection with two lanes. In [20], a branch and bound algorithm is proposed. Both algorithms can be generalized easily for the control of any layout of intersection [20, 32]. To the best of our knowledge, the time complexity is not lower than $\Theta(n^L)$, where n and L denote the total number of vehicles present at intersection and the number of lanes, respectively. However, it should be noted that even if the proposed control strategy has greatly reduced the time complexity, as the geometries of intersection become more complex, the computing time increases significantly. Hereby, the control strategy based on an exact algorithm cannot fulfill the real-time requirement of AIM. That argues for seeking help from meta-heuristics which have good reputations in solving complex problem in short time. In the following, we use the ant colony system to solve the problem (1).

4 Control strategy: ant colony system

Ant colony optimization (ACO) is one of the meta-heuristic optimization methods, which is inspired by the foraging behavior of ant colonies, and targets discrete optimization problems [33]. In the framework of ACO, a set of artificial ants cooperate to find the solution of a problem by exchanging information via pheromone deposited on graph edges [34]. ACO has been successfully applied to many combinatorial optimization problems, e.g., TSP [35, 36], assignment problems [37–39], scheduling problems [40–44], and vehicle routing problems [45–47].

TSP [48] is one of the most famous combinatorial optimization problems, where the goal is to find the shortest route that visits all the cities once and just once, given the finite number of cities and the distance between any two cities. In fact, there is an analogy between the problem (1) and the TSP. If we consider each vehicle as a city to be visited, then the required headway between each pair of vehicles determines the distance between two adjacent cities. Thus, the problem (1) can be converted to an analogous TSP, where the shortest path that visits each city once and just once corresponds to the sequence of vehicles with the minimal exit time. In this way, the ACS algorithm, one of the most successful ACO variants, is proper to solve (1). Before introducing the ACS algorithm, a graphical representation of the problem (1) is needed.

4.1 Graphical representation of the problem

In this subsection, we consider a 4-lane intersection (Fig. 1). However, we draw the reader's attention to the fact that the proposed control strategy can be applied to all intersection layouts. An acyclic and directed graph $G = (V, A, w)$ with $n + 2$ nodes is constructed as shown in Fig. 6. V (nodes) represents all the present vehicles and two dummy nodes, A (arcs) indicates the precedence relationship between two nodes (vehicles) and w (weights) denotes the weight of arc (safe headway time).

A node (i, q_i) represents the q_ith arrival vehicle of the lane L_i, where $1 \le q_i \le n_i$, $i \in [1, L]$. The two dummy

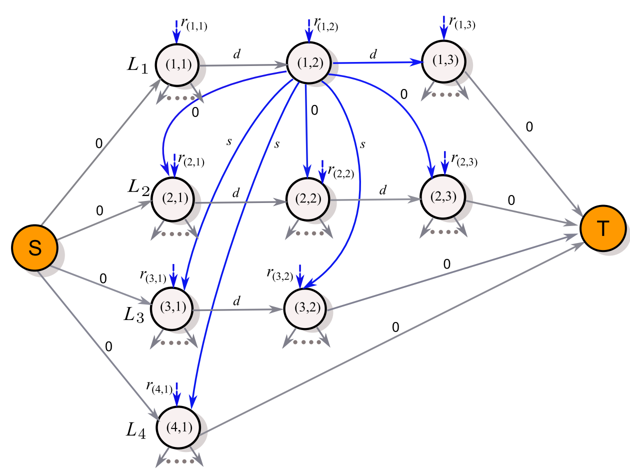


Fig. 6 Graphical representation of the example in Table 2 (arcs from the node $(1, 2)$ are illustrated)

nodes S (Source node) and T (Terminate node) indicate the beginning and the end of a sequence, respectively. The weight of arc (v_i, v_j), are defined as follows:

$$w(v_i, v_j) = \begin{cases} d, & \text{if } v_i = (i, q_i) \text{ and } v_j = (i, q_i + 1) \\ s, & \text{if } v_i = (i, q_i),\ v_j = (j, q_j) \text{ and } i \hat{\neq} j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{\neq}$ denotes two conflicting movements coming from lanes L_i and L_j.

Figure 6 illustrates all the successors of a node $(1,2)$ in the graph G. A weight d is assigned to the arc from $(1,2)$ to $(1,3)$ since the vehicles $(1,2)$ and $(1,3)$ on the same lane must respect a minimal safe headway d; while the weight of arcs from $(1,2)$ to all the nodes on the lane L_2 equals to 0 since movements of vehicles on the lanes L_1 and L_2 are non-conflicting. For the arcs from $(1,2)$ to $(3,1)$, $(3,2)$, and $(4,1)$, a weight s is assigned to each arc because the movement of the vehicle $(1,2)$ conflicts with any movement of vehicles coming from L_3 and L_4. It should be noted that for other nodes, the similar precedence relations (arcs) can be deduced as the node $(1,2)$. Hence, all arcs of the graph (Fig. 6) are not represented, for sake of readability.

A feasible solution of the problem (1) is a *path* $P(S,T)$ from the source node S to the terminate node T which visits all the nodes of the graph once and only once with respect to the precedence constraints defined by the arcs. In the reminder of this paper, a path refers to a feasible solution of the problem (1).

Based on the definition of the graph G, the length of a sub-path from S to a node (i, q_i) (denoted as $L_{(i,q_i)}$) indicates the access time of the corresponding vehicle (i, q_i) in a sequence. However, $L_{(i,q_i)} \geq r_{(i,q_i)}$, $r_{(i,q_i)}$ denotes the theoretical arrival time. Hence, $r_{(i,q_i)}$ is marked at the top of each node (i, q_i) in order to restrict the length of sub-path. With this in mind, the length of a path $P(S,T)$ denoted as L_T is calculated by the following recurrent formula that begins from the node S and terminates at the node T:

$$L_{v_j} = \max\{L_{v_i} + w(v_i, v_j), r_{v_j}\} \tag{4}$$

with v_i the direct predecessor of v_j and r_{v_j} the theoretical arrival time of the node v_j, $L_S = 0$. A path $P^*(S,T)$ is an optimal solution of the problem, iff for all paths $P(S,T)$, we have $L_T^* \leq L_T$.

4.2 Solution construction

The ACS algorithm (Algorithm 1) is applied, in order to find the shortest path $P^*(S,T)$. It works as follows: m ants are positioned on the node S at first. Each ant builds a path from S to T by the state transition rule. While constructing its path, an ant also modifies the amount of pheromone on the visited arcs by applying the local updating rule. Once all ants have arrived at the node T, the amount of pheromone on arcs is modified by a global updating rule.

From the description of the above algorithm, the initialization of pheromone, the state transition rule, and the pheromone updating rule are needed to be specified.

Algorithm 1 ACS ALGORITHM

```
Initialize pheromone values based on a heuristic;
Deposit m ants on the node S;
for i = 1, ..., C do (C: Number of iterations)
    for j = 1, ..., m do (m : Number of ants)
        Ant j uses a state transition rule to build a
        solution;
        A local pheromone updating rule is applied;
    end for
    A global pheromone updating rule is applied;
end for
```

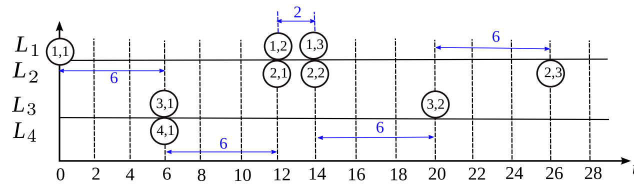


Fig. 7 Sequence of vehicles calculated by the proposed heuristic

4.2.1 *Initialize pheromone values by heuristic*

We use the method in [23] to set the initial pheromone $\tau_0 = (n \cdot L_0)^{-1}$, where n is the total number of nodes (i, q_i) and L_0 is the maximum exit time of the sequence found by a proposed heuristic.

This simple heuristic works as follows: right-of-ways are assigned to vehicles of non-conflicting movements whose headways are equal to the minimal value d. Figure 7 shows the sequence of the example introduced in Table 2 by applying this heuristic.

4.2.2 *State transition rule*

An ant a is in the position of the node j, then it chooses the node k to move to by applying the state transition rule:

$$k = \begin{cases} \arg \max\limits_{u \in V_a(j)} \{\tau(j,u) \cdot \eta(j,u)^\beta\}, \\ \quad \text{if } q \leq q_0 \text{ } \textit{exploitation} \\ K, \quad \text{otherwise } (\textit{biased exploration}) \end{cases} \tag{5}$$

where $\tau(j,u)$ is the pheromone on the arc (j,u) and $\eta(j,u)$ is the corresponding heuristic information if ant a moves from j to u, $V_a(j)$ is the set of nodes that remain to be visited by the ant a positioned on the node j. The value of parameter β ($\beta > 0$) denotes the importance of heuristic information. The parameter q_0 ($0 \le q_0 \le 1$) determines the relative importance of exploitation versus exploration: if q, a random number uniformly distributed in [0, 1], is smaller than q_0, then the ant a chooses the next vehicle k with the maximal value $\tau(j,u) \cdot \eta(j,u)^\beta$, that is called exploitation; otherwise, the next node is expressed as a random variable K, which is chosen according to the probability distribution (6):

$$p_a(j,k) = \begin{cases} \frac{\tau(j,k)\cdot\eta(j,k)^\beta}{\sum_{u\in V_a(j)} \tau(j,u)\cdot\eta(j,u)^\beta}, & \text{if } k \in V_a(j) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Equation (6) gives the probability with which the ant a chooses to move to the node k.

The heuristic information $\eta(j,k)$ represents the impact on the length of sub-path L_k after visiting the node k, which is defined as the reciprocal of the increase of length plus 1, as shown in (7).

$$\eta(j,k) = (L_k - L_j + 1)^{-1} \tag{7}$$

where L_j and L_k are given in (4). The reason of adding 1 is to prevent the denominator from being 0.

The state transition rule favors transitions toward nodes connected by short arcs and with large amount of pheromone.

4.2.3 Pheromone updating rule

During the process of building a solution, ants change the pheromones of the visited arcs by the local updating rule:

$$\tau(j,k) = (1-\rho)\cdot\tau(j,k) + \rho\cdot\tau_0 \tag{8}$$

where τ_0 is the initial pheromone level and ρ is a pheromone decay parameter ($0 < \rho < 1$). The smaller the value of ρ, the faster the solution converges; whereas the larger it is, the diversity of the solution increases.

By local updating rule, each time an ant passes through an arc, this arc becomes slightly less desirable for other ants. In this way, ants would not search in a narrow neighborhood of the best previous path.

After one iteration, all ants have finished a path from the node S to T and a global pheromone updating rule (9) is applied. However, only the globally best ant is allowed to deposit pheromone. This intends to lead ants to the neighborhood of the best path and results in a higher convergence speed.

$$\tau(j,k) = (1-\alpha)\cdot\tau(j,k) + \alpha\cdot\Delta\tau(j,k) \tag{9}$$

where

$$\Delta\tau(j,k) = \begin{cases} (L_{gb})^{-1}, & \text{if } (j,k) \in \text{global-best-tour} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

α is the pheromone decay parameter ($0 < \alpha < 1$) and L_{gb} is the length of the globally best path until this iteration.

Table 3 Experimental design for generating test scenarios

Experimental factors	Values	Levels
Sum of arrival rates λ	U[a][0, 0.15]	L
	U[0.15, 0.3]	M
	U[0.3, 0.5]	H
Simulation time T	50	L
	100	M
	150	H
Number of conflicting traffic movements M	2	L
	4	M
	6	H
Safe headway d	2	
Safe headway s	U[3, 10]	

[a]Uniform distribution

4.3 Analysis of ACS parameters

From the description of the ACS algorithm, we can see that the parameters α, β, ρ, q_0, m, C decide the quality of the solution. Next, we will conduct a series of experiments to further investigate these parameters. For each parameter, its influence on the relative percentage deviation (RPD) of the optimal solution is studied. RPD is defined as follows:

$$RPD = \frac{s - s^*}{s^*} \times 100\% \tag{11}$$

where s is the solution obtained by ACS and s^* is an optimal solution calculated by the dynamic programming algorithm [19]. The following experiments are implemented in Matlab on a 2.40 Ghz Pentium IV processor based PC.

Table 3 summarizes the experimental factors used to define the test scenarios: sum of flow rates of each compatible movement λ, simulation time T, number of conflicting movements M, safe headways d and s. The total number of vehicles is up to: $n = \lambda \times T \times M$. We assume that the arrivals of vehicles obey the Poisson distribution which represents a real-life traffic system with sufficient accuracy [49–51]. The first three factors are divided into three levels: L (Low), M (Middle) and H (High) according to the values. The safe headway d is set to 2 seconds since it is widely recognized that the maximum traffic flow rate is lesser than 0.5 vehicles per second [52]. Another safe headway s is set in the range of [3, 10] seconds.

Three test cases: case L, case M, and case H, are considered in the experiments. Ten independent runs are carried out and are averaged for each case. The values of parameters are set initially to the following values according to the report of Dorigo [23]:

$$\alpha = 0.1, \quad \beta = 2, \quad \rho = 0.1, \quad q_0 = 0.9, \quad m = 10$$

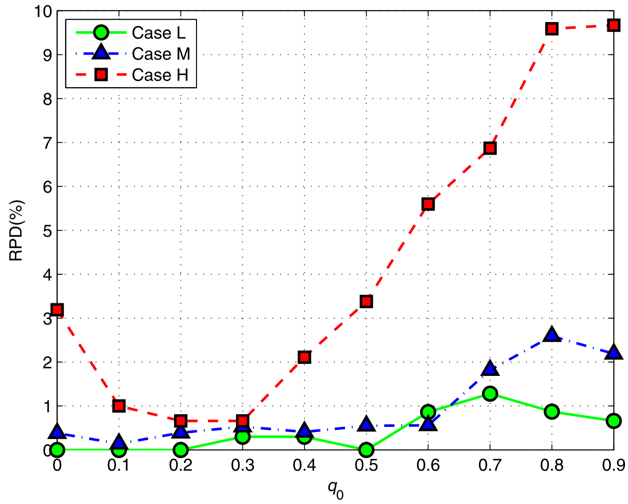


Fig. 8 Analysis of the parameter q_0

During the investigations, one or two parameters are changed each time to study its influence or their influences on the solution.

The investigation begins with the parameter q_0, which decides the degree of exploration versus exploitation. The value is set from 0 to 0.9 with a step length of 0.1. The result is shown in Fig. 8. We can see that the small value of q_0 (between 0.1 and 0.3) gives better results. Therefore, q_0 is set to 0.1. This configuration favors the biased exploration, which encourages exploring the arcs with less pheromone.

Next, the parameters α and ρ are tested together. They decide the speed of pheromone decay during the local updating and the global updating. The two parameters are set from 0 to 0.9 with a step length of 0.1. The values of the rest parameters: $\beta = 2$, $q_0 = 0.1$, $m = 10$. The results are reported in Table 4.

We can see that in each case, the combination of $\alpha = 0.3$, $\rho = 0.1$ give the minimum RPD. This configuration will increase the speed of the pheromone evaporation on the arcs out of the best path. As a consequence, the optimal solution obtained so far will be reinforced.

Now let us consider the parameter β. The value of β determines the importance of heuristic information. It is set from 0 to 5 with a step length of 1. The values of tested parameters are set to the following values: $\alpha = 0.3$, $\rho = 0.1$, $q_0 = 0.1$, $m = 10$. Figure 9 presents the experiment results.

As shown in Fig. 9, it is better to set β as 3. Besides, for each case, in particular case H, when β is 0, the performance of ACS become extremely poor since the information of heuristic plays an important role in the algorithm.

The last two important parameters are m and C. These parameters are related to not only the quality of solution but also to the computation cost. We begin the investigation by the parameter m, which denotes the number of ants. In the experiments, m varies from 5 to 25 with a step length of 5. Other parameters are set to $\alpha = 0.3$, $\rho = 0.1$, $\beta = 3$ $q_0 = 0.1$, $C = 5$. The results about the RPD and the computation cost (CPU time) are presented in Fig. 10. Obviously, the quality of the solution enhances as the number of m increases. However, considering the computation cost, a compromise must be made. Since $m = 5$ gives a relative good solution (*RPD* < 2.5%) in short calculation time (<1 s), the value of 5 is suggested.

Table 4 Analysis of parameters α and ρ

(a) Case L

$\rho\backslash\alpha$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0	0	0	0	0	0	0	0	0
0.1	0	0	0	**0**	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0
0.4	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0.31	0	0	0	0
0.6	0	0	0	0	0	0	0	0	0	0
0.7	0	0	0	0	0	0	0	0	0	0
0.8	0	0	0	0	0	0	0	0	0	0
0.9	0	0	0	0	0	0	0	0	0	0

(b) Case M

$\rho\backslash\alpha$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	0	0.38	0.13	0.63	0.65	0.37	0	0.12	0.12	0.75
0.1	0	0.13	0	**0**	0	0	0.13	0.14	0.13	0
0.2	0.25	0	0	0	0	0.14	0	0.13	0.13	0.13
0.3	0	0	0.13	0	0.26	0	0	0	0.13	0.14
0.4	0.12	0	0.14	0.14	0	0.13	0.13	0	0.27	0.27
0.5	0	0	0.26	0	0.12	0.39	0	0.13	0.51	0.12
0.6	0.13	0.27	0	0.13	0	0	0.13	0.13	0	0.13
0.7	0.13	0.13	0.12	0	0	0	0.26	0.13	0	0.13
0.8	0	0	0.26	0	0.13	0.25	0	0	0.13	0.12
0.9	0.26	0	0.13	0.25	0	0.14	0	0.13	0	0.13

(c) Case H

$\rho\backslash\alpha$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0	1.49	1.57	2.21	2.27	2.53	2.09	2.33	2.00	2.51	3.13
0.1	0.92	1.18	0.72	**0.58**	1.09	0.72	1.16	1.37	1.28	1.22
0.2	0.73	1.37	0.86	0.80	1.29	1.15	1.20	1.00	1.29	0.88
0.3	0.95	1.01	1.01	1.02	1.10	0.93	1.29	1.50	0.70	1.10
0.4	0.72	0.79	1.10	1.16	1.30	1.54	0.93	1.58	1.29	1.21
0.5	1.15	1.45	1.16	0.88	0.92	1.30	1.01	1.43	0.87	1.00
0.6	1.31	1.22	0.74	1.20	1.38	1.02	1.08	1.17	0.80	1.29
0.7	1.53	0.94	0.86	1.08	1.38	1.01	1.07	1.31	1.01	0.84
0.8	1.08	1.38	1.58	1.31	1.10	1.00	1.39	1.08	1.23	1.36
0.9	0.8	1.08	1.22	1.00	1.23	1.08	0.72	1.39	1.16	0.86

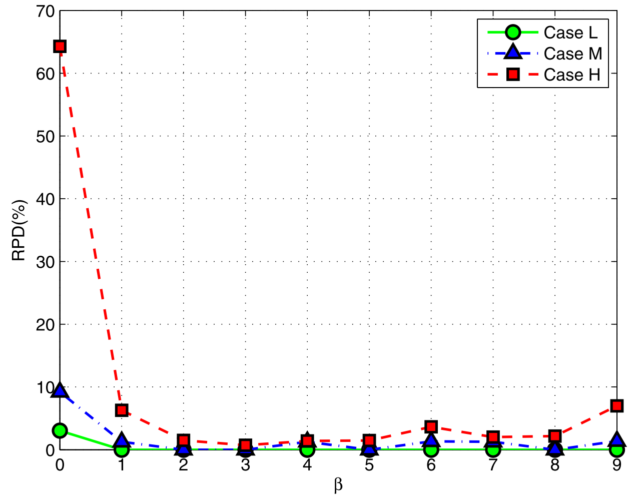
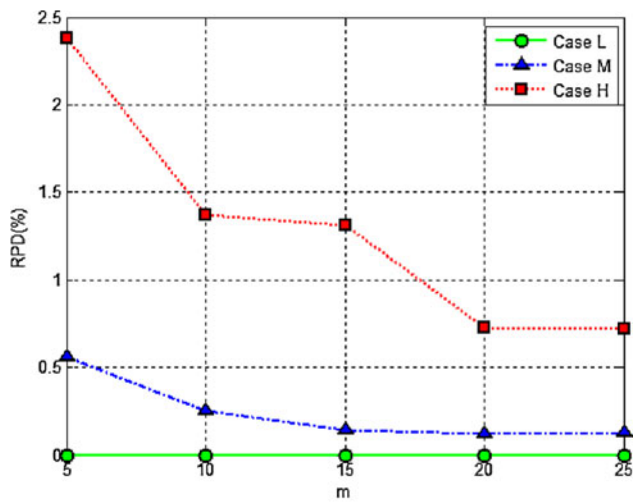
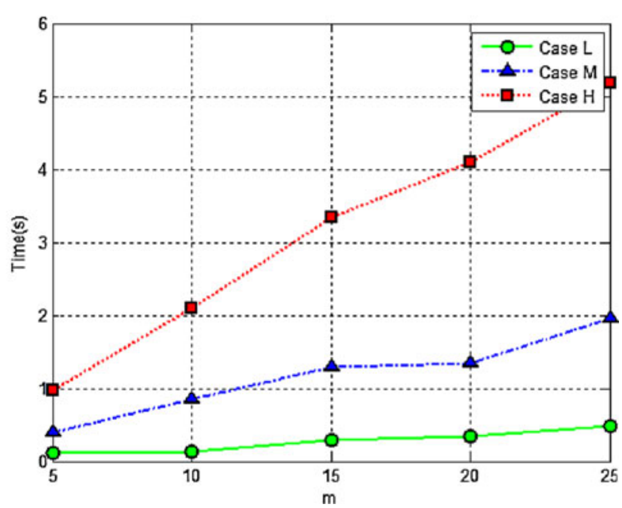


Fig. 9 Analysis of the parameter β

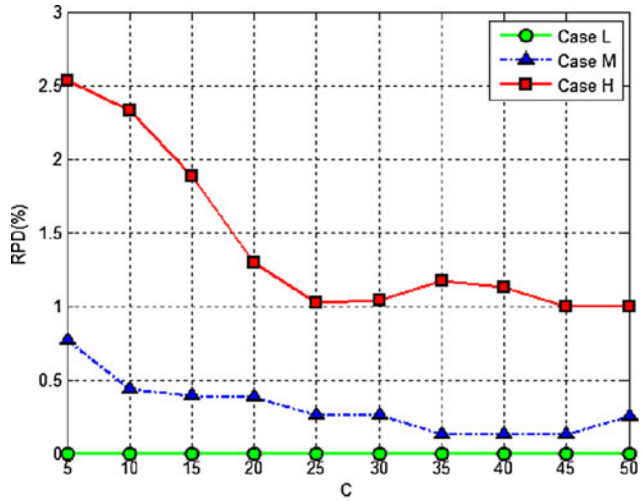


(a) RPD

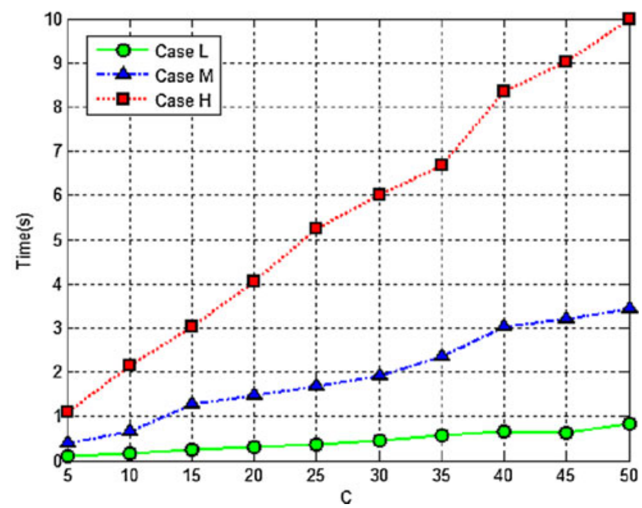


(b) Time of calculation (s: second)

Fig. 10 Analysis of the parameter m



(a) *RPD*



(b) Time of calculation (s: second)

Fig. 11 Analysis of the parameter C

As the parameter m, C determines the precision of solution and the computation cost. The value of C is taken from 5 to 50, while other parameters α, ρ, q_0, β, m take values 0.3, 0.1, 0.1, 3, 5, respectively. Figure 11 shows the influence of the parameter C on the quality of the solution and the computation cost. Based on these results, we can see that a value of 5 will achieve the best possible compromise.

From the above experiments, the parameters of ACS are: $\alpha = 0.3$, $\rho = 0.1$, $\beta = 3$, $q_0 = 0.1$, $m = 5$, $C = 5$. The configuration of the parameters is adapted to any layout of intersection, since several intersections are considered to obtain these values.

4.4 Example

Now we apply ACS algorithm to the example introduced at the end of Sect. 3.2.

At first, the pheromone is initialized:

$$\tau_0 = (n \cdot L_0)^{-1} = \frac{1}{9 \times 26} = 0.0043$$

where L_0 is the length of a feasible path found by the proposed heuristic shown in Fig. 7.

Then Algorithm 1 is implemented and yields a solution in 0.17 seconds, as shown in Fig. 12. The number at the top of each node (i, q_i) denotes the access time of the corresponding vehicle. The sequence of nodes on the shortest path corresponds to the optimal passing sequence of vehicles (Fig. 5).

Figure 13 presents the simulation of microscopic car-following behavior based on the optimal passing sequence. Gipps model [53, 54] is used to simulate the vehicle behavior since it shows lower error values on the accuracy of a traffic-simulation system [55]. Each curve represents vehicle's relative distance to the position marker 1 over time.

At the beginning, vehicles (1, 1) and (2, 1) receive the right-of-ways distributed by the intersection manager. Once they have passed over the position marker 3, the intersection manager sends the right-of-ways to vehicles (3, 1), (3, 2), and (4, 1) of conflicting movements. As the last vehicle (3, 2) has passed over the position marker 3, the left vehicles (1, 2), (1, 3), (2, 2), and (2, 3) receive the right-of-ways.

We can note that the last four vehicles receive the right-of-way after 12 s rather than 15 s (Fig. 5). Since the exit time can be reduced in practical if the vehicles exit the intersection more quickly than expected. The position marker 3 will be helpful to save the time.

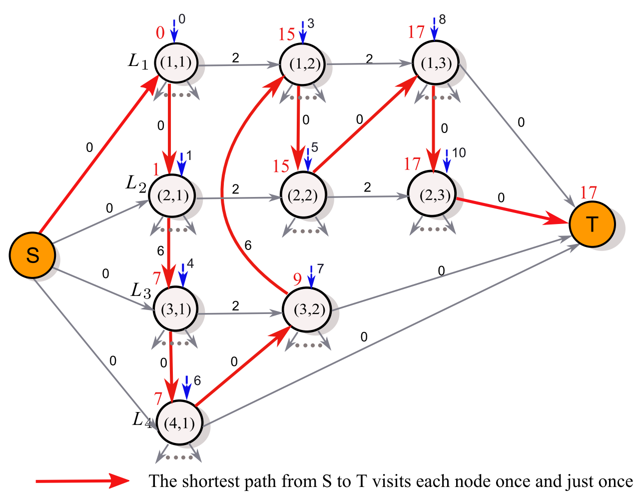


Fig. 12 An optimal solution found by ACS

5 Experiments

In this section, we will conduct a series of experiments to evaluate the robustness of the proposed algorithm. In addition, the traffic control effectiveness of AIM based on ACS will be studied.

5.1 Algorithm performance comparisons

The performance of Dynamic programming and ACS under different scenarios are investigated, i.e. level L, M and H defined in Table 3. Fifty test problems are generated for each level.

First, the accuracy of the algorithm ACS is measured by two criteria: the Relative Percentage Deviation (RPD) given in (11) and the Percentage of the Optimal Solution produced (POS). The results (Table 5) show that as the scale of problem increases, although the average POS decreases, the average RPD keeps in a low value, i.e. the maximal average $RPD = 2.45\%$.

Next, the computation times of ACS and the dynamic programming algorithm are compared (Table 6). We note that ACS finds a high quality solution in a very short period of time (less than one second), even in the level H.

From above comparisons, we see that ACS consistently performs well in various scenarios on the solution quality while reducing the computational cost.

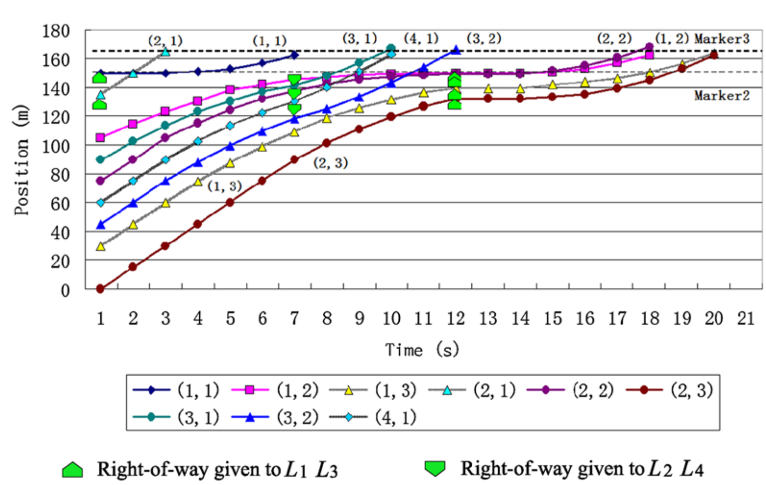


Fig. 13 Simulation of Microscopic car-following behavior

Table 5 Accuracy of ACS

Levels	RPD			Average POS
	Minimum	Maximum	Average	
L	0%	3.13%	0.23%	93%
M	0%	3.90%	0.56%	66%
H	0%	5.19%	2.45%	8%

Table 6 Average computation time (seconds)

Levels	Dynamic programming	ACS
L	0.1439	0.0832
M	10.1722	0.3642
H	350.2716	0.9939

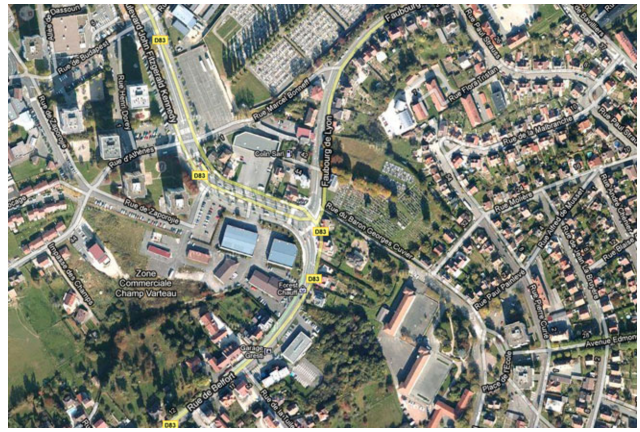


Fig. 14 T-intersection between the roads Boulevard John Fitzgerald Kennedy, Faubourg de Lyon, and Rue de Belfort at Belfort, France (map from http://maps.google.com)

5.2 Traffic control effectiveness

In this part, the traffic control effectiveness (improvement in efficiency of traffic flow) of AIM based on ACS is evaluated. The simulation is implemented in a traffic simulation environment where the car-following model (Gipps model) [53, 54] is used to simulate the behavior of vehicles. The inputs of the simulation (d, s and τ) are provided by the regional project (Belfort, France) "Edude de Définition pour la Desserte en Transport Collectif et l'Intermodalité autour de la Future TGV de l'Aire Urbaine".

The effectiveness of AIM based on ACS 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 in [8] is used for comparison;
2. New traffic control system based on wireless communication proposed by Gradinescu [9]: the controller keeps tracking the vehicles by the wireless communication, thus the estimation of the traffic volume is more precise. The timing plan is updated in time (during each cycle). The famous Webster's formula [50] is applied to calculate the cycle length and the green time according to the estimated demand;
3. New traffic control system proposed in [10]: the control strategy is based on the Little's formula [56]. The average delay experienced by the vehicles in the network is directly proportional to the average queue size. Hence, the controller tries to minimize the queue size by giving a green light 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 in [11, 21]: 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 FCFS "First Come, First Served" control policy.

It is 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. "AIM (ACS)" represents the AIM system based on the proposed ACS algorithm.

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 intersection under consideration is a T-intersection of six lanes presented in Fig. 14. The traffic is able to execute the left turn and right turn at this intersection. The set of conflicting movements are illustrated in Fig. 15. The distance between the position marker 1 and 2 is 150 m and the lane width is 3.5 m. The simulation step is 1 s. The simulation runs sixty minutes and each point in the following

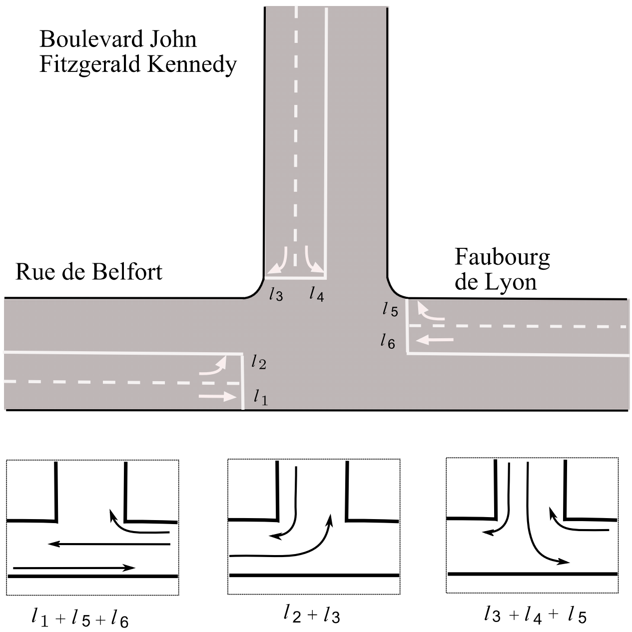


Fig. 15 Layout and set of conflicting movements of T-intersection shown in Fig. 14

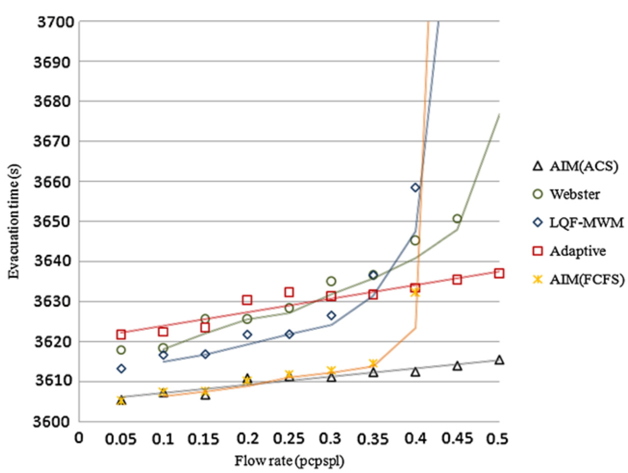


Fig. 16 Evacuation time (s: second)

figures (Figs. 16–19) represents an average of ten runs. The unit of flow rate "pcpspl" denotes passenger cars per second per lane.

First, let us compare AIM (ACS) with the traditional traffic control system. The results show that AIM (ACS) outperforms the system "Adaptive". Besides, its performance is as stable as the system "Adaptive". Turning to the comparison with other new traffic control systems, AIM (ACS) 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 low traffic load (≤0.3 pcpspl), however as the traffic load in-

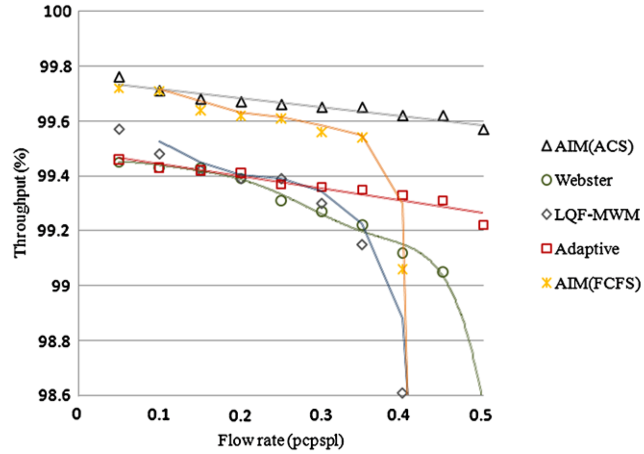


Fig. 17 Throughput

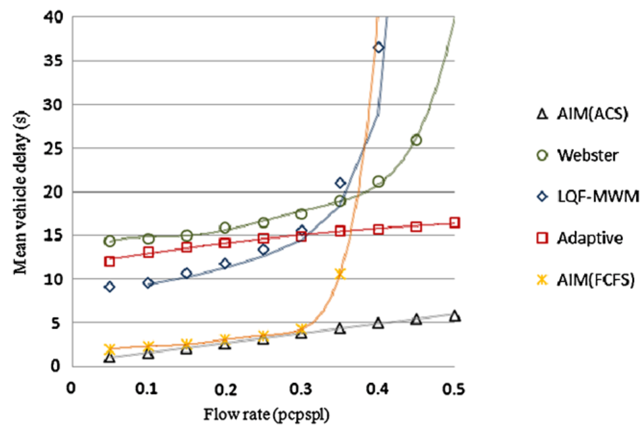


Fig. 18 Mean vehicle delay (s: second)

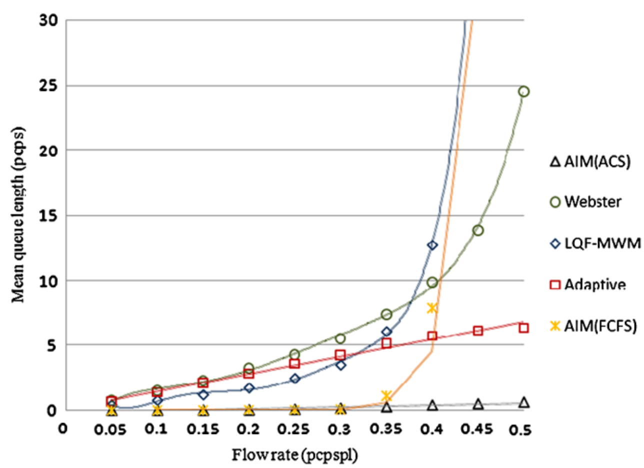


Fig. 19 Mean queue length (pcps: passenger cars per second)

creases, its performance grows worse. 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)".

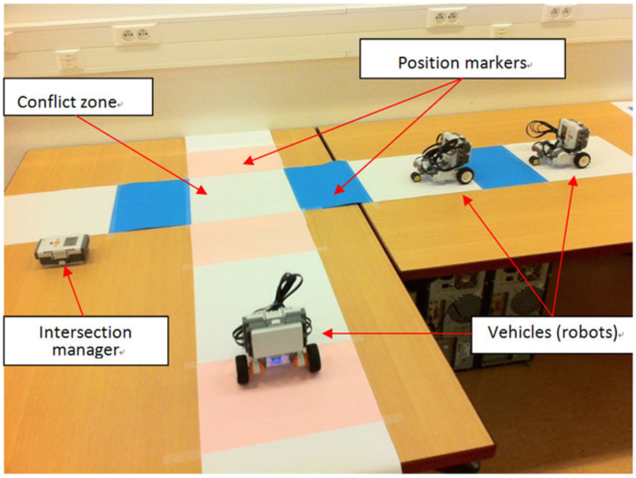


Fig. 20 Prototype of an elementary AIM

When the traffic load is low, i.e. $\leq$0.35 pcpspl, it has the compatible performance with "AIM (ACS)"; as the traffic load increases, it shows a sharp decline in its performance. The comparison with "AIM (FCFS)" proves that reasonable arrangement of vehicle passing sequence will improve the traffic efficiency.

6 Discussion

The simulations show that the proposed system has capacity to improve the traffic efficiency. Whereas, there are several issues that deserve further investigations:

6.1 Feasibility

A prototype based on NXT robots (Fig. 20) has been built to study the feasibility of AIM. During the tests, we have not observed any collision. (As shown in the animation: Online Resource 1 and 2.) It is noted that the safety issue depends on the precision and the robustness of the positioning system. The prototype proves the feasibility of the proposed protocol of AIM, for at least industrial applications (e.g. robots). Moreover, the European Rail Traffic Management (ERTMS)/ETCS levels 2 and 3 control the train access to the block segments using electronic position markers (Eurobalise), as we have proposed in this paper.

We draw the reader's attention to the fact that AIM can be forthcoming because 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 in France, City-Car in Switzerland, IntelliShare in the USA) have planned to limit the access of certain zones of the city to green car sharing. In other words, some intersections of the urban area will be exclusively shared by fleets of vehicles that can be quickly equipped together for negotiating their access. Hence, AIM has a great potential to be applied in green car sharing zones.

6.2 Priority vehicles

Vehicles such as bus, vehicle with less pollution, can have the priority to cross intersection than common vehicles, as stated in [18]. In this case, the intersection manager could take into account giving priority to these vehicles by assigning a weight to each vehicle. Vehicles with high priority have greater weights. The object of control turns to minimize the weighted sum of exit times of all present vehicles. The proposed algorithm can easily adapt to this new problem except that the length of the path corresponds to the weighted sum of vehicle exit time.

6.3 Reaction to dangerous situations

Since generally the actual traffic is very complex, the intersection manager should be flexible to deal with emergencies. For example, how to deal with the situation where a vehicle suddenly stops in the conflict zone? The position marker 3 (Fig. 1) will help to reduce risks of collisions. If a vehicle stalls at the conflict zone, the position marker 3 will not inform the intersection manager of its departure. Thus, the intersection manager will not free the vehicles of the conflicting movements. Nevertheless, some supplemental surveillance equipments, such as traffic cameras, are needed to aid monitoring the traffic, as it is required for the current traffic control system (traffic lights). Hence, the proposed system just contributes to a better supervision and reduces the risk of collision.

7 Conclusion

This paper introduces a protocol of autonomous intersection management system to efficiently control the traffic. This new traffic control system is capable of managing the vehicles individually based on wireless communication and positioning technologies. The right-of-way is sent to the robot or displayed to the driver by means of the onboard signalization. Analysis of the new system renders the control as a combinational optimization problem. An algorithm of ant colony system is proposed to solve the control in real time. Simulations show that the proposed system performs better than the adaptive controller which is popularly used at present and even better than the innovative systems based on the new technologies. In addition, this system has the potential to be applied to any intersection with more complicated layout. In the future, we will provide more validation data not just limit on the simulation environment. Moreover, the proposed ant colony system is used to quickly improve a sequence that initially results from simple rules based on the headway time of each vehicle and on its temporal distance from the conflict zone. This encourages us to explore some interesting properties of the optimal solution for contributing to the decentralization of the sequence formation in the future.

References

1. Sakaguchi T, Uno A, Kato S, Tsugawa S (2000) Cooperative driving of automated vehicles with inter-vehicle communications. In: Proceedings of the IEEE intelligent vehicles symposium, pp 516–521
2. Uno A, Sakaguchi T, Tsugawa S (1999) A merging control algorithm based on inter-vehicle communication. In: Proceedings 1999 IEEE/IEEJ/JSAI international conference on intelligent transportation systems, pp 783–787
3. Tsugawa S (2002) Inter-vehicle communications and their applications to intelligent vehicles: an overview. In: IEEE intelligent vehicle symposium 2002, vol 2, pp 564–569
4. Gehring O, Fritz H (1997) Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication. In: IEEE conference on intelligent transportation system 1997, pp 117–122
5. Shladover SE, Desoer CA, Hedrick JK, Tomizuka M, Walrand J, Zhang WB, McMahon DH, Peng H, Sheikholeslam S, McKeown N (1991) Automated vehicle control developments in the PATH program. IEEE Trans Veh Technol 40(1):114–130
6. Arora S, Raina AK, Mittal AK (2000) Collision avoidance among AGVs at junctions. In: Proceedings of the IEEE intelligent vehicles symposium 2000, pp 585–589
7. Li L, Fei-Yue W (2006) Cooperative driving at blind crossings using intervehicle communication. IEEE Trans Veh Technol 55(6):1712–1724
8. Fang FC, Elefteriadou L (2006) Development of an optimization methodology for adaptive traffic signal control at diamond interchanges. J Transp Eng 132(8):629–637
9. Gradinescu V, Gorgorin C, Diaconescu R, Cristea V, Iftode L (2007) Adaptive traffic lights using car-to-car communication. In: IEEE 65th vehicular technology conference, 2007, Spring, pp 21–25
10. Wunderlich R, Cuibi L, Elhanany I, Urbanik T (2008) A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection. IEEE Trans Intell Transp Syst 9(3):536–547
11. Dresner K, Stone P (2006) Traffic intersections of the future. In: The twenty-first national conference on artificial intelligence. NECTARTrack (AAAI 06), pp 1593–1596
12. Neuendorf N, Bruns T (2004) The vehicle platoon controller in the decentralised, autonomous intersection management of vehicles. In: Proceedings of the IEEE international conference on mechatronics 2004, pp 375–380
13. Naumann R, Rasche R, Tacken J, Tahedi C (1997) Validation and simulation of a decentralized intersection collision avoidance algorithm. In: IEEE conference on intelligent transportation system 1997, pp 818–823
14. Lachner R (1997) Collision avoidance as a differential game: real-time approximation of optimal strategies using higher derivatives of the value function. In: IEEE international conference on systems, man and cybernetics 1997. Computational cybernetics and simulation, vol 3, pp 2308–2313
15. Hunt PB (1982) The SCOOT on-line traffic signal optimisatoin technique. Traffic Eng Control 23:190–192
16. Gartner N (1983) OPAC: a demand-responsive strategy for traffic signal control. Transp Res Rec 906:75–81
17. Farges JL, Henry JJ, Tufal J (1983) The PRODYN real-time traffic algorithm. In: The 4th IFAC symposium on transportation systems, pp 307–312
18. Dresner K, Stone P (2007) Learning policy selection for autonomous intersection management. In: The AAMAS 2007 workshop on adaptive and learning agents (ALAg 2007), pp 34–39
19. Wu J, Abbas-Turki A, Correia A, El Moudni A (2007) Discrete intersection signal control. In: IEEE international conference on service operations and logistics and informatics 2007, pp 1–6
20. Yan F, Dridi M, El Moudni A (2009) A branch and bound algorithm for new traffic signal control system of an isolated intersection. In: International conference on computers & industrial engineering 2009, pp 999–1004
21. Dresner K, Stone P (2004) Multiagent traffic management: a reservation-based intersection control mechanism. In: Proceedings of the third international joint conference on autonomous agents and multiagent systems 2004, pp 530–537
22. Quinlan M, Tsz-Chiu A, Zhu J, Stiurca N, Stone P (2010) Bringing simulation to life: a mixed reality autonomous intersection. In: IEEE/RSJ international conference on intelligent robots and systems (IROS) 2010, pp 6083–6088
23. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66
24. Grünewald M, Rust C, Witkowski U (2006) Using mini robots for prototyping intersection management of vehicles. In: The 3rd international symposium on autonomous minirobots for research and edutainment (AMIRE 2005), pp 287–292
25. FHWA (2005) Traffic control systems handbook. Dunn engineering asociates and siemens intelligent transportation systems
26. NHTSA (2005) Vehicle safety communications project task 3 final report—identify intelligent vehicle safety applications enabled by DSRC. US Department of Transportation: National Highway Traffic Safety Administration, Washington, DC
27. Kutz M (2004) Handbook of transportation engineering. McGraw-Hill, New York
28. Chen X, Li L, Zhang Y (2010) A Markov model for headway/spacing distribution of road traffic. IEEE Trans Intell Transp Syst 11(4):773–785
29. Bertolazzi E, Biral F, Da Lio M, Saroldi A, 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 Trans Intell Transp Syst 11(3):525–538
30. Cohen S (1993) Ingénierie du trafic routier. Eléments de théorie du trafic et applications. Seconde edn. Presses de l'Ecole National des Ponts et Chaussées
31. Diakaki P, Kotsialos D, Wang Y (2003) Review of road traffic control strategies. Proc IEEE 91(12):2041–2042
32. Wu J, Abbas-Turki A, El Moudni A (2009) Intersection traffic control by a novel scheduling model. In: IEEE/INFORMS international conference on service operations, logistics and informatics 2009, pp 329–334
33. Dorigo M, Stutzle T (2004) Ant colony optimization. MIT Press, Cambridge
34. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. Theor Comput Sci 344(2–3):243–278
35. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern, Part B, Cybern 26(1):29–41
36. Dorigo M, Maniezzo V, Colorni A (1991) Positive feedback as a search strategy. Tech Report 91-016. Laboratorio di Calcolatori, Dipartimento di Elettronica, Politecnico di Milano, Milano, Italy

37. Maniezzo V (1999) Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS J Comput 11(4):358–369
38. Maniezzo V, Colorni A (1999) The ant system applied to the quadratic assignment problem. IEEE Trans Knowl Data Eng 11(5):769–778
39. Costa D, Hertz A (1997) Ants can colour graphs. J Oper Res Soc 48(3):295–305
40. Blum C, Sampels M (2004) An ant colony optimization algorithm for shop scheduling problems. J Math Model Algorithms 3(3):285–308
41. Gagné C, Price WL, Gravel M (2002) Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times. J Oper Res Soc 53(8):895–906
42. Merkle D, Middendorf M, Schmeck H (2002) Ant colony optimization for resource-constrained project scheduling. IEEE Trans Evol Comput 6(4):333–346
43. Merkle D, Middendorf M (2003) Ant colony optimization with global pheromone evaluation for scheduling a single machine. Appl Intell 18(1):105–111
44. Zhang N, Feng Z-R, Ke L-j (2010) Guidance-solution based ant colony optimization for satellite control resource scheduling problem. Appl Intell 1–9. doi:10.1007/s10489-010-0234-3
45. Bullnheimer B, Hartl RF, Strauss C (1999) An improved ant system algorithm for the vehicle routing problem. Ann Oper Res 89(0):319–328
46. Lee C-Y, Lee Z-J, Lin S-W, Ying K-C (2010) An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. Appl Intell 32(1):88–95
47. Reimann M, Doerner K, Hartl RF (2004) D-Ants: savings based ants divide and conquer the vehicle routing problem. Comput Oper Res 31(4):563–591
48. Thomas HC, Charles EL, Ronald LR, Clifford S (2009) Introduction to algorithms, 3rd edn. MIT Press, Cambridge
49. May A (1990) Traffic flow fundamentals. Prentice Hall, Englewood Cliffs
50. Webster FV (1958) Road research technical paper. Road Research Laboratory, London, UK
51. Mannering FL, Kilareski WP, Washburn SS (2004) Principles of highway engineering and traffic analysis, 3rd edn. Wiley, Hoboken
52. Barney GC (2003) Elevator traffic handbook: theory and practice. Taylor & Francis, London
53. Gipps PG (1981) A behavioural car-following model for computer simulation. Transp Res, Part B, Methodol 15(2):105–111
54. Gipps PG (1986) Multsim: a model for simulating vehicular traffic on multi-lane arterial roads. Math Comput Simul 28(4):291–295
55. Panwai S, Dia H (2005) Comparative evaluation of microscopic car-following behavior. IEEE Trans Intell Transp Syst 6(3):314–325
56. Little JDC (1961) A proof for the queuing formula: $L = \lambda W$. Oper Res 9(3):383–387