

Research Article

A Genetic Algorithm Approach for Expedited Crossing of Emergency Vehicles in Connected and Autonomous Intersection Traffic

Qiang Lu¹ and Kyoung-Dae Kim²

¹Department of Electrical and Computer Engineering, University of Denver, Denver, CO 80210, USA

²Department of Information and Communication Engineering, DGIST, Daegu 42988, Republic of Korea

Correspondence should be addressed to Kyoung-Dae Kim; kkim@dgist.ac.kr

Received 12 May 2017; Revised 27 July 2017; Accepted 29 August 2017; Published 11 October 2017

Academic Editor: Meng Wang

Copyright © 2017 Qiang Lu and Kyoung-Dae Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an intersection control algorithm which aims to determine an efficient vehicle-passing sequence that allows the emergency vehicle to cross an intersection as soon as possible while the travel times of other vehicles are minimally affected. When there are no emergency vehicles within the intersection area, the vehicles are controlled by the DICA that we proposed in our earlier work. When there are emergency vehicles entering the communication range, we prioritize emergency vehicles through optimal ordering of vehicles. Since the number of possible vehicle-passing sequences increases rapidly with the number of vehicles, finding an efficient sequence of vehicles in a short time is the main challenge of the study. A genetic algorithm is proposed to solve the optimization problem which finds the optimal vehicle sequence that gives the emergency vehicles the highest priority. The efficiency of the proposed approach for expedited crossing of emergency vehicles is validated through comparisons with DICA and a reactive traffic light algorithm through extensive simulations. The results show that the proposed genetic algorithm is able to decrease the travel times of emergency vehicles significantly in light and medium traffic volumes without causing any noticeable performance degradation of normal vehicles.

1. Introduction

Roads have become more and more congested because of the rapidly increasing demand for transportation due to the larger and growing population in the world in recent years. Usually, such transportation problems could be alleviated by careful city planning, but planning does not usually work well with the unexpected growth in the population and vehicle usage. As efficient ways to address congestion problems, self-driving vehicles and autonomous transportation systems have attracted a lot of research and development efforts from academia, industry, and governments. For example, during the mid-1990s, the California PATH (Partners for Advanced Transportation Technology) launched the Automated Highway System program [1] and the US DARPA (Defense Advanced Research Projects Agency) held a series of autonomous vehicle challenges during the 2000s [2]. Also,

many companies have already made decisions to hugely invest in developing their own self-driving vehicles or vehicles with advanced driving assistance systems [3]. Furthermore, it is also shown in [4] that the overall capacity of the traffic can be improved substantially when all vehicles on roads are autonomous and connected for collision avoidance. In recent years, many researches have been done to improve intersection control performance by using optimization approaches. Reference [5] developed a novel linear programming formulation for autonomous intersection control in which the nonlinear constraints were relaxed by a set of linear inequalities. While the objective function of the optimization problem in [5] involves the travel time, other studies [6, 7] are trying to solve similar control problems using an objective function with multiple criteria like safe speeds and acceleration while avoiding collisions. The authors of [8] proposed an algorithm with provable guarantees on

safety and performance. A rigorous upper bound is provided for the expected wait time in their approach. Compared with centralized control approaches, infrastructure support is not needed in decentralized control. In the approach [9] proposed by Wu et al., the estimated arrival time is shared wirelessly among vehicles to obtain the best passing sequence. The problem of coordinating online a continuous flow of Connected and Autonomous Vehicles crossing two adjacent intersections was formulated as a decentralized optimal control problem in [10]. The solution gives the optimal acceleration/deceleration for each vehicle at any time to minimize fuel consumption. Some researchers also studied the control mechanism when only a part of the traffic is autonomous vehicles [11]. While there are already numerous research efforts in the autonomous traffic area, the development of safe and efficient autonomous transportation systems is still at its early stage. We have proposed the DTOT- (Discrete-Time Occupancies Trajectory-) based Intersection Control Algorithm toward this objective in previous work [12, 13], and we want to augment the algorithm to give emergency vehicles (EVs) preferences to cross intersections quickly.

Human lives and the amount of financial loss highly depend on the response time (from the time the emergency service is called to the time help is offered) of emergency vehicles. The travel time of emergency vehicles to the accident scene is critical to the response time. So, it is very useful and helpful to reduce the travel time of emergency vehicles on roads, especially on intersections where congestions are more likely to happen. The survival chance of injured people in an accident falls sharply if they reach the operating table later than 60 minutes after the accident [14]. Hence, shortening the travel times of crossing intersections for emergency vehicles will help save lives. In reality, the current way to handle emergency vehicles is similar to using Vehicle-to-Vehicle (V2V) communication (siren and lights) to warn nonemergency vehicles on roads to yield to the emergency vehicle. Some drivers cannot respond quickly to the warnings, which may result in additional time delay for emergency vehicles and even serious accidents. Many studies have been carried out to allow emergency vehicles to have a faster travel across intersections. Based on MAS (Multiagent System), [15] introduced a state machine for the intersection controller to change traffic signal status according to lane occupation when an emergency vehicle is approaching. Some researchers have explored the priority evacuation of emergency vehicles under an autonomous and connected traffic environment. Viriyasitavat and Tonguz proposed an intersection control system that only uses Vehicle-to-Vehicle (V2V) communication to give emergency vehicles priority of crossing [16]. The paper proposed that, at an intersection, a leader should be elected from all approaching vehicles to serve as the temporary traffic light infrastructure and stop at the intersection to coordinate the traffic. The green signal is always given to the lane of detected emergency vehicles and through coordination “green-wave” signals are displayed for the emergency vehicles to let them move at a faster speed. Dresner and Stone proposed a simple way to deal with emergency vehicles under their intersection control framework AIM (Autonomous Intersection Management) [17]. Their algorithm only grants

reservations to vehicles in the lanes that have approaching emergency vehicles, which allows the emergency vehicle to continue on its way relatively unhindered. However, the travel times of nonemergency vehicles will be affected significantly and no global coordination is made for optimal traffic flow.

This paper extends our previously proposed DTOT-based Intersection Control Algorithm (DICA) approach to include emergency vehicles in the traffic to be controlled. Our goal of intersection control is to let emergency vehicles cross intersections as fast as possible while maintaining adequate traffic performance. In this paper, we assume that emergency vehicles are taking normal routes, which means that they will not travel in a wrong lane. A genetic algorithm is proposed to find the optimal passing sequence of vehicles whose trajectories can be rearranged. This optimal sequence aims to make the emergency vehicles cross the intersection in the fastest way. Among many sequence forming approaches [18–21] in the literature, the authors of [21] proposed the most similar approach to ours, which also proposed a genetic algorithm to form vehicle sequences. However, unlike the approach proposed in this paper, they are essentially not allowing vehicles with conflicting routes to be inside the intersection at the same time.

The rest of the paper is arranged as follows. Section 2 recalls the previously proposed intersection control scheme, DICA, which is the foundation of this paper’s approach for emergency vehicles. A new architecture and algorithm, called *Reactive DICA*, for emergency vehicle handling is described in Section 3. In Section 4, we propose a genetic algorithm that efficiently solves the optimal vehicle sequencing problem formulated in Section 3. The efficiency and accuracy of the approach are evaluated through simulations in Section 5. Finally, we draw the conclusion in Section 6.

2. DICA

In our previous work, we proposed a novel algorithm named *DICA* [12] to coordinate autonomous intersection traffic and later improved the computational complexity of the algorithm significantly through several computational approaches [13]. In this section, we review the basic idea of the algorithm *DICA* which is the base algorithm to be extended for emergency vehicles in this paper.

At the intersection considered in our work, a vehicle communicates with the Intersection Control Agent (ICA) to get a permission to cross. As shown in Figure 1, we define two regions in an intersection. The *communication region* is determined by the wireless vehicular communication range, and the *intersection region* is the common area within an intersection that is shared by all roads connected to the intersection. All vehicles are Connected and Autonomous Vehicles (CAVs) and are capable of wireless communications with the ICA. It is also assumed that each vehicle is equipped with a RFID (Radio Frequency IDentification) chip and detectors are installed on every incoming lane at the entrance of the communication region. Thus, ICA can detect the information of each vehicle such as Vehicle Identification Number (VIN), the lane on which a vehicle is approaching an intersection, and the time when a vehicle enters the

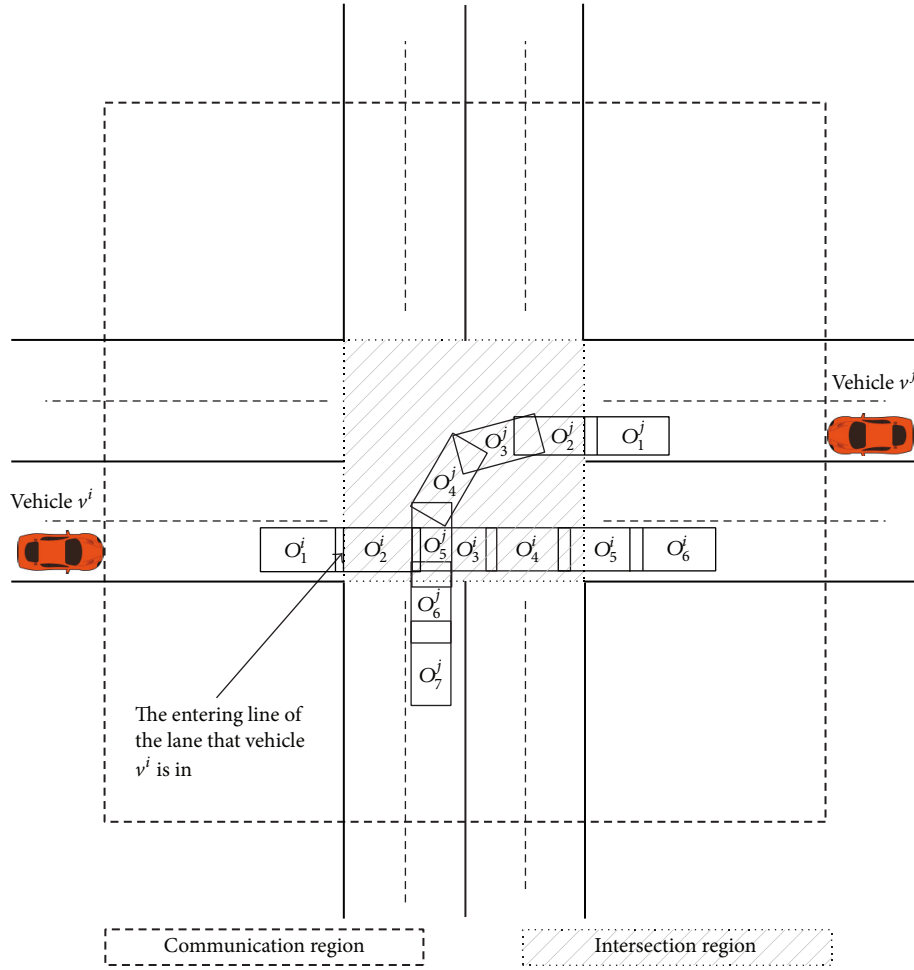


FIGURE 1: DTOTs of two conflicting vehicles. O_q^p represents the q th occupancy in a vehicle v^p 's DTOT. Note that occupancies in this figure are intentionally made very sparse for clear illustration purposes. DTOT starts with the occupancy in which the vehicle's front bumper first comes into contact with the entering line of its lane of an intersection, and ends with the occupancy that the vehicle is completely out of the intersection region.

communication region. It is assumed that a CAV can obtain its current position, speed, and the relative distance to an intersection precisely and also can avoid collisions with other vehicles autonomously. We also assume an ideal wireless vehicular communication performance such that all data packages are exchanged correctly and timely since this paper focuses on developing an algorithm for ICA to evacuate emergency vehicles as quickly as possible. It is important to note that, however, despite this ideal communication assumption, the DICA algorithm could still be applicable in practice with minor modifications to deal with imperfect communication thanks to the above-mentioned information collection mechanism through detectors. For example, if the communication between CAVs and ICA is not perfect, we may have two possible problems to solve: package delay and loss. For delayed data packages, we can obtain the upper bound for communication delay and integrate this delay with the proposed DTOT by extending every occupancy to take into account the delay upper bound. This approach will make the algorithm a little bit more conservative but safe for

vehicles. For lost data packages, we can add an ACK message in the algorithm. A head vehicle (or ICA) will send REQUEST (or RESPONSE) again if it does not receive an ACK message from ICA (or the head vehicle). These simple approaches can make DICA applicable in practice to deal with imperfect communications.

2.1. Interaction between a CAV and ICA. Based on the above basic assumptions, we now introduce the interactions between CAVs and ICA in the autonomous and connected intersection traffic considered in this paper. As shown in Figure 2, a CAV and ICA may begin their interaction with each other through vehicular wireless communications if the CAV enters the communication region of ICA. A vehicle is called the *head vehicle* on its lane if there are no vehicles in front of it or the vehicle which is immediately in front of it has begun entering the intersection region. One can see from the figure that a CAV and ICA exchange information through two specific types of messages, REQUEST and RESPONSE, for the ICA's intersection crossing coordination.

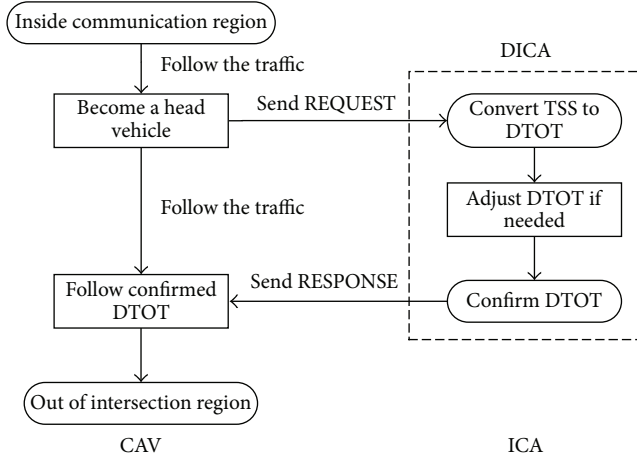


FIGURE 2: Interaction between a CAV and ICA.

A REQUEST message contains information that is necessary for a vehicle's space-time reservation for its intersection crossing, for example, (i) the VIN, (ii) the Vehicle Size (VS) which consists of a vehicle's length and width, and (iii) a vehicle's discrete-time state trajectory, which we name the Timed State Sequence (TSS). A RESPONSE message contains (i) the VIN and (ii) TSS. The TSS in RESPONSE is a feasible TSS processed by ICA. Inside the DICA block in Figure 2, ICA first uses the VS information contained in the received REQUEST message to convert the TSS to the corresponding DTOT which is simply a sequence of timed rectangular spaces that the vehicle needs to occupy to cross the intersection. If the requested DTOT has any potential risk of collision with all other vehicles that have already been approved to cross the intersection, then it is adjusted appropriately to avoid the collisions. Then, ICA confirms the adjusted collision-free DTOT and sends it back to the vehicle. DICA will be introduced in more detail in the next section. In the sequel, we call a vehicle *confirmed vehicle* if it has received a confirmed DTOT from ICA. We assume that each CAV is equipped with a trajectory tracking controller so that the vehicle can follow the confirmed DTOT with reasonably small tracking error as long as the DTOT satisfies the kinematic constraints for feasibility such as maximum speed, acceleration, and deceleration.

2.2. DTOT-Based Intersection Traffic Coordination. This section introduces the high-level idea of how a DTOT is checked and adjusted to avoid potential collisions by ICA. Algorithm 1 which we call the DTOT-based intersection traffic coordination algorithm (DICA) shows the process for ICA to deal with a REQUEST message from a head vehicle.

We introduce the sets and notations that are used in DICA in Algorithm 1. \mathcal{S} is used to denote the set of vehicles which have already been confirmed at the time when the REQUEST message is received by ICA. A pair of vehicles is defined to be *space-time conflicting* if their DTOTs have at least one pair of occupancies that are conflicting in both space and time. Another set \mathcal{C} in the algorithm represents the subset of \mathcal{S} which contains vehicles whose confirmed

DTOTs have space-time conflicts with the DTOT of the vehicle that is currently being processed for confirmation. As shown in Algorithm 1, after converting TSS to DTOT, ICA determines if there exist *front vehicles* that affect the vehicle v^i 's motion and also adjusts v^i 's DTOT if needed by calling the function `checkFV()`. Then, the function `getCV()` is called to determine the initial set \mathcal{C} . The first vehicle v^j in the set \mathcal{C} is the earliest vehicle that is space-time-conflicting with vehicle v^i . The `updateDTOT()` function adjusts $\text{DTOT}(v^i)$ appropriately so that $\text{DTOT}(v^i)$ avoids space-time conflicting with vehicle v^j 's DTOT. These two functions are iteratively called within the `while` loop in Algorithm 1 until the set \mathcal{C} becomes empty, which indicates that no vehicles in the set \mathcal{C} will potentially collide with vehicle v^i . After $\text{DTOT}(v^i)$ is appropriately adjusted and confirmed that there is no space-time conflicting with all other confirmed vehicles, then the confirmed $\text{DTOT}(v^i)$ is converted into $\text{TSS}(v^i)$. Finally, ICA sends RESPONSE with the confirmed $\text{TSS}(v^i)$ back to the vehicle v^i so that the vehicle can cross the intersection safely by following the confirmed DTOT. More detailed explanation on each individual function called within DICA can be found in our previous papers [12, 13].

3. Reactive DICA

In this paper, the problem we want to solve is how to let EVs which are driven autonomously cross an intersection as soon as possible under the connected and autonomous traffic environment. In the meantime, we aim to keep all other vehicles having similar travel times as when there are no EVs in the traffic. In short, our objective is to evacuate EVs through an intersection as quickly as possible while other vehicles' travel times are minimally affected. Note that, for simplicity, the term "emergency vehicle" in this paper means an emergency vehicle in an emergency status (i.e., with siren and the lights on). The same assumptions as in our previous work [12, 13] are employed in this problem. Overtaking and lane changing inside the communication region are not allowed, which means that vehicles on each lane will keep their lane once they enter the communication region. As an approach to give preference to EVs in autonomous traffic, we give priority to EVs in an intersection crossing traffic by optimizing the sequence of crossing vehicles. Also, since we are augmenting the original DTOT-based intersection control algorithm, the new algorithm will only be used to coordinate vehicles when there is an EV within the communication region of an intersection while the crossing traffic is controlled the same way as before when all vehicles are normal vehicles inside the communication region. Thus, the entering of an EV activates the new algorithm, so we call the augmented DICA the *Reactive DICA* (R-DICA). DICA is only taking care of head vehicles which reduces computational complexity and communication load of ICA a lot. However, unlike in DICA, more vehicles are needed to be considered in R-DICA in order to allow EVs to cross an intersection as fast as possible. Specifically, all vehicles on the lane of an EV which are ahead of the EV should be included in the set of vehicles whose intersection crossing order is to

```

(1) Let  $\mathcal{S}$  be the set of confirmed vehicles and  $n = |\mathcal{S}|$ .
(2) Let  $v^j$  be the vehicle to be considered for confirmation.
(3) Convert  $\text{TSS}(v^j)$  to  $\text{DTOT}(v^j)$ 
(4) Call  $\text{checkFV}(\mathcal{S}, \text{DTOT}(v^j)) \rightarrow \text{DTOT}(v^j)$ 
(5) Call  $\text{getCV}(\mathcal{S}, \text{DTOT}(v^j)) \rightarrow \mathcal{C}$ 
(6) while  $\mathcal{C} \neq \emptyset$  do
(7)   Pop the first vehicle in  $\mathcal{C} \rightarrow v^i$ 
(8)   Call  $\text{updateDTOT}(\text{DTOT}(v^i), \text{DTOT}(v^j)) \rightarrow \text{DTOT}(v^j)$ 
(9)   Call  $\text{getCV}(\mathcal{S}, \text{DTOT}(v^j)) \rightarrow \mathcal{C}$ 
(10) end while
(11) Store  $\text{DTOT}(v^j)$  for vehicle  $v^j$ 
(12) Convert  $\text{DTOT}(v^j)$  to  $\text{TSS}(v^j)$ 
(13) Send  $\text{TSS}(v^j)$  to vehicle  $v^j$ 

```

ALGORITHM 1: DICA (DTOT-based intersection traffic coordination algorithm).

be optimized. In the sequel, we call all those vehicles *vehicles on EV's lane*. Thus, the set of vehicles that we need to consider for vehicle ordering includes all unconfirmed vehicles on EV's lane and also all confirmed vehicles which are not on EV's lane. All these vehicles can be divided into two types: vehicles whose DTOTs cannot be modified (vehicles that have already entered the intersection or cannot make a stop at the entering line even with maximum deceleration) and vehicles whose DTOTs could be changed (vehicles that are stopping at the entering line of the intersection or are able to make a stop at the entering line, or unconfirmed vehicles that are ahead of the EV). The sequence of vehicles of the latter type is what we can optimize to expedite the crossing of EVs. We define the set of these vehicles as \mathcal{S}^* .

Roughly speaking, our approach for fast crossing of emergency vehicles is to assign the highest priority to them and delay confirmation for all other normal vehicles. Thus, incorporating a priority based ordering of vehicles into the basic DICA framework would achieve this goal. To find such an optimal vehicle ordering, we formulate an optimization problem based on the *entrance time* of vehicles which is the time a vehicle enters the line of an intersection. Let $\mathcal{P}(\mathcal{S}^*)$ be the set of ordered vehicle sequences (or simply called a *sequence* in the sequel) from the set of vehicles in \mathcal{S}^* . Then, if we use T_e^v to represent entrance time of vehicle v , a reasonable objective function for our optimization problem would be

$$\min_{\mathcal{P}(\mathcal{S}^*)} T_e^{\text{EV}}, \quad (1)$$

where T_e^{EV} is the entrance time of an EV at an intersection. Thus, to solve this optimization problem, we first need to introduce an approach that determines the entrance time of an EV.

First, we note that some sequences in $\mathcal{P}(\mathcal{S}^*)$ can be eliminated if we impose some constraints for optimal vehicle ordering. For example, the order of vehicles on EV's lane cannot be altered and hence should be preserved. Also, since all confirmed vehicles \mathcal{S}^* are able to stop before the entering line of an intersection, we can allocate higher priorities for vehicles on EV's lane than those in other lanes. We use $\overline{\mathcal{P}}(\mathcal{S}^*)$ to denote the set of ordered sequences of vehicles satisfying

these constraints. Now, let us consider a sequence s in the set $\overline{\mathcal{P}}(\mathcal{S}^*)$. Then, if we consider the first vehicle v in the sequence s , then it is easy to see that the vehicle is always a head vehicle on EV's lane and has a confirmed DTOT. Hence, the entrance time of this vehicle v can be determined simply by its $\tau(\mathcal{O}_1^v)$ which is the time when the vehicle v occupies the first occupancy of its DTOT. For any other vehicles which are not the first vehicle in the sequence s , the way of computing their entrance times is a bit different. We need a time interval between any two successive vehicles in a sequence to ensure safety. This time interval is called *separation time* τ_s . In this paper, as shown in Figure 3, we define three separation times for different situations between two vehicles.

$$\tau_s = \begin{cases} \delta_c & v_i \otimes v_j \text{ Figure 3(a), or} \\ \delta_s & v_i < v_j \text{ or } v_j < v_i \text{ Figure 3(b), or} \\ 0 & v_i \odot v_j \text{ Figure 3(c),} \end{cases} \quad (2)$$

where symbols \odot and \otimes are used to represent that two vehicles' routes are compatible and conflicting, respectively. $v_i < v_j$ represents that vehicles v_i and v_j are on the same lane and v_i is following v_j . The separation time's value depends on pavement conditions, vehicle mechanical errors, and weather conditions. The focus of this paper is on proposing a coordination algorithm not the determination of these values. Thus, we just approximate the values from current empirical estimations which are widely accepted [22]. Then, the expression to compute the entrance time of v^j which is not the first vehicle v^1 in the sequence is

$$T_e^j = \max \{T_a^j, T_e^i + \tau_s\}, \quad (3)$$

where v^i is the immediate predecessor of v^j in the sequence and T_a^j is the *predicted arrival time* of the vehicle v^j which is the shortest time for the vehicle to arrive at the entering line of an intersection under the constraints of maximum acceleration and speed without considering other vehicles in traffic. T_e^i is v^i 's entrance time and τ_s is the separation time between v^i and v^j . Starting from the second vehicle in sequence, this equation is iteratively used to compute the

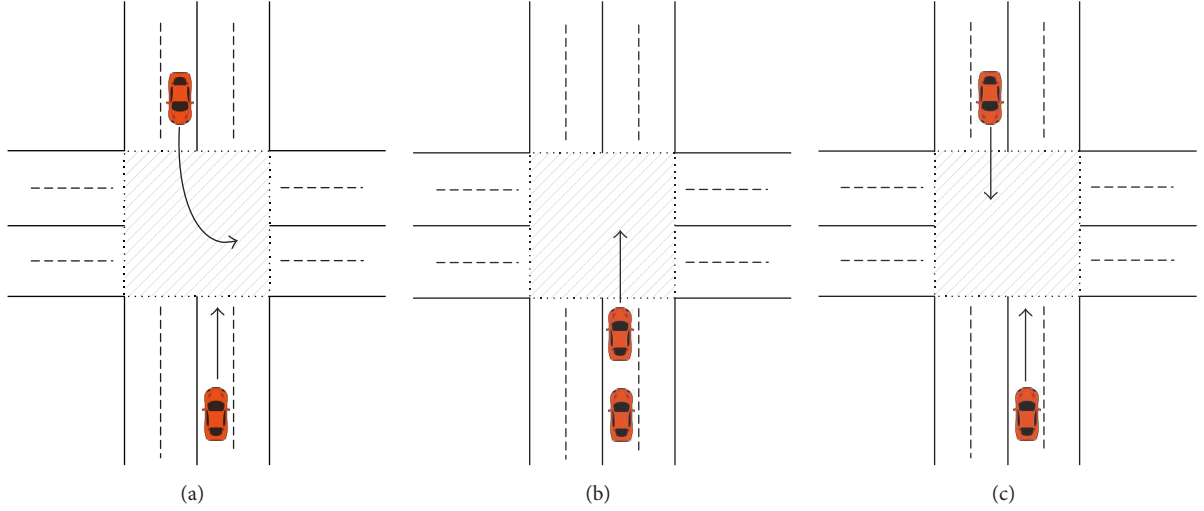


FIGURE 3: Three different situations for separation time.

entrance time of each vehicle in the sequence until the entrance time of the emergency vehicle is computed.

Now, the complete form of an optimization problem for optimal vehicle ordering to minimize the entrance time of the EV is formulated as follows: Given predicted arrival times $T_a^{v_i}$ for all $v_i \in \mathcal{S}^*$, find s^* such that

$$s^* = \min_{s \in \mathcal{P}(\mathcal{S}^*)} T_e^{\text{EV}}$$

$$\text{s.t. } |T_e^i - T_e^j| \geq \begin{cases} 0 & v_i \odot v_j \\ \delta_c & v_i \otimes v_j \\ \delta_s & v_i < v_j \text{ or } v_j < v_i \end{cases} \quad (4)$$

$$T_e^v \geq T_a^v \quad \forall v \in \mathcal{S}^*.$$

A naive approach to solve the optimization problem in (4) is an exhaustive search in all possible sequences that can be generated from the set \mathcal{S}^* . If we suppose that there are n vehicles in \mathcal{S}^* (i.e., $n = |\mathcal{S}^*|$) and there are n_{EV} vehicles on EV's lane, then there are $n!/n_{\text{EV}}!$ sequences in $\mathcal{P}(\mathcal{S}^*)$. However, if n is becoming large, then the computational time and resources required to solve the optimization problem are increasing significantly. Hence, it might not be an efficient approach to use an exhaustive search method when we want to solve problem (4) with many vehicles. Such computation issues of the problem present the need to seek heuristic approaches which are good at solving complex problems in a very short time compared with exhaustive search. Several heuristic optimization approaches like genetic algorithm, ant colony system, and artificial neural networks exist in the literature. Reference [23] used permutation encoding scheme and solved the flow shop scheduling problem with an objective of minimizing the makespan. The paper [21] proposed a genetic algorithm to optimize the groups of compatible vehicles in a very short time. References [24, 25] reviewed many researches where genetic algorithms can be used to solve job scheduling problems which can meet our

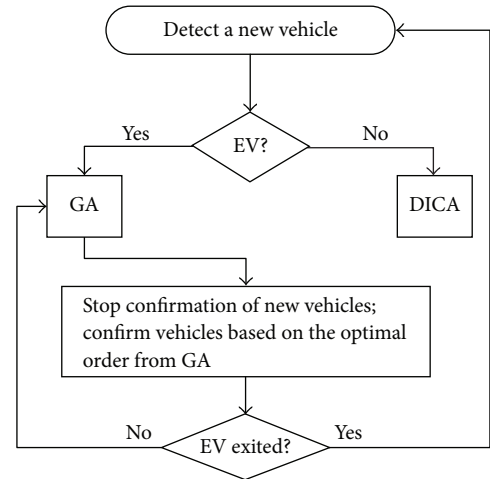


FIGURE 4: Control flow diagram of ICA in R-DICA.

requirements. Thus, we also choose to use genetic algorithm (GA) to obtain the optimal sequence of vehicles.

The high-level architecture of R-DICA combining GA and DICA is shown in Figure 4. R-DICA activates GA when ICA detects an EV. Then, ICA stops accepting any confirmation of new vehicles which are detected after the EV. All vehicles that belong to \mathcal{S}^* are rearranged to obtain the optimal sequence for the EV's crossing by GA. Then, ICA only confirms vehicles that are already included in the set \mathcal{S}^* until the EV exits the intersection. Once the EV is completely out of the intersection, ICA switches back to using DICA to manage normal intersection crossing traffic.

4. Genetic Algorithm for Vehicle Ordering

In this section, we discuss the details of how GA is used to find the optimal vehicle sequence in (4).

```

(1) Generate  $N_{\text{pop}}$  different individuals randomly for  $n$  vehicles in  $\mathcal{S}^* \rightarrow \mathcal{F}$ 
(2)  $\text{feasibilityCheck}(\mathcal{F}) \rightarrow \mathcal{F}$ 
(3)  $k = 0$ 
(4)  $j = 0$ 
(5)  $\text{fitness\_best\_last} = 0$ 
(6)
(7) while  $k < N_{\text{max}}$  and  $j < N_{\text{noChange}}$  do
(8)    $\text{crossover}(P_c, \mathcal{F}) \rightarrow \mathcal{F}$ 
(9)    $\text{feasibilityCheck}(\mathcal{F}) \rightarrow \mathcal{F}$ 
(10)   $\text{mutation}(P_m, \mathcal{F}) \rightarrow \mathcal{F}$ 
(11)   $\text{feasibilityCheck}(\mathcal{F}) \rightarrow \mathcal{F}$ 
(12)
(13)   $\text{fitness\_best}, \text{individual\_best} = \text{fitness}(\mathcal{F})$ 
(14)  if  $\text{fitness\_best} > \text{fitness\_best\_last}$  then
(15)     $j = 0$ 
(16)  else
(17)     $j = j + 1$ 
(18)  end if
(19)  top  $N_{\text{pop}}$  individuals  $\rightarrow \mathcal{F}$ 
(20)   $k = k + 1$ 
(21) end while
(22) Decode  $\text{individual\_best}$ 

```

ALGORITHM 2: Genetic algorithm for vehicle ordering.

Genetic algorithms, which have been widely used to solve problems in computer science, artificial intelligence, information technology, and engineering, are techniques of self-organized and self-adapting artificial intelligence mimicking the evolutionary process of creatures in nature [24, 26]. A solution in GA is called an *individual* which is encoded compactly to facilitate the processes of crossover and mutation that are essential in a genetic algorithm. A group of individuals is called a *population* in which some individuals are selected as parents to generate offspring through crossover and mutation. Based on some features of each individual, some individuals survive and others die among all the original population and new individuals. Individuals who correspond or near correct solution have a better chance to survive during evolving since they have high objective values, which is called *fitness*. Fitness function should be defined properly to evaluate each individual. As introduced above, solutions in GA evolve to adapt the objective problem. Optimal or near-optimal solutions are expected to be obtained after a certain number of generations. In this paper, we propose a GA to solve the complex traffic control problem for emergency vehicles in a short time. Permutation encoding scheme is used in the algorithm. And crossover and mutation operators suitable for the permutation encoding scheme are devised. The proposed GA for vehicle ordering is shown in Algorithm 2. The detailed discussion for permutation scheme, crossover, mutation, and so forth of the proposed GA is given in the following sections.

In the proposed GA, we first generate a random population \mathcal{F} that contains N_{pop} individuals which are encoded by permutation scheme. The function $\text{feasibilityCheck}()$ takes a set of individuals and makes modifications to the infeasible individuals. Feasible individual corresponds to

a sequence of vehicles that does not violate the order of vehicles on EV's lane. After proper modification, the function returns a set containing individuals which are all feasible. The function $\text{crossover}()$ then perform crossover on randomly selected pairs of individuals from the population \mathcal{F} with a probability P_c to generate new offspring. Then, the feasibility of the offspring is checked. Notice that, after crossover, the number of individuals is larger than N_{pop} . Mutation on the produced offspring with probability of P_m is done by function $\text{mutation}()$. The mutated individuals also need to be checked for feasibility and modified if needed. Based on given conditions, each individual in \mathcal{F} is evaluated by a fitness function $\text{fitness}()$ which computes the reciprocal of the entrance time of the emergency vehicle in that individual. The highest fitness value and the corresponding individual are recorded. Notice that the fitness can also be obtained by using other metrics like the exit time of the EV, the trip time of the EV, and so forth. These metrics will give us similar results. We choose the entrance time because we have the predicted arrival time for each vehicle. Thus, it is easy to implement the algorithm. Then, we use the top N_{pop} individuals from the original population and offspring to form the new population. If any of the stopping criteria (maximum number of iterations or the best solution is not updated for a certain number of generations) is met, then the algorithm terminates. Otherwise, the algorithm repeats the steps inside the *while* loop.

4.1. Chromosome Encoding and Feasibility Check. Instead of using the popular binary encoding scheme for genetic algorithms, we choose to use permutation encoding scheme which is more suitable to find an optimal sequence for vehicle

v^1	v^2	v^3	v^4	v^7	EV	v^5	v^6
-------	-------	-------	-------	-------	----	-------	-------

FIGURE 5: An example of the permutation encoding scheme; the leftmost vehicle has the highest priority while the rightmost one has the lowest priority.

ordering. As shown in Figure 5, the individual corresponds to a sequence of vehicles which is $\{v^1, v^2, v^3, v^4, v^7, \text{EV}, v^5, v^6\}$ where the leftmost vehicle v^1 is the first one and the rightmost vehicle v^6 is the last one. Different chromosomes denote different sequences of vehicles. Once an individual is created, it is not always true that the corresponding sequence is a feasible one since vehicles' order on EV's lane cannot be altered. Every newly generated individual should be checked against the subsequence of vehicles on EV's lane for feasibility. Figure 6 is provided to have a visual impression of the situation when vehicles' sequence needs to be optimized. In the figure, v^1 , v^3 , and EV are the vehicles on EV's lane whose order could not be altered. And note that, except for vehicles on EV's lane, all other vehicles that are not a head vehicle are not part of the sequence. The vehicles from south and west that are not head vehicles are such vehicles that will be confirmed only after EV exits. If an individual is not feasible, the corresponding bits of vehicles on EV's lane should be changed to conform to the correct relative order. The function *feasibilityCheck()* is making the corresponding modifications to an infeasible individual. An example of adjustment according to the sequence of vehicles that are ahead of the EV on the same lane is shown in Figure 7.

4.2. Crossover and Mutation. Two individuals perform crossover to generate offspring if they are selected to be parents. The offspring inherit features (i.e., gene structures) from their parents. Different encoding schemes have different crossover operators since they have different gene structures. For the most popular binary encoding scheme, it is easy to do crossover and mutation since a chromosome only contains binary bits. For our permutation encoding scheme, we choose to apply one-point crossover [26] which is implemented in the function *crossOver()*. As shown in Figure 7, the same bits may exist in one chromosome after the parts behind the randomly chosen position are swapped. In the second step in the figure, the two children have the same bits $\{v^2, v^3\}$ and $\{v^5, v^6\}$, respectively. To generate correct chromosomes, we adjust the chromosome of one child by swapping those same bits from another child's chromosome while preserving the relative ordering of parents. Note that the new chromosomes may also not be feasible since the order of vehicles on EV's lane in a chromosome may not be the same as the actual order. If this happens, since the order of the vehicles on EV's lane cannot be changed, we manually adjust the relative order of vehicles to be the correct order to have a feasible chromosome. For example, in Figure 7, we adjust the order of v^1 and v^3 for the second child in the last step. Feasibility check and adjustment are done by the function *feasibilityCheck()*.

Similar to probabilistically selecting two individuals for crossover, we apply mutation on the produced chromosomes

based on a given probability by the function *mutation()*. Different from binary encoding scheme's mutation which could be done by simply changing the value of a randomly selected bit from 1 to 0 or 0 to 1, our permutation encoding scheme exchanges the bits on two randomly chosen positions to obtain a new chromosome. As shown in Figure 8, positions of v^2 and EV are randomly chosen to exchange values and feasibility check based on vehicles' order on EV's lane is performed after mutation.

4.3. Fitness and Selection of New Generation. The reciprocal of the entrance time of the emergency vehicle is defined as the fitness of an individual in our proposed GA. The entrance time of the emergency vehicle can be determined as discussed in Section 3. Among all individuals in the population and the offspring produced, the best N_{pop} individuals are selected to form the next generation.

4.4. Stopping Criterion. The constant N_{max} represents the number of maximum generations and N_{noChange} represents the number of continuous generations where solutions are not changed. As shown in Algorithm 2, if the best solution is not updated after N_{noChange} generations or the N_{max} th generation has been reached, then the algorithm terminates and stops searching for a better solution.

5. Simulation

The performance of the proposed optimization approach for EVs is evaluated against the DICA and a reactive traffic light algorithm which is explained below. All simulations are implemented in an open-source traffic simulator, SUMO (Simulation of Urban MObility) [27]. The default traffic management for intersections in SUMO is not used and the control algorithms are programmed as Python applications. The TraCI is used for the interaction between the Python applications and SUMO. Configurations for intersections in the simulation and corresponding results are described in this section, followed by discussions on the obtained results.

5.1. Simulation Setup. Extensive simulations of different traffic volumes are performed on an isolated perfect 4-way intersection where each approach has three incoming lanes and two exit lanes. Similar to real intersections in the United States, among the three incoming lanes, the leftmost lane is dedicated for left-turn vehicles, and through vehicles can use the other two lanes. The rightmost lane can also be used by right-turn vehicles. All roads have a speed limit of $v_m = 70$ km/h. The maximal acceleration (a_{max}) and deceleration (a_{min}) for all vehicles are set to be 2 m/s^2 and -4.5 m/s^2 , respectively. In the simulation, for simplicity, we used the same size for normal vehicles and EVs so that they both have 5-meter length and 1.8-meter width. We let vehicles approach an intersection with different speeds when they enter into the communication region of the intersection to make the simulation more realistic. In detail, when a new vehicle is spawned outside of the communication region, the speed of the vehicle is set with a random value within the range from 40% to 100% of the maximum allowed speed v_m . In those

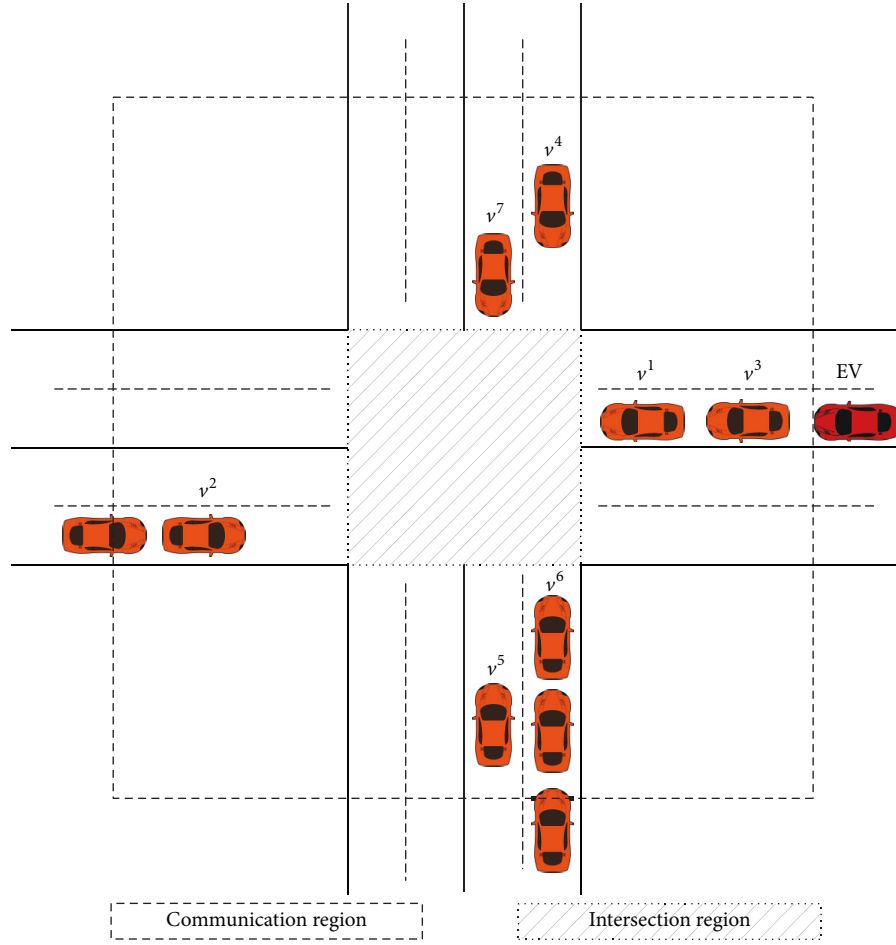


FIGURE 6: Example situation of vehicles whose sequence is to be optimized in the intersection space; note: vehicles on EV's lane are v^1 , v^3 , and EV.

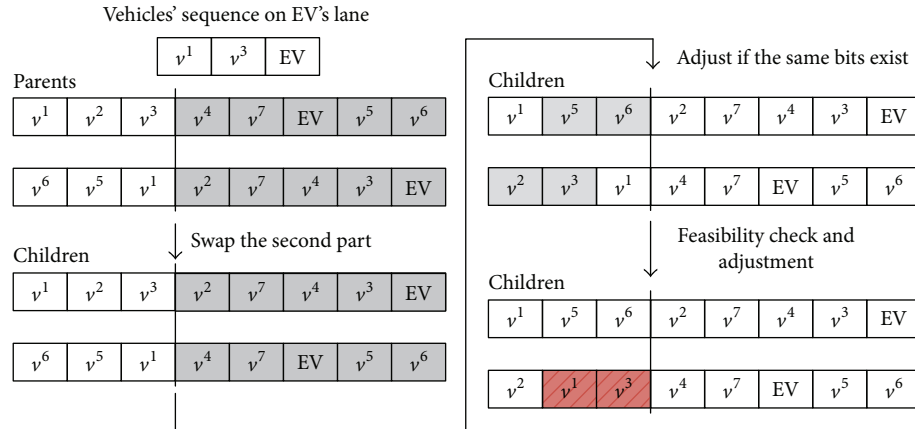


FIGURE 7: An example of one-point crossover. The relative ordering of parents is preserved when the chromosomes are adjusted due to the existence of the same bits. Feasibility is checked for the two children based on vehicles' sequence on EV's lane and adjustments are made.

cases where vehicles need to stop just before the entering line of the intersection region to avoid potential collisions with other confirmed vehicles, the distance between the entering line of the communication region and the entering line of the intersection region should be long enough for a vehicle to

be able to stop from the maximum allowed speed v_m . Thus, it is easy to conclude that the distance should be at least $-v_m^2/(2a_{\min}) \approx 42.01$ m. So, in simulation, we set the distance between the entering lines of the communication region and the intersection region as 50 m.

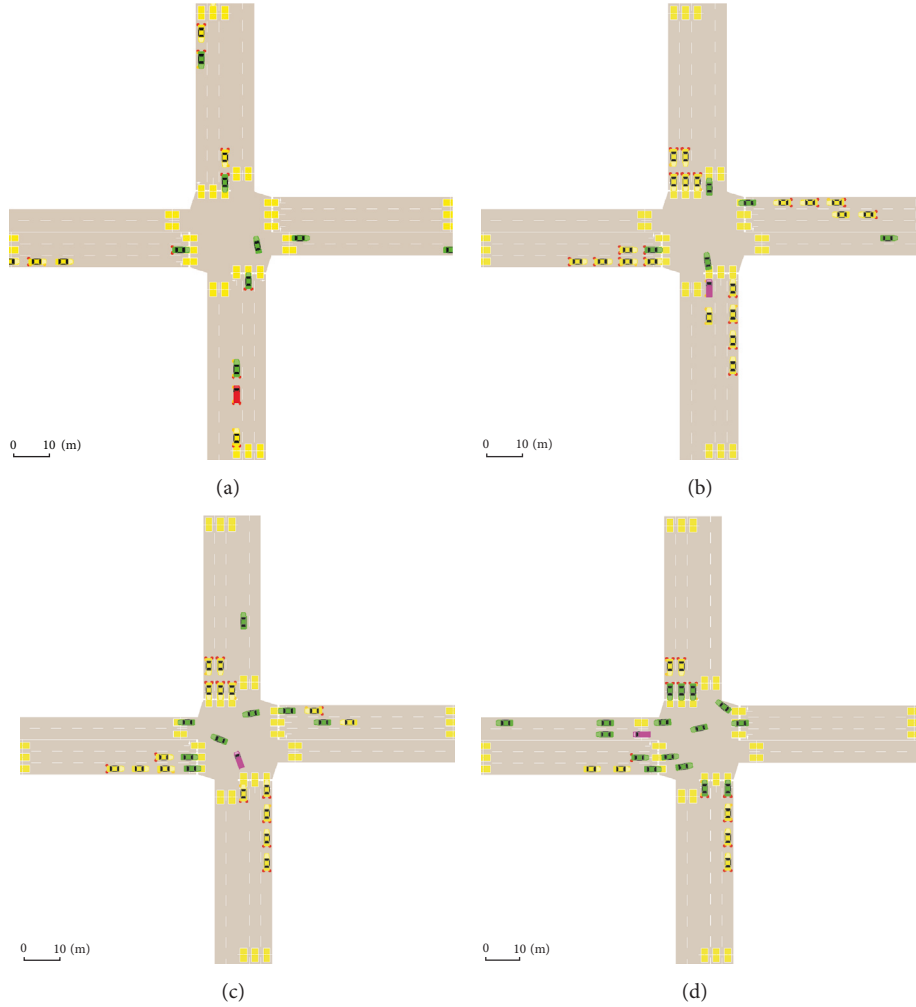


FIGURE 10: A series of screenshots of simulation which illustrates a situation when an EV is crossing the intersection.

TABLE 2: Computation time comparison between ES and GA.

Traffic volume (number of vehicles per 10 minutes)	100	200	300	400	500
Computation time of ES (h)	0.05	0.52	N/A	N/A	N/A
Computation time of GA (h)	0.05	0.15	0.35	0.65	0.72

exits the intersection, all head vehicles get confirmed, which means R-DICA operates the same way as DICA. The optimal vehicle-passing sequence from the genetic algorithm ensures the fast crossing of an intersection for the EV.

5.3.1. Computation Time. To show the computational efficiency of R-DICA using GA, we implemented R-DICA in two different versions: one with GA and the other one with the exhaustive search (ES) method to solve the optimization problem. Computation times of different volume cases are recorded for both methods. Simulation results are shown in Table 2 where “N/A” means that the computer was not able to complete the simulation due to memory errors.

From the result, we can see that R-DICA with GA has definite advantages over R-DICA with ES in terms of computational efficiency. As shown in the table, the exhaustive search method only works for light traffic volumes while it has memory issues for traffic of higher volumes.

5.3.2. Performance of EVs. The following performance measures are obtained to compare the performance of R-DICA with DICA and the reactive traffic light: a vehicle’s *trip time* (τ) is defined as the time taken for a vehicle from the moment when it enters into the communication region of an intersection until the vehicle exits the intersection. Based on the measurement of τ for all crossed vehicles, we obtained the *average trip time* ($\bar{\tau}$) and the *maximum trip time* (τ_m) which show the performance of the crossed vehicles. Besides these performance measures, we also calculated *throughput* (ρ) which is the percentage of all crossed vehicles against the total number of generated vehicles. We calculated the rate of average trip time to throughput, which we call the *effective average trip time* ($\bar{\tau}_e$). The detailed explanation for this metric can be found in [12].

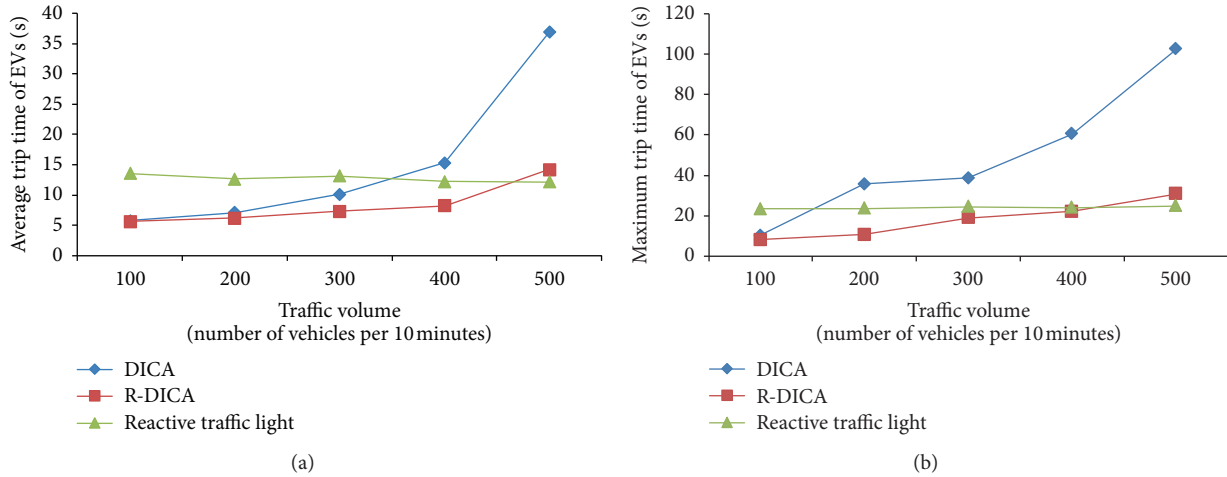


FIGURE 11: Performance comparison of EVs for DICA, R-DICA, and the reactive traffic light: (a) average trip time; (b) maximum trip time.

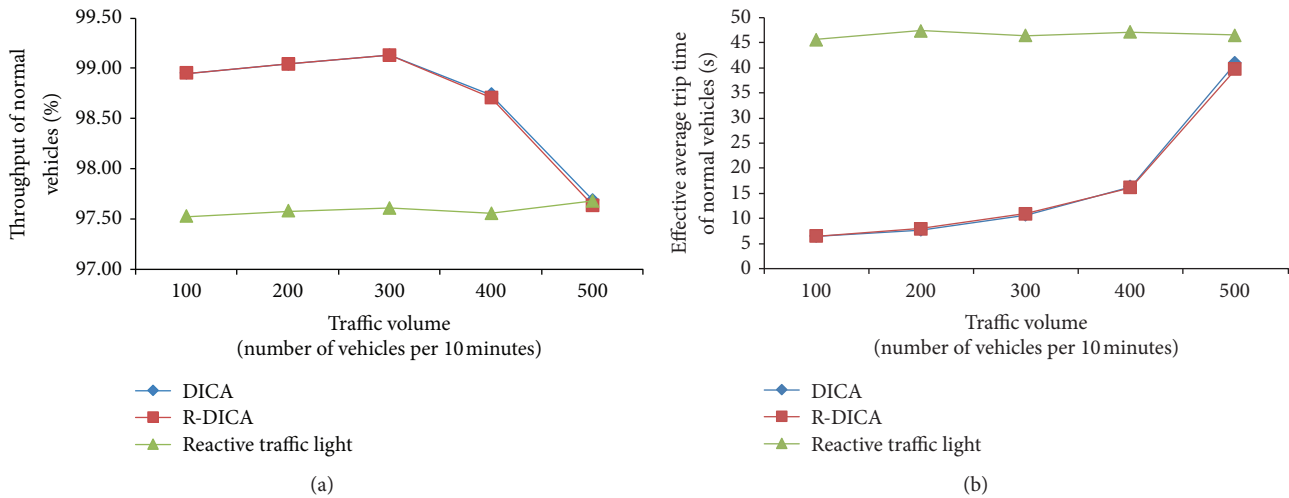


FIGURE 12: Performance comparison of normal vehicles for DICA, R-DICA, and the reactive traffic light: (a) throughput; (b) effective average trip time.

As shown in Figure 11, the average trip times of EVs in all three algorithms (DICA, R-DICA, and the reactive traffic light) are compared. For traffic volumes from 100 to 400, R-DICA has the least average trip time of EVs compared with the other two algorithms. Especially in light traffic volumes, R-DICA reduces the EVs' average travel time by more than 50% from the reactive traffic light. However, the algorithm has a bit longer average trip time for EVs than that of the reactive traffic light in 500 traffic volume. The worst case for EVs' travels is illustrated by the maximum trip time of EVs in Figure 11(b). The maximum trip time of R-DICA increases and becomes greater than that of the reactive traffic light. Both average trip time and maximum trip time of EVs for DICA are increasing along the volumes. One may note that the average trip time and the maximum trip time of the reactive traffic light keep almost the same with the increase of the traffic volume. Through observation of the simulations, part of this is because too many vehicles accumulate before the intersection when the lane of the EV is under red light. In

this situation, the EV is not detected and is stopping outside the communication region. When the light for the lane of the EV turns green, the EV accelerates from rest to enter the communication region which results in a higher speed for the EV. Thus, for the heavier traffic volumes, EVs always have a higher speed when detected and are expedited to cross by preference. The trip time within the communication region is then reduced compared with R-DICA.

5.3.3. Performance of Normal Vehicles. Comparison of the performance for normal vehicles for all three algorithms is shown in Figure 12. From this result, we can see that the throughput and effective average trip time of R-DICA are nearly the same as those of DICA, which shows that the performance of normal vehicles is minimally affected by EVs. Both the throughput and effective average trip time of normal vehicles become worse with the increase of traffic volumes. This is consistent with the result in our previous work [12, 13]. Also, one can see from Figure 12 that the reactive traffic light

TABLE 3: Comparison of maximum trip times of normal vehicles between DICA and R-DICA.

Traffic volume (number of vehicles per 10 minutes)	100	200	300	400	500
Maximum trip time of normal vehicles (DICA) (s)	17.88	35.80	55.37	89.47	132.90
Maximum trip time of normal vehicles (R-DICA) (s)	18.62	38.52	61.68	99.15	150.00

has steady and worse performance for normal vehicles than the other two algorithms.

To more investigate the negative effect of prioritizing EVs on other normal vehicles, we compare the maximum trip time of normal vehicles for DICA and R-DICA in Table 3. The maximum trip time of R-DICA is very close to that of DICA and their difference increases with traffic volumes. This shows that it will bring about more negative effect on normal vehicles to evacuate an EV in congested traffic.

6. Conclusion

In this paper, we have shown that the DICA algorithm can be augmented to allow emergency vehicles to cross intersections faster. A genetic algorithm based approach is proposed as part of the augmented algorithm, called R-DICA, to optimize the sequence of vehicles which gives the emergency vehicle the highest priority and keeps the influence on other vehicles' travel times as minimum as possible. The R-DICA operates the same way as DICA if there is no EV inside the communication region and optimizes vehicle-passing sequence if an EV enters the communication region. Reactive traffic light and DICA algorithms are also implemented for simulation and their results are compared with R-DICA to evaluate the performance of R-DICA. Simulation results show that R-DICA is effective in reducing travel times of EVs and has better performance than the reactive traffic light for normal vehicles. We conclude that the performance of normal vehicles is not noticeably affected based on the simulation results of DICA and R-DICA. Currently, it is in progress to optimize each vehicle's trajectory to achieve the maximum traffic control performance. In the future, we will relax assumptions like perfect communication and accurate prediction of DTOT to make the algorithm more applicable to real situations. Also, DICA will be generalized to include nonautonomous traffic like human-driven vehicles and pedestrians.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, 2000.
- [2] DARPA, *The darpa urban challenge*, 2007, <http://archive.darpa.mil/grandchallenge/>.
- [3] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.
- [4] P. Tientrakool, Y.-C. Ho, and N. F. Maxemchuk, "Highway capacity benefits from using vehicle-to-vehicle communication and sensors for collision avoidance," in *Proceedings of the IEEE 74th Vehicular Technology Conference, VTC Fall*, The Hilton San Francisco Union Square, San Francisco, USA, September 2011.
- [5] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.
- [6] P. Dai, K. Liu, Q. Zhuge, E. H.-M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-experience-oriented autonomous intersection control in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1956–1967, 2016.
- [7] M. A. S. Kamal, J.-I. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1136–1147, 2015.
- [8] D. Miculescu and S. Karaman, "Polling-systems-based autonomous vehicle coordination in traffic intersections with no traffic signals," 2016, <https://arxiv.org/abs/1607.07896>.
- [9] W. G. Wu, J. B. Zhang, A. Luo, and J. N. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 65–74, 2015.
- [10] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *Proceedings of the 2016 American Control Conference, ACC 2016*, pp. 6227–6232, usa, July 2016.
- [11] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 2014.
- [12] Q. Lu and K. Kim, "Intelligent intersection management of autonomous traffic using discrete-time occupancies trajectory," *Journal of Traffic and Logistics Engineering*, 2016.
- [13] Q. Lu and K.-D. Kim, "Autonomous and connected intersection crossing traffic management using discrete-time occupancies trajectory," 2017, <https://arxiv.org/abs/1705.05231>.
- [14] F. J. Martinez, C.-K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Emergency services in future intelligent transportation systems based on vehicular communication networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 2, pp. 6–20, 2010.
- [15] E. Oliveira and N. Duarte, "Making way for emergency vehicles," in *Proceedings of the 2005 European Simulation and Modelling Conference*, vol. 128, p. 135, 2005.
- [16] W. Viriyasitavat and O. K. Tonguz, "Priority management of emergency vehicles at intersections using self-organized traffic control," in *Proceedings of the 76th IEEE Vehicular Technology Conference, VTC Fall 2012*, can, September 2012.
- [17] K. Dresner and P. Stone, "Human-usable and emergency vehicle-aware control policies for autonomous intersection

- management,” in *Proceedings of the Fourth International Workshop on Agents in Traffic and Transportation (ATT)*, Hakodate, Japan, 2006.
- [18] D. Carlino, S. D. Boyles, and P. Stone, “Auction-based autonomous intersection management,” in *Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems: Intelligent Transportation Systems for All Modes, ITSC 2013*, pp. 529–534, nld, October 2013.
 - [19] X. Qian, J. Gregoire, F. Moutarde, and A. De La Fortelle, “Priority-based coordination of autonomous and legacy vehicles at intersection,” in *Proceedings of the 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, pp. 1166–1171, chn, October 2014.
 - [20] J. Wu, A. Abbas-Turki, and A. El Moudni, “Cooperative driving: An ant colony system for autonomous intersection management,” *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, 2012.
 - [21] F. Yan, M. Dridi, and A. E. Moudni, “An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach,” *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 183–200, 2013.
 - [22] X. Chen, L. Li, and Y. Zhang, “A markov model for headway/spacing distribution of road traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 773–785, 2010.
 - [23] T. Murata, H. Ishibuchi, and H. Tanaka, “Genetic algorithms for flowshop scheduling problems,” *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 1061–1071, 1996.
 - [24] E. Hart, P. Ross, and D. Corne, “Evolutionary scheduling: a review,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 191–220, 2005.
 - [25] F. Werner, “Genetic algorithms for shop scheduling problems: a survey,” Preprint 11-31, FMA, OvGU Magdeburg, 66 p., 2011.
 - [26] K. Dahal, K. C. Tan, and P. I. Cowling, *Evolutionary scheduling*, vol. 49, Springer Science Business Media, 2007.
 - [27] K. Daniel, E. Jakob, B. Michael, and B. Laura, “Recent development and applications of sumo - simulation of urban mobility,” in *Proceedings of the International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

