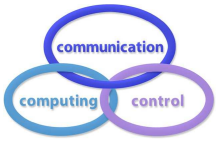
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 6, Month: December, Year: 2025
Article Number: 6677, https://doi.org/10.15837/ijccc.2025.6.6677



CCC Publications

Decentralized Right-of-Way Protocol for Connected and Automated Vehicles at Unsignalized Intersections

G. H. F. Diédié, B. S. Zounémé, T. N'Takpé

Gokou Hervé Fabrice Diédié*
Laboratoire de Mathématiques et d'Informatique
Université Peleforo Gon Coulibaly, Ivory Coast
BP 1328, Korhogo, Ivory Coast
*Corresponding author: herve.diedie@upgc.edu.ci

Boris Stéphane Zounémé
Laboratoire de Mathématiques et d'Informatique
Université Nangui Abrogoua
02 BP 801, Abidjan, Ivory Coast
zounemeste.sfa@univ-na.ci

Tchimou N'Takpé
Laboratoire de Mathématiques et d'Informatique
Université Nangui Abrogoua
02 BP 801, Abidjan, Ivory Coast
tchimou.ntakpe.sfa@univ-na.ci

Abstract

In urban areas, traffic mainly depends on decisions made by drivers, especially at junctions. The advent of Connected and Automated Vehicles aims to prevent driver error susceptible to causing accidents or congestion. This problem becomes more challenging when the intersection is signal-free. Numerous solutions have addressed this question, also called the conflict-free scheduling problem. However, most of them do not consider handling simultaneously spillbacks, vehicle heterogeneity, and prioritization. Besides, some of these contributions are limited only to pedestrians and ignoring other vulnerable road users such as cyclists and motorcyclists. These shortcomings are detrimental to any real-world efficient implementation. This paper aims to cope with these issues concurrently. To this end, we formulate this question as a Conflict-free and Precedence-Constrained Multiple Knapsack Problem (CPCMKP) solved through a distributed heuristic. Simulation results showed that this solution outperforms related state-of-the-art schemes concerning delay and fairness.

Keywords: Connected and Automated Vehicles, Unsignalized Intersection, V2X, PCKP.

1 Introduction

Urbanization is synonymous with social, economic, and environmental transformations. Managing their impact is a key challenge for the movement of people and goods. Such a mobility has benefited enormously from the recent advances in the realm of Intelligent Transportation Systems (ITS) [1]. Many solutions have been proposed especially regarding safety and traffic management. Using Connected and Autonomous Vehicles (CAVs) is undoubtedly one of the most promising in reducing congestion, accidents, or pollution, and thus, enhancing overall transportation efficiency [2, 3, 4]. These CAVs refer to vehicles that can operate without human intervention and communicate with each other or with the surrounding infrastructure using wireless technologies [5]. This V2X (Vehicle-to-Everything) communication paradigm generally implies using signal-free intersections. Such a policy also known as Autonomous Intersection Management (AIM) [6], endeavors to help CAVs interact to schedule their rights-of-way. This cooperative decision-making problem becomes more challenging when trying to maximize traffic flow [7].

AIM is intensively studied in the literature. In recent years, various methods have been proposed by researchers [8]. In this paper, we focus mainly on the optimization-oriented solutions. The latter leverage Operations research-based methods to address inherent issues such as spillbacks. This phenomenon occurs when the intersection is blocked by upstream traffic that is prevented from progressing by a downstream one. Spillbacks result in a rapid propagation of congestion to the entire intersection [9].

Another hot topic is pedestrian handling. Indeed, particularly in urban areas walking humans and vehicles share the same space. An effective management of these interactions is crucial to real pedestrian-friendly infrastructure especially when vehicles are autonomous.

The third important question is prioritization. A good AIM should be able to classify road users then define accordingly a relevant passage order. Priorities must be determined while still reducing sojourn times and improving throughput. In addition to vehicles, access should be fairly granted to pedestrians as well as other Vulnerable Road Users (VRUs) such as motorists, animals, cyclists, and moped riders [10]. Most of the existing solutions struggle to address the above-mentioned problems concurrently. Such a limitation holds back any real-life deployment and thus prevents their usefulness. We propose an optimization-based AIM scheme which tackles spillbacks prioritization and VRU handling.

The main contribution of this paper is threefold:

- formulation of this spillback-aware prioritization and VRU-oriented AIM question as a novel combination of the famous 0-1 Multiple-Knapsack Problem (MKP)[11] and Precedence Constrained Knapsack Problem (PCKP)[12]. We call it the 0-1 Conflict and Precedence-Constrained Multiple Knapsack Problem (CPCMKP);
- design of a distributed heuristic which helps to mitigate spillbacks and achieve optimal throughput ;
- extensive simulation campaign to prove that our solution can minimize delay while ensuring fairness, compared to existing similar studies.

The remainder of this paper is organized as follows: Section 2 reviews related works on AIM. In section 3, we detail our proposition. Section 4 presents the performance evaluation process, its results, their analyses, and their discussions. Section 5 concludes this paper and discusses some possible future work.

2 Related work

In recent years, different AIM solutions have been proposed with various objectives such as safety, efficacy, passenger ease, ecology [13]. To do this, these studies generally consider lowering delay, increasing throughput, and prevent collisions by scheduling the rights-of-way of the vehicles. The techniques commonly used for the traffic scheduling also called trajectory planning [14] or cruise

control [8], can be categorized mainly according to the intersection reservation scheme or the priority policy used.

With respect to the reservation scheme, four strategies are used to avoid conflicts. They are categorized into intersection-based, tile-based, conflict points-based, and vehicle-based [10]. The intersection-based reservation allows only one CAV at a time within the conflict zone. In the tile-based strategy, the space is discretized into a grid where two CAVs must not occupy a cell at the same time. Conflict points-based scheme requires the CAVs to follow pre-defined trajectories where the conflictual ones result in several collision points. Each solution is aimed to not simultaneously allow conflictual trajectories. As for the vehicle-based techniques, they leverage the kinematics of the CAVs to schedule their movements and avoid collisions. Besides the reservation schemes some CAVs use their stop-and-go-based or platoon-based scheme to cross the intersection. Using stop-and-go a CAV receives a stop sign until another one leaves the conflict zone; while in platoon-based strategy a CAV considers a second one as a virtual obstacle and adjusts its speed accordingly [8].

Regarding the priority policies often used by the AIM solutions, they can be grouped into four main categories: First-Come First-Served (FCFS)-based, system-level performance-based, queue length-based, vehicle type-based, auction-based also called First-Ready Out), and Time To React-based [8]. The first two categories are the most commonly used in the literature. The FCFS or its variation FIFS (First In First Served)-based policy are the synonyms of the well-known FIFO(First In First Output) policy applied in the queueing theory; whereas the system-level performance-based policies also called optimization-based [13] leverage global metrics such as the total delay, the travel time or the throughput to grant rights-of-the-way to the CAVs.

Based on the number of entities involved in the decision making process, AIM solutions can also be categorized into centralized, decentralized, and distributed [14]. Therefore, the solution is said to be centralized when traffic information is sent to a single device that schedules the crossings. In decentralized solutions all the nodes cooperate to have the right-of-way whereas in distributed schemes each CAV makes decisions solely to reach its own goal.

From modeling perspective AIM solutions commonly leverage schemes from various fields such as operations research (optimization), queuing theory, game theory, graph theory, and machine learning. However, most of the contributions are optimization-based [13]. In this paper we will focus on such solutions.

Wang *et al.* [15] considered to dynamically assign lane to CAVs leveraging the congestion level to reduce the average delay. A pipeline-based strategy is used to handle the parallel communications between the CAVs. Each lane is divided into four zones and as many stages in the decision process. CAVs gradually go through the lane-assignment stage, the preparing stage, then the scheduling stage, and the reacting zone before entering the intersection. A grid-based reservation technique is used to estimate the congestion level of the trajectories and to schedule the vehicles. Unfortunately, due to the number of zones it requires, this scheme cannot be applied to short roads. The same authors propose to improve their solution to consider the pedestrians, on-demand road crossing requests and, the spillback problem. But the main shortcomings still exist. Indeed, the number of zones has even been increased to five ; the crossing times of all the VRUs including the pedestrians are fixed. Moreover, the decision making process ignores the number these road users.

Li *et al.*[16] proposed a decentralized and priority-based framework where AIM is formulated as a multi-CAV trajectory optimization problem. A* the well-known pathfinding algorithm is adopted to generate a trajectory for each CAV. Unfortunately, this reservation scheme is a vehicle-only-based solution.

Bakibillah *et al.* [17] suggested a bi-level scheme consisting of two levels of control. This strategy is implemented using two zones namely the clustering zone and merging-execution zone. Indeed, the first level (the higher one) groups the approaching vehicles based on traffic state. Whereas the lower level calculates the vehicles' optimal sequences merging times based on a predictive technique known as the Receding Horizon Control (RHC). However, this platooning scheme is dedicated only to single-lane roundabouts and does not consider VRUs.

Chalaki *et al.* [18] formulated AIMs as an energy-optimal control problem with interior-point constraints through a decentralized bi-level strategy. Each lane is split into three zones; namely, the

lane-changing zone, the control zone, and the merging one. The upper level enables each CAV in the control zone to determine the optimal speed variation that can minimize energy consumption and improve traffic throughput by eliminating stop-and-go-driving. To do this, each CAV calculates the arrival time at each merging zone; then, the exit times of merging zones become the interior-point constraints for the lower level solve the optimal control problem. Regrettably, VRUs are not handled by this framework. The same authors proposed a similar solution to minimize the total travel time [19] but still with the same shortcoming.

Pei *et al.* [20] suggested a dynamic programming scheme using a transition state strategy to assign the right-of-way to several vehicles at a time. Each transition state corresponds to a specific kind of conflicts between vehicles satisfying the Markov property. Regrettably, this solution does not scale.

Zhang *et al.* [21] presented a decentralized priority-based scheme. A multi-objective function is designed to get the optimal trajectories. Each lane is split into two zones. CAVs in the near zone use the FCFS (First Come First Served) rule. The crossing oder of emergency CAVs is based on the arrival time. Unfortunately, this solution does not consider either VRUs or spillbacks.

Chen *et al.* [22] suggested a crossing point-based strategy. The CAVs' trajectory conflict relationship is modeled as an undirected graph. Then an improved depth-first spanning tree algorithm is used to find the local optimal crossing order for each vehicle. Afterwards, a virtual platoon-based minimum clique cover algorithm is applied to identify the global optimal solution. A distributed feedback control method is designed to apply the results. Regrettably, neither VRUs nor spillbacks are considered.

Deng *et al.* [23] suggested a bi-level optimization scheme with a rolling horizon-based process to balance traffic performance and computational efficiency. The lower-level generates dynamically-feasible and energy-efficient trajectories to prevent spillbacks. However, the proposed scheme requires CAVs to form platoons before entering the conflict zone. Nevertheless, this scheme could significantly increase delay.

Niels *et al.* [24] proposed a conflict point-based approach combining signal-free vehicle control with pedestrian signal phases. Each lane is split into two zones. A rolling horizon-based scheme is implemented to define the crossing order. But to resolve the conflicts, only one vehicle or a group of pedestrians is allowed within the conflict region at a time. Moreover, spillbacks are not handled. Other VRUs are not considered while pedestrian crossing time is fixed irrespective of their number.

3 Proposed solution

In this section we discuss the motivations and objectives of this work. Then, we present the assumptions before detailing our solution. The latter will be referred to as UICP (Unsignalized Intersection Crossing Protocol) in the rest of this paper. Note that UICP is asynchronous, message-passing and works under a distributed unfair daemon.

3.1 Motivation and Objectives

As shown in section 2, most existing solutions surprisingly leverage unrealistic assumptions (lanes with infinite capacity, vehicle-only rights-of-way, fixed crossing time etc.). Such considerations hinder their efficiency and prevent any real-world implementation. We believe that a realistic AIM should be, fully decentralized, spillback-aware, and able to handle all the VRUs. Besides it should consider both vehicles' priority and traffic heterogeneity. To the best of our knowledge, such a solution does not exist in the literature. Therefore, we aim to design a similar hybrid scheme that avoids collisions, minimizes delay while ensuring fairness.

3.2 Assumptions

We assume that all the vehicles are CAVs equipped with wireless units to communicate among them and with the infrastructure. Conversely, that the latter can:

- get and provide information such as the Euclidean distance between lanes, the number of lanes, lanes' lengths, traffic movements' conflicts;

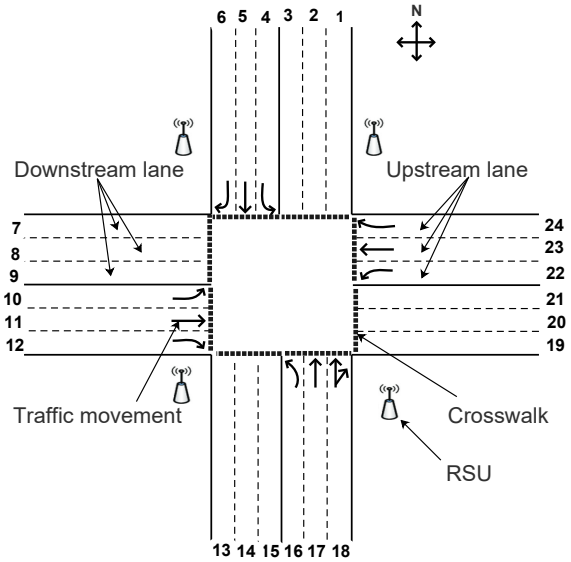


Figure 1: A typical unsignalized four-legged intersection with three lanes and four crosswalks.

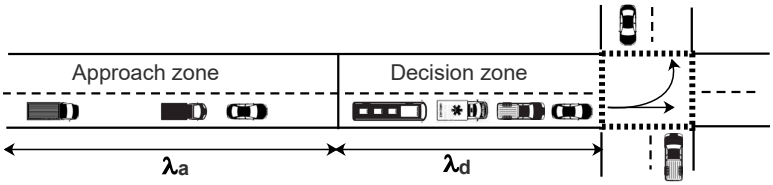


Figure 2: Illustration of the two zones considered by UICP.

- estimate real-time traffic parameters such as CAVs' arrival rates, the number of queued CAVs, CAV's lengths, inter-vehicle mean distance, lanes' residual capacity;

- assess the number of waiting VRUs and collect their crossing requests.

Figure 1 depicts a typical intersection considered by UICP. Let O, I, and J be three sets that respectively denote the origins (i.e. the upstream lanes), the CAVs in the decision zones of each upstream lane, and the possible destinations (i.e. the approach zones of the downstream lanes) ; with $|I|= n$ and $|J|= m$.

Definition 1 (Traffic movement)**.** *A traffic movement α or simply a movement is a possible displacement (clearance) of a vehicle from an upstream lane to a downstream one. Formally, $\alpha = (i, j) \in O \times J$.*

Definition 2 (Crossing area)**.** *The crossing area $\mathcal{M}$ is the set of all the traffic movements. Formally, $\mathcal{M} = \{(i, j) \in O \times J\}$.*

Note that each lane i with length λ_i is divided into two zones, namely the approach zone and the decision zone respectively lengths λa and λd defined as parameters. Such as the Equation (1)

$$\lambda_i = \lambda a + \lambda d \tag{1}$$

3.3 Vehicle demand handling

UICP is aimed to find a right-of-way for the CAVs in the decision zones (see Figure2) that avoids collisions, maximizes throughput, and minimizes their waiting times. To do so, each CAV $i \in I$ is defined by the tuple $\{\ell_i,\pi_i, o_i, \delta_i\}$ where ℓ_i, π_i, $o_i \in O$, and $\delta_i \in J$ respectively denote its length, its priority index, its lane (i.e. its origin), and its destination.

Note that the priority indices are defined by the underlying application according to vehicles' types.

$$t_{prd}(i) = \mathcal{U}(\mu 1; \mu 2) \tag{2}$$

Periodically after $t_{prd}(i)$ seconds, each CAV i estimates the euclidean distance to the exit of its lane according to its current position (see Figure 2). $t_{prd}(i)$ is obtained using Equation (2) where $\mu 1$ and $\mu 2$ respectively denote two fixed parameters such as $\mu 1 < \mu 2$ and $\mathcal{U}(.)$ refers to the Uniform distribution. If the estimated distance is inferior to λa CAV i must send to its neighborhood a LEAD_REQ message containing its ID (id_i) and the tuple of variables defined above. CAV i will wait for any response during $t_{lead}(i)$ seconds. The latter duration is calculated using Equation (3) where ψ, κ, $\widehat{d}_i$ and c respectively denote the length of the message, the bit rate, the euclidean distance to lane exit, and the propagation speed.

$$t_{lead}(i) = 2 \times (\frac{\psi}{\kappa} + \frac{\widehat{d}_i}{c}) \tag{3}$$

After receiving a LEAD_REQ message a CAV j must send a LEAD_ACK message as a reply only if its lanes' index o_j is equal to that of CAV i (i.e. o_i). The reception or not of such a LEAD_ACK message helps a CAV i know its status namely, leader or follower. If CAV i is a leader then it must check (from the closest RSU) if there is any pending crossing session. If so, it must wait until the end of this crossing session before initiating a new one, by broadcasting a CROSS_REQ message. By contrast, if there is no active crossing session, CAV i must initiate a new one immediately.

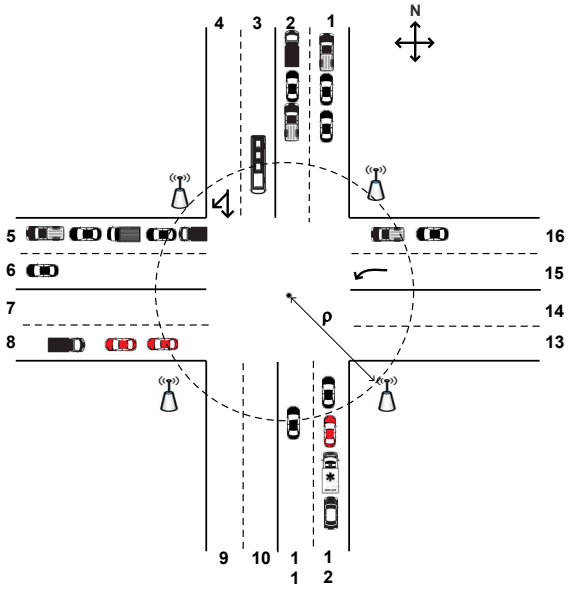


Figure 3: Illustration of the conflict zone and the resulting range denoted by ρ.

Note that CROSS_REQ and LEAD_REQ messages contain the same information. Only leaders must start the data collection process on their lanes when receiving a CROSS_REQ message. To do so, the leader must send a DATA_REQ containing the same information as CROSS_REQ, to its followers then wait for $t_{data}(i)$ seconds. This duration is calculated via Equation (4) where λd_{o_i} is the length of the decision zone of CAV i 's lane.

$$t_{data}(i) = 2 \times (\frac{\psi}{\kappa} + \frac{\lambda d_{o_i} - \widehat{d}_i}{c}) \tag{4}$$

On receiving a DATA_REQ message, a CAV must check if the sender is on its lane. If so, and if this CAV has two neighbors, it must forward this message to the second neighbor i.e. not to the one that has forwarded it. This is a scheme commonly used to mitigate the message overhead on a linear topology. However, if the follower has only one neighbor, it must reply with a DATA_ACK message destined to the leader by adding its own quadruple variables as discussed above. DATA_ACK and DATA_REQ messages are forwarded using the same scheme. Except that each rightful forwarder must add its own information before sending the message. After $t_{data}(i)$ seconds, the leader i will broadcast all the data collected from its followers in addition to its own variables via a CROSS_ACK message. The latter message is destined to the other leaders. Then CAV i will in turn trigger a timer

and wait for $t_{cross}(i)$ seconds. This duration is estimated using Equation (5)where ρ is the range of the conflict zone (see Figure 3).

$$t_{cross}(i) = 2 \times (\frac{\psi}{\kappa} + \frac{\widehat{d}_i + \rho}{c}) \tag{5}$$

Note that without loss of generality, followers' data on a lane could also be provided by the RSUs.

After the expiration of its t_{cross}, a leader must determine the right-of-way for each vehicle in the decision zones based on data received from other leaders. To this end, we model an isolated unsignalized intersection as a set of m knapsacks (i.e. the downstream lanes) which must be filled with the most valuable items among n ones. Knowing that each item i has a profit (i.e. fitness score) $p_i > 0$ and consumes an amount $w_i \geq 0$ resources (i.e. vehicle's length $w_i = \ell_i$) from each knapsack j with capacity $C_j > 0$. The goal is to maximize the sum of profits of the items so that the sum of weights in each knapsack j does not exceed C_j.

Indeed, we formulate this CAV's right-of-way issue as a combination of the well-known 0-1 Multiple-Knapsack Problem (MKP) [25] [11] and the Precedence Constrained Knapsack Problem (PCKP)[12]. We refer to this question as the 0-1 Conflict and Precedence-Constrained Multiple Knapsack Problem (CPCMKP) and model it via the following integer linear program:

Let x_{ij}=$\begin{cases} 1\,, \text{if CAV } i \text{ can move to destination } j \\ 0\,, \text{otherwise} \end{cases}$

Let a_{ik}=$\begin{cases} 1\,, \text{if movements of CAVs } i \text{ and } k \text{ are conflicting} \\ 0\,, \text{otherwise} \end{cases}$

Let b_{ik}=$\begin{cases} 1\,, \text{if CAV } i \text{ precedes CAV } k \text{ on the same lane} \\ 0\,, \text{otherwise} \end{cases}$

$$\max \sum_{i \in I} \sum_{j \in J} p_i x_{ij} \tag{6}$$

s.t. :

$$\sum_{i \in I} w_i x_{ij} \leq C_j, \forall j \in J \tag{7}$$

$$\sum_{j \in J} x_{ij} \leq 1, \forall i \in I \tag{8}$$

$$a_{ik}(\sum_{j \in J} x_{ij} + \sum_{l \in J} x_{kl}) \ \leq 1 \ , \forall i, k \in I \tag{9}$$

$$\sum_{j \in J} x_{ij} \ \geq b_{ik} \times \sum_{l \in J} x_{kl} \ , \forall i, k \in I \tag{10}$$

$$p_i \in R^+, \ \forall i \in I \tag{11}$$

Equation (6) states the objective namely, maximizing the total fitness score of the selected CAVs. Equation (7) ensures that the current capacity of each downstream lane is not exceeded by the total length of the CAVs to receive. Equation (8) requires that each CAV is destined to only one downstream lane. Equation (9) ensures that the movements of the scheduled CAVs are not conflicting. Equation (10) ensures that each vehicle is scheduled after its predecessors on the same lane. Equation (11) states that the fitness score of each scheduled CAV is positive.

$$p_i = \frac{\theta_i}{\widehat{\theta}} \times \pi_i \tag{12}$$

p_i is estimated using Equation (12) where θ_i, π_i, and $\widehat{\theta}$ respectively denote CAV i 's waiting time, its priority index and the maximum authorized waiting time (defined as a parameter).

Let $\Upsilon_{ij}=\begin{cases}1\,, \text{if movements } i \text{ and } j \text{ are conflicting}\\ 0\,, \text{otherwise}\end{cases}$

Note that $\forall i,k \in I, a_{ik} = \Upsilon_{pq}$; $p,q \in \mathcal{M}$ where $p=(o_i,\delta_i)$ and $q=(o_k,\delta_k)$

$$t^*_{veh} = \max_{i \in I'}\{t_i\} \tag{13}$$

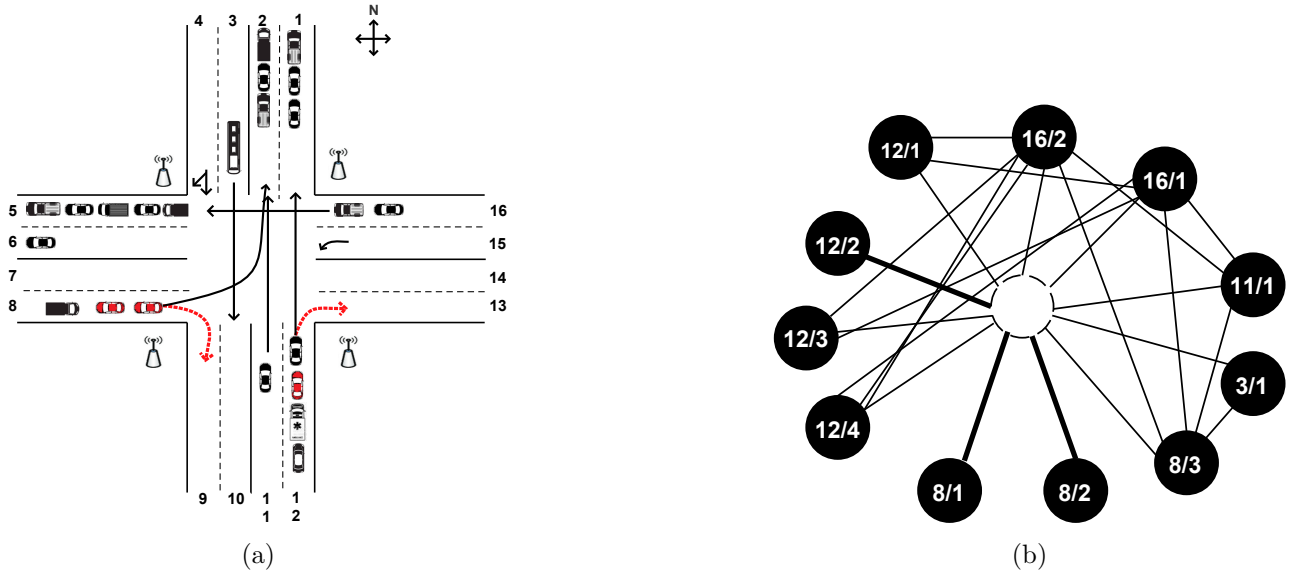
$$t_i = \frac{d_{ij}}{v_i} \tag{14}$$

The duration of the optimal crossing session referred to as t^*_{veh} is calculated using Equation (13) where t_i denotes CAV i 's travel time delay and $I' \subseteq I$ the set of the selected CAVs.

t_i is calculated using Equation (14) where v_i and d_{ij} respectively denote the departure speed of CAV i and the distance to its destination j.

Unfortunately, MKP has been shown to be strongly NP-hard [26]. Moreover the 0-1 MKP has been proved NP-Complete [27]. Consequently, to determine I' we propose the following heuristic :

- Step 1: find all the MISs(Maximal Independent Sets) from the constructed conflict graph ;
- Step 2: choose a new MIS and group CAVs according to their lanes otherwise go to Step 4 ;
- Step 3: choose a new group and sort CAVs according to their positions otherwise go to Step 2 ;
- Step 4: choose only the CAVs that meet the constraint expressed by Equation (7) and calculate their fitness scores otherwise go to Step 3 ;
- Step 5: choose the MIS that has yielded the best total fitness score ;
- Step 6: calculate the duration of the crossing session from the chosen MIS see Eqs. (13) and (14).



(a) (b)

Figure 4: Illustration of the conflict graph construction process: a) Possible conflictual movements of the CAVs in the decision zones. b) The resulting conflict graph where vertices refer to CAVs designated according to lane/position (e.g. 12/1 denotes the first CAV on lane 12) , the dotted vertex is a virtual one.

Indeed, to satisfy the constraint expressed by Equation(9) and avoid collisions a method commonly used in the literature suggests the extraction of MISs from a conflict graph. Figure 4a depicts an intersection where solid black and dotted red arrows refer to movements of black and red CAVs respectively. The resulting conflict graph is presented in Figure 4b where vertices are named according to the lane/position syntax. We use a virtual node (the dotted vertex) to connect all the components. Without this scheme 12/2, 8/1, and 8/2 would have been isolated from other vertices.

After calculating t^*_{veh} using Equation (13) a leader i must wait until $t_{sched}(i)$ before broadcasting a SCHED message. If this CAV is concerned by this schedule it must apply the crossing decisions.

$$t_{sched}(i) = \Lambda - \max(t^*_{veh}; t^*_{walk}) + \mathcal{U}(0;1) \tag{15}$$

The waiting time $t_{sched}(i)$ is estimated using Equation (15). Λ denotes a fixed time referred to as the competition time; while $\mathcal{U}(.)$ is a duration randomly and uniformly chosen in interval $[0, 1]$.

$$t^*_{walk} = \max_{\psi \in \Psi}(t_{walk}(\psi)) \tag{16}$$

t^*_{walk} is the duration of the crossing session granted to all the VRUs, calculated using Equation (16) where $t_{walk}(\psi)$ is the time to traverse crosswalk ψ. This duration is provided by the RSUs in charge of this crosswalk at the beginning of the current process via CROSS_ACK messages. The calculation process of $t_{walk}(\psi)$ is discussed in the next section (3.4).

On receiving a SCHED message, a leader i must stop calculating t^*_{veh} and apply the decisions contained in this message if it is concerned by the schedule (i.e. $i \in I'$). Then it must send this message to its followers. The SCHED messages will be forwarded by these followers like the DATA_REQ ones. Then leader i must adjust its speed v_i according to its new status in this schedule as defined by Equation (17). Where $t_{prd}(i)$ is the duration of the upcoming crossing session and consequently the deadline for any possible new crossing request via a LEAD_REQ message. d_{ij} is the distance from CAV i to downstream lane j; $\widehat{d}_i$ denotes the distance from CAV i to lane exit and V_{max} is the speed limit for a CAV. $\widetilde{d}_{ij}$ is the distance from CAV i to its front neighbour j (the one who forwarded the SCHED message). Ω is the inter-vehicle safety distance.

$$v_i = \begin{cases} \min\left(\frac{d_{ij}}{t_{prd}(i)}; V_{max}\right), \text{if } (i \in I') \wedge (s_i = 1) \\ \min\left(\frac{\widehat{d}_i}{t_{prd}(i)}; V_{max}\right), \text{if } (i \notin I') \wedge (s_i = 1) \\ \frac{\widetilde{d}_{ij} - \Omega}{t_{prd}(i)}, \text{otherwise} \end{cases} \tag{17}$$

3.4 Vulnerable user demand handling

As mentioned above, we assume that the system continuously receives information about the waiting VRUs (Vulnerable Road Users) such as pedestrians, cyclists, motorcyclists or person of reduced mobility. Each crosswalk is managed by two RSUs. This concurrent control implies mutual exclusion. In other words, for the same crosswalk, only one demand can be handled by only one RSU at a time. Let Ψ be the set of crosswalks that have received a crossing request from a VRU. Each crosswalk $\psi \in \Psi$ implies a set of upstream and downstream lanes. Formally, $\psi \subset (O \cup J)$. Therefore, when a RSU receives a new crossing request from a VRU, it checks if the average waiting time of the VRUs is greater than the average waiting times of the leaders on the upstream lanes of the concerned crosswalk and there is no emergency CAV in the decision zones. If so, it virtually closes each downstream and upstream concerned by the destined crosswalk. In other words, for the next crossing session the system parameters will be formally updated likewise $\forall j \in (J \cap \psi), C_j = \lambda_j$ and $O \setminus (\psi \cap O)$. Then all the resulting changes will be broadcasted to the other RSUs via a PEDS_REQ message. $t_{walk}(\psi)$ the duration of this VRU crossing session is estimated using Equation (18) inspired from Gorodokin *et al.* [28].

$$t_{walk}(\psi) = \begin{cases} \widehat{\tau} + \frac{W_\psi}{sp_\psi} + \phi\frac{\widehat{\delta}(n_\psi - 1)}{sp_\psi^2} + (n_\psi - 1)\,, \text{if } n_\psi > 0 \\ 0\,, \text{otherwise} \end{cases} \tag{18}$$

$\widehat{\tau}$ is the starting delay of the first row of pedestrians at the beginning of the process; W_ψ is the length of crosswalk ψ; sp_ψ is the average speed of the VRUs; ϕ is the distance between rows of these users; n_ψ is the number of these rows. $\widehat{\delta}$ is the distance to the first row from the edge of the crosswalk. Without loss of generality, $\widehat{\tau}$, $\widehat{\delta}$, ϕ are fixed parameters.

Note that $t_{walk}(\psi)$ is sent via the CROSS_ACK messages to the leaders by the RSUs in charge of crosswalk ψ.

It is also noteworthy that the average waiting time of the VRUs can be trivially estimated without loss of generality; by subtracting the time of the last crossing request to the current time.

Table 1 describes the parameters used by UICP. The main process of UICP is detailed by Algorithm 1. Algorithms 2 and 3 present its timer management process and the rights-of-way assignment scheme.

4 Performance evaluation

To evaluate the performance of UICP we conducted a simulation campaign with an isolated intersection through four metrics namely Average delay, Average pedestrian delay, Fairmess, and Degree of saturation. Infrastructure and traffic demands were created with the simulator SUMO 1.18.0 [29] coupled with OMNeT++ 6.0.1 [30]. This second simulator was used to implement all the evaluated protocols. Results are compared with those obtained with two related state-of-the-art protocols namely ROADRUNNER+ by Wang *et al.* [31] and the solution proposed by Niels *et al.* [24]. The latter will be referred to as OPT-AIM. Tables 2 and 3 show all the simulation parameters. Average arrival rates per upstream edge were scaled with parameter α. 450 veh/h and 900 veh/h average arrival rates were respectively used for lanes on North/South and West/East directions. α were increased from 1 to 2 with a 0.2 step. The maximum waiting time for CAVs $\widehat{\theta}$ was set to 300 s (the default value in SUMO). Note that as shown in Table 3 VRUs' average speed was randomly and uniformly chosen in interval $[1.39, 5.56]$. The dimensions of the five zones as defined by Wang *et al.* [31] were applied for ROADRUNNER+. As for UICP, we chose $\lambda a = 75$ and $\lambda d = 125$ for the two zones respectively. We applied the same values for the two zones of OPT-AIM based on the ratio defined in by Niels *et al.* [24]. To simulate the occurrence of spillbacks we randomly and uniformly chose the residual capacity of each downstream lane in interval $[0, 200]$ at the beginning of each new crossing session.

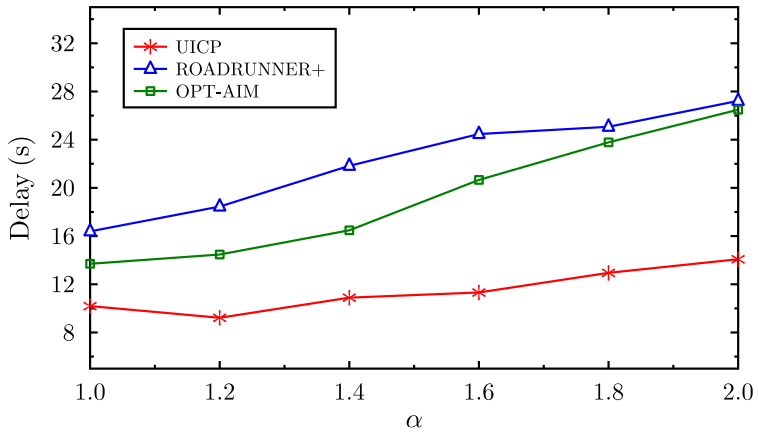


Figure 5: Average delay of the CAVs with traffic intensity = $\alpha\times$ the arrival rate, namely, 450 veh/h and 900 veh/h respectively for lanes on North/South and West/East directions.

Table 1: Summary of Notations.

Symbol	Definition
$\mathcal{A}$	matrix of conflicts between CAVs
a_{ik}	relationship between CAVs i and k
$\mathcal{B}$	matrix of precedences between CAVs in the decision zones
b_{ik}	precedence between CAVs i and k
C_j	residual capacity of downstream lane j
d_{ij}	distance from CAV i to downstream lane j
$\widetilde{d_{ij}}$	distance from CAV i to CAV j
$\widehat{d_i}$	distance from CAV i to lane exit
$\widehat{\delta}$	distance first row to crosswalk's edge
δ_i	destination of CAV i
E_i	initial energy of CAV i
I	set of CAVs in the decision zones
I'	set of CAVs that received rights-of-way
J	set of downstream lanes
λ_i	length of lane i
λa	length of the approach zone
λd	length of the decision zone
ℓ_i	length of CAV i
$\mathcal{M}$	set of possible movements
m	cardinality of I
n	cardinality of J
n_ψ	number of rows of VRUs on crosswalk ψ
O	set of upstream lanes
o_i	origin of CAV i
Ω	inter-vehicle safety distance
ϕ	distance between rows of vulnerable users
π_i	priority index of CAV i
Ψ	set of the requested crosswalks
s_i	status of CAV i (0=follower, 1=leader)
sp_ψ	average speed of the VRUs on crosswalk ψ
τ	current time
$\widehat{\tau}$	starting delay of the first row of users
θ_i	CAV i's waiting time
$\widehat{\theta}$	maximum allowed CAVs'waiting time
$t_{cross}(i)$	time of the next crossing decision of CAV i
$t_{data}(i)$	end time of the data collection by CAV i
$t_{lead}(i)$	time of the status update of CAV i
$t_{prd}(i)$	time of the next status request by CAV i
$t_{sched}(i)$	time before broadcasting a new schedule by CAV i
$t_{walk}(\psi)$	time to use crosswalk ψ
Υ_{ij}	relationship between movements i and j
v_i	speed of CAV i
V_{max}	speed limit for the CAVs
W_ψ	width of crosswalk ψ
w_i	length of CAV i
x_{ij}	CAV i allowance to destination j
ξ_i	residual energy of CAV i

Algorithm 1 CAV i's UICP main process

Input: $I, J, ...$ ▷ see Table1
Output: o_i, δ_i, v_i

calculate $t_{prd}(i)$ ▷ see Equation (2)
while $(\xi_i > E_i)$ **do** ▷ is residual energy enough ?
  $N(i) \leftarrow$ `discover_neighbors()` ▷ update the list of neighbors
  `check_timers()` ▷ see Algorithm(2)
  $message \leftarrow$ `get_messages()` ▷ look for any incoming message
  if $(message \neq$ " " $)$ **then**
    $j \leftarrow$ extract from $message$ ▷ get id of the message's sender
    switch $message$ **do**
      case LEAD_REQ
        $o_j \leftarrow$ extract from $message$
        if $(o_i = o_j)$ **then**
          send LEAD_ACK() to j
        end if
      case LEAD_ACK
        if $(t_{lead}(i) > 0)$ **then**
          $s_i \leftarrow 0$
          $t_{lead}(i) \leftarrow 0$
        end if
      case CROSS_REQ
        if $(s_i = 1)$ **then**
          send DATA_REQ() to $N(i) \setminus j$
          calculate $t_{data}(i)$ ▷ see Equation (4)
        end if
      case CROSS_ACK
        save $\{\ell_j, \pi_j, o_j, \delta_j\} \cup t_{walk}(\psi)$ from $message$
      case DATA_REQ
        $o_j \leftarrow$ extract from $message$
        if $(o_i = o_j)$ **then**
          if $(|N(i)| > 1)$ **then**
            send DATA_REQ() to $N(i) \setminus j$
          else
            send DATA_ACK() to j
          end if
        end if
      case DATA_ACK
        $o_j \leftarrow$ extract from $message$
        if $(o_i = o_j)$ **then**
          $4uple \leftarrow \{\ell_i, \pi_i, o_i, \delta_i\}$
          if $(s_i = 1)$ **then**
            broadcast $message \cup$ CROSS_ACK($4uple$)
            calculate $t_{cross}(i)$ ▷ see Equation (5)
          else
            send $4uple \cup$ DATA_ACK() to j
          end if
        end if
      case SCHED
        $t_{sched}(i) \leftarrow 0$
        $t_{prd}(i) \leftarrow$ extract from $message$
        $I' \leftarrow$ extract from $message$
        if $(i \in I')$ **then**
          send SCHED($t_{prd}(i)$, I') to $N(i) \setminus j$
        end if
        calculate v_i ▷ see Equation(17)
    end switch
    $message \leftarrow$ " "
  end if
end while

Algorithm 2 Timers management process of CAV i

Input: $d_i, \lambda a, s_i$
Output: $t_{lead}(i), t_{data}(i), t_{prd}(i), t_{cross}(i), t_{sched}(i), v_i$

$\tau \leftarrow$ `get_current_time()`
switch τ **do** ▷ comparing the current time to the active timers
  case $t_{prd}(i)$
    if $(d_i < \lambda a)$ **then**
      send LEAD_REQ()
      calculate $t_{lead}(i)$ ▷ see Equation (3)
    end if
    calculate $t_{prd}(i)$ ▷ see Equation (2)
  case $t_{lead}(i)$
    $s_i \leftarrow 1$
    $t_{lead}(i) \leftarrow 0$
    broadcast CROSS_REQ()
  case $t_{data}(i)$
    $t_{data}(i) \leftarrow 0$
  case $t_{cross}(i)$
    $t_{cross}(i) \leftarrow 0$
    $I' \leftarrow$ `get_schedule()` ▷ see Algorithm(3)
    calculate t^*_{veh} ▷ see Equations. (13) and (14)
    calculate $t_{sched}(i)$ ▷ see Equation (15)
  case $t_{sched}(i)$
    $t_{prd}(i) \leftarrow \max(t^*_{veh}; t^*_{walk})$
    broadcast SCHED($t_{prd}(i)$, I')
    $t_{sched}(i) \leftarrow 0$
    calculate v_i ▷ see Equation(17)
end switch

Algorithm 3 Rights-of-way assignment process of CAV i

Input: $I, J, ...$ ▷ see Table 1
Output: I'

$\mathcal{L}_{\mathcal{MIS}} \leftarrow$ get all the MISs of $\mathcal{A}$
$r^* \leftarrow -\infty$
for each set $S \in \mathcal{L}_{\mathcal{MIS}}$ **do**
  $r \leftarrow 0$
  $\Psi \leftarrow \emptyset$
  $\mathcal{G} \leftarrow$ group CAVs in S according to lane
  $\mathcal{G} \leftarrow$ sort CAVs in each group g of $\mathcal{G}$ according to their positions
  for each $g \in \mathcal{G}$ **do**
    for each CAV $i \in g$ **do**
      $j \leftarrow \delta_i$
      if $(\ell_i + C_j + \Omega) \leq \lambda_j$ **then**
        $C_j \leftarrow C_j + \ell_i + \Omega$
        calculate p_i ▷ see Equation (12)
        $r \leftarrow r + p_i$
        $\Psi \leftarrow \Psi \cup \{i\}$
      else
        $g \leftarrow \emptyset$
      end if
    end for
  end for
  if $r > r^*$ **then**
    $r^* \leftarrow r$
    $I' \leftarrow \Psi$
  end if
end for

Table 2: Parameters of the intersection

Parameter	Value
Lane length	200 m
Number of lanes	3 per edge
Traffic	(N/S) 450 (W/E) 900 veh/h
V_{max}	50 km/h
α (Scale)	1 - 2
Ω	2.5 m
$\widehat{\tau}$	3 s
$\widehat{\delta}$	0.7 m
$\widehat{\theta}$	300 s
ϕ	1 m
Length of the crosswalks	22 m
sp_{ψ}	$\mathcal{U}(1.39\,;5.56)$ m/s
Experiment duration	10000 s
Warm-up	100 s

Table 3: Parameters for the vehicle types

Name	Length	Probability	Priority
Personnal	5 m	20%	1
Bus	12.5 m	2%	1
Truck	16.25 m	1%	1
Motorcycle	2.2 m	1.8%	1
Ambulance	8 m	0.2%	5
Police	8 m	0.5%	2
Fire brigade	8 m	0.1%	4

4.1 Average delay of CAVs

We studied the average delay under different traffic demands to assess the ability of the three protocols to reduce waiting time. To do so, the queued CAVs' waiting times were assessed on each upstream lane and averaged every 1.5 s. Each experiment was repeated 40 times with different seeds. Results are averaged with a 95% confidence interval.

Figure5 shows that UICP yields the lowest delays irrespective of the traffic intensity; a reduction of 38.90% and 48.05% on average compared to OPT-AIM and ROADRUNNER+ respectively. This is due to the fact that unlike UICP, the two other protocols struggle to allow simultaneous non-conflicting movements of CAVs and VRUs. Indeed, UICP handles all the users indistinguishably while creating its conflict graph after just ignoring the CAVs of the risky lanes when necessary. Furthermore, ROADRUNNER+ often leads CAVs to adopt a stop-and-wait behaviour. Additionally, when traffic intensity increase some of its control zones (Lane-Assign Zone, Prepare Zone and Buffer Zone) become useless and by contrast favour delays; whereas UICP and OPT-AIM use only two control zones. However, OPT-AIM is run in fixed intervals of time; such a scheme is detrimental to the intersection throughput and thus increases delay.

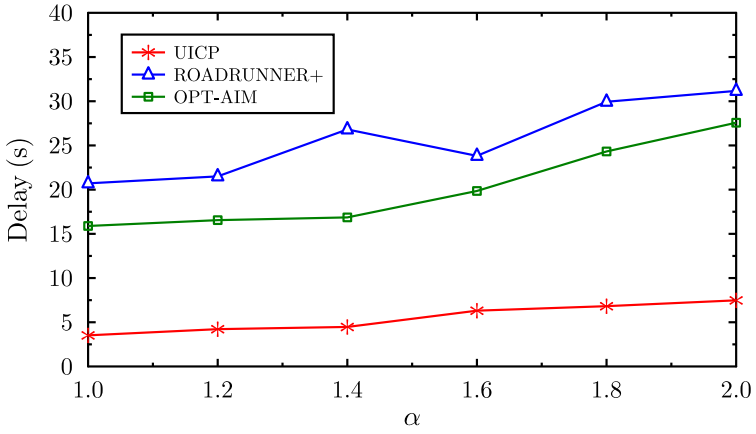


Figure 6: Average delay of the prioritized CAVs with traffic intensity = $\alpha\times$ the arrival rate, namely, 450 veh/h and 900 veh/h respectively for lanes on North/South and West/East directions.

4.2 Average delay of prioritized CAVs

This experiment aimed at measuring the ability of the three protocols to specifically reduce emergency CAVs' waiting times. Queued emergency CAVs' sojourn times were assessed on each upstream lane and averaged every 1.5 s. This experiment was repeated 40 times with different seeds. Results are averaged with a 95% confidence interval.

The results depicted in Figure6 show that irrespective of traffic intensity, when it comes to prioritize emergency vehicles, UICP can significantly reduce delays (to 7s on average); a reduction of 38.82% on average. This is due to its integration of priority as a multiplier of waiting tie ratio. By contrast, ROADRUNNER+ and OPT-AIM does not apply vehicle prioritization explicitly. Therefore, they regrettably handle all the CAVs on the same basis; hence the delays that corroborate on average the ones discussed in the previous section.

4.3 Average delay of VRUs

This experiment was conducted to evaluate the ability of each protocol to minimize VRUs' waiting times. The number of these users were assessed on each upstream lane and averaged every 1.5 s. Each experiment was repeated 40 times with different seeds. Results are averaged with a 95% confidence interval.

Figure7 suggests that irrespective of the traffic intensity UICP provides on average the lowest delays regarding the VRUs; a reduction of 12.60% and 18.67% compared to ROADRUNNER+ and OPT-AIM

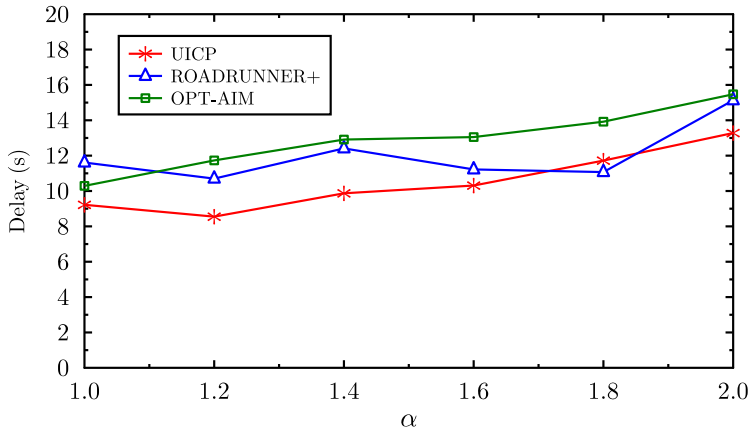


Figure 7: Average delay of the VRUs with traffic intensity = $\alpha\times$ the arrival rate, namely, 450 veh/h and 900 veh/h respectively for lanes on North/South and West/East directions.

respectively. These results are somewhat similar to that of the CAVs in section 4.1. This supports the fact that UICP handles the CAVs and the RSUs indistinctly as mentioned above. By contrast, ROADRUNNER+ and OPT-AIM endeavour to integrate a crossing time for the VRUs (mainly the pedestrians) inside the current CAVs' schedule. Unlike UICP the two other protocol estimate VRUs' crossing time irrespective of their number; hence the high delays that increase on average with the traffic intensity. However, the values of ROADRUNNER+ although high, are more variable because of its spillback handling scheme. Indeed, the latter often delays some CAVs unwillingly for the benefit of the pedestrians.

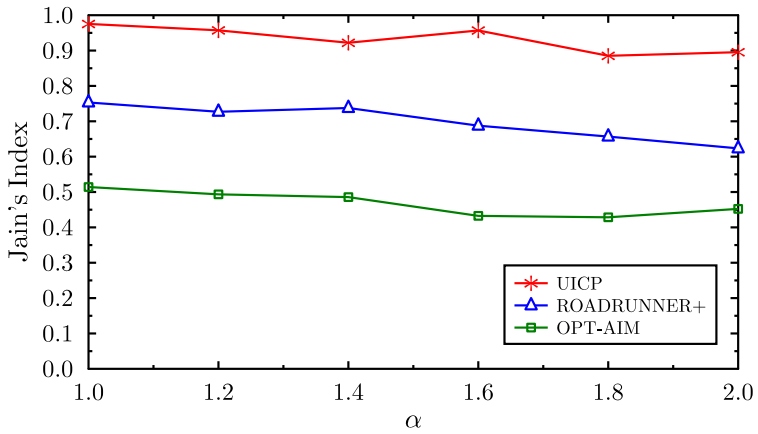


Figure 8: Fairness estimation with traffic intensity = $\alpha\times$ the arrival rate, namely, 450 veh/h and 900 veh/h respectively for lanes on North/South and West/East directions.

4.4 Fairness

We used the Jain's fairness index [32] denoted by f to explore the fairness of each solution regarding delay performance. This index was estimated using Equation (19) where Δ_i is the delay experienced by road user i, and M is the total number of road users. Note that obviously the emergency (i.e. prioritized) CAVs were not considered.

$$f = \frac{\left(\sum_{i=1}^{M} \Delta_i\right)^2}{M \sum_{i=1}^{M} \Delta_i^2} \tag{19}$$

This index was calculated before each new crossing session. Results are averaged with a 95% confidence interval. Each experiment was repeated 40 times with different seeds. Note that if all the considered road users experience the same delay, the fairness index is 1, and the protocol is said to be 100% fair in terms of delay [33].

Figure 8 shows that UICP yields the highest results irrespective of the traffic intensity; namely, an index of 0.93 on the average compared to ROADRUNNER+ and OPT-AIM with indices of 0.70 and 0.47 respectively. In other words, regarding the common CAVs and the VRUs, UICP is more fair in terms of delay than the two other protocols. This is due once again, to the fact that the VRUs' requests are handled by UICP as CAVs' requests with a virtual spillback occurrence. By contrast, when using OPT-AIM and ROADRUNNER+ durations for the crossing of the VRUs are fixed. The latter durations are often extended when a real spillback occurs (i.e. when the capacity of a downstream lane is exceeded). Since OPT-AIM is not spillback-aware this protocol struggle more to handle such situations; hence its poor results.

5 Conclusion

In this paper we addressed the topic of designing an Autonomous Intersection Management(AIM) with the purpose of congestion mitigation. We formulated this question as a Conflict-free and Precedence-Constrained Multiple Knapsack Problem (CPCMKP) a combination of the well-known 0-1 Multiple-Knapsack Problem (MKP) and Precedence-Constrained Knapsack Problem (PCKP). We suggested an integer programming model and a distributed heuristic which lessen the risk of occurrence of spillback. The resulting protocol referred to as UICP, leverages a strategy which considers concurrently vehicle prioritization, vehicle heterogeneity, and most of the Vulnerable Road-Users (VRUs). To find a right-of-way for the CAVs (Connected Autonomous Vehicles) lanes are split into just two zones. The closest to the junction is the decision area. Each intersection was modeled as a set of knapsacks (i.e. the downstream lanes) to be filled with the most valuable items (i.e. the CAVs). Knowing that each item has a profit and consumes an amount of resources (i.e. vehicle's length) of each knapsack whose capacity is known. The goal is to maximize the sum of profits of the items so that the sum of weights in each knapsack does not exceed its capacity. VRUs' requests are handled as CAVs' requests in the presence of a virtual spillback. Simulation results showed that UICP outperforms two major related solutions recently proposed in the literature, in terms of delay and queue length minimization. This protocol also avoids collisions, maximizes throughput while ensuring fairness. All these features prove that UICP is an efficient scheme for real-world congestion mitigation.

As a future work, we plan to extend this solution with features like lane change or platooning in environments implying human-driven vehicles and CAVs interaction.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Daniela Müller-Eie and Ioannis Kosmidis. Sustainable mobility in smart cities: a document study of mobility initiatives of mid-sized nordic smart cities. *European Transport Research Review*, 15(1), October 2023.

[2] Prashant Singh, Maxim A. Dulebenets, Junayed Pasha, Ernesto D. R. Santibanez Gonzalez, Yui-Yip Lau, and Raphael Kampmann. Deployment of autonomous trains in rail transportation: Current trends and existing challenges. *IEEE Access*, 9:91427–91461, 2021.

[3] Attila Simo, Simona Dzitac, Liviu Ferestyan, Cristian-Dragos Dumitru, and Adrian Gligor. Optimizing electric vehicle performance: Advances in battery management systems for enhanced efficiency and longevity. *International Journal of Computers Communications & Control;*, 19(5), September 2024.

[4] Liping Zhao, Feng Li, Dongye Sun, and Zihan Zhao. An improved ant colony algorithm based on q-learning for route planning of autonomous vehicle. *International Journal of Computers Communications & Control;*, 19(3), May 2024.

[5] Faran Awais Butt, Jawwad Nasar Chattha, Jameel Ahmad, Muhammad Umer Zia, Muhammad Rizwan, and Ijaz Haider Naqvi. On the integration of enabling wireless technologies and sensor fusion for next-generation connected and autonomous vehicles. *IEEE Access*, 10:14643–14668, 2022.

[6] K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, March 2008.

[7] Biyao Wang, Yi Han, Siyu Wang, Di Tian, Mengjiao Cai, Ming Liu, and Lujia Wang. A review of intelligent connected vehicle cooperative driving development. *Mathematics*, 10(19):3635, oct 2022.

[8] Abdeljalil Abbas-Turki, Yazan Mualla, Nicolas Gaud, Davide Calvaresi, Wendan Du, Alexandre Lombard, Mahjoub Dridi, and Abder Koukam. Autonomous intersection management: Optimal trajectories and efficient scheduling. *Sensors*, 23(3):1509, jan 2023.

[9] Balázs Varga and Tamás Tettamanti. Jam propagation analysis with mesoscopic traffic simulation. *IEEE Transactions on Intelligent Transportation Systems*, 24(12):14162–14173, December 2023.

[10] Ashkan Gholamhosseinian and Jochen Seitz. A comprehensive survey on cooperative intersection management for heterogeneous connected vehicles. *IEEE Access*, 10:7937–7972, 2022.

[11] Soukaina Laabadi, Mohamed Naimi, Hassan El Amri, and Boujemâa Achchab. The 0/1 multidimensional knapsack problem and its variants: A survey of practical models and heuristic approaches. *American Journal of Operations Research*, 08(05):395–439, 2018.

[12] Oscar H. Ibarra and Chul E. Kim. Approximation algorithms for certain scheduling problems. *Mathematics of Operations Research*, 3(3):197–204, 1978.

[13] Fayez Alanazi. A systematic literature review of autonomous and connected vehicles in traffic management. *Applied Sciences*, 13(3):1789, jan 2023.

[14] Zijia Zhong, Mark Nejad, and Earl E Lee. Autonomous and semiautonomous intersection management: A survey. *IEEE Intelligent Transportation Systems Magazine*, 13(2):53–70, 2021.

[15] Michael I.-C. Wang, Jiacheng Wang, Charles H.-P. Wen, and H. Jonathan Chao. Roadrunner: Autonomous intersection management with dynamic lane assignment. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, sep 2020.

[16] Bai Li, Youmin Zhang, Ning Jia, and Xiaoyan Peng. Autonomous intersection management over continuous space: A microscopic and precise solution via computational optimal control. *IFAC-PapersOnLine*, 53(2):17071–17076, 2020.

[17] A. S. M. Bakibillah, Md Abdus Samad Kamal, Chee Pin Tan, Susilawati Susilawati, Tomohisa Hayakawa, and Jun ichi Imura. Bi-level coordinated merging of connected and automated vehicles at roundabouts. *Sensors*, 21(19):6533, sep 2021.

[18] Behdad Chalaki and Andreas A. Malikopoulos. Optimal control of connected and automated vehicles at multiple adjacent intersections. *IEEE Transactions on Control Systems Technology*, 30(3):972–984, may 2022.

[19] Behdad Chalaki and Andreas A. Malikopoulos. Time-optimal coordination for connected and automated vehicles at adjacent intersections. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13330–13345, aug 2022.

[20] Huaxin Pei, Yuxiao Zhang, Yi Zhang, and Shuo Feng. Optimal cooperative driving at signal-free intersections with polynomial-time complexity. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12908–12920, aug 2022.

[21] Hui Zhang, Rongqing Zhang, Chen Chen, Dongliang Duan, Xiang Cheng, and Liuqing Yang. A priority-based autonomous intersection management (AIM) scheme for connected automated vehicles (CAVs). *Vehicles*, 3(3):533–544, aug 2021.

[22] Chaoyi Chen, Qing Xu, Mengchi Cai, Jiawei Wang, Jianqiang Wang, and Keqiang Li. Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):21897–21914, nov 2022.

[23] Zhiyun Deng, Kaidi Yang, Weiming Shen, and Yanjun Shi. Cooperative platoon formation of connected and autonomous vehicles: Toward efficient merging coordination at unsignalized intersections. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5625–5639, may 2023.

[24] Tanja Niels, Klaus Bogenberger, Markos Papageorgiou, and Ioannis Papamichail. Optimization-based intersection control for connected automated vehicles and pedestrians. *Transportation Research Record: Journal of the Transportation Research Board*, page 036119812311729, jun 2023.

[25] Ming S. Hung and John C. Fisk. An algorithm for 0-1 multiple-knapsack problems. *Naval Research Logistics Quarterly*, 25(3):571–579, September 1978.

[26] Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[27] Li'ang Zhang and Suyun Geng. The complexity of the 0/1 multi-knapsack problem. *Journal of Computer Science and Technology*, 1(1):46–50, mar 1986.

[28] Vladimir Gorodokin, Irina Alferova, and Elena Shepeleva. Calculating the duration of the traffic light green interval allowing pedestrians entering the traffic way. *Transportation Research Procedia*, 36:220–224, 2018.

[29] Pablo Alvarez Lopez, Evamarie Wiessner, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flotterod, Robert Hilbrich, Leonhard Lucken, Johannes Rummel, and Peter Wagner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, nov 2018.

[30] [omnet++], a public-source, component-based, modular and open-architecture discrete event simulation environment. official homepage: http://www.omnetpp.org/ accessed on May 2024.

[31] Michael I.-C. Wang, Charles H.-P. Wen, and H. Jonathan Chao. Roadrunner+: An autonomous intersection management cooperating with connected autonomous vehicles and pedestrians with spillback considered. *ACM Transactions on Cyber-Physical Systems*, 6(1):1–29, nov 2021.

[32] Raj Jain, Dah-Ming Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR*, cs.NI/9809099, 1998.

[33] Jian Wu, Dipak Ghosal, Michael Zhang, and Chen-Nee Chuah. Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Transactions on Vehicular Technology*, 67(2):896–909, feb 2018.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Diédié, G. H. F.; Zounémé, B. S.; N'Takpé, T. (2025). Decentralized Right-of-Way Protocol for Connected and Automated Vehicles at Unsignalized Intersections, *International Journal of Computers Communications & Control*, 20(6), 6677, 2025.
https://doi.org/10.15837/ijccc.2025.6.6677