POLITECNICO DI TORINO
Repository ISTITUZIONALE

Platoon-Local Dynamic Map: Hierarchical cooperative perception for platooning vehicles

Original
Platoon-Local Dynamic Map: Hierarchical cooperative perception for platooning vehicles / Carletti, C.M.R., Casetti, C.E., Haerri, J., Risso, F.. - In: IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. - ISSN 1939-9359. - 74:10(2025), pp. 15132-15147. [10.1109/TVT.2025.3569742]

Availability:
This version is available at: 11583/2999541 since: 2025-09-03T09:58:33Z

Publisher:
IEEE

Published
DOI:10.1109/TVT.2025.3569742

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

Platoon-Local Dynamic Map: Hierarchical cooperative perception for platooning vehicles

Carlos Mateo Risma Carletti*†, Claudio Casetti*, Jérôme Härri†, Fulvio Risso*
*Politecnico di Torino - Department of Control and Computer Engineering, Turin, Italy
†EURECOM - Communication Systems Department, Sophia-Antipolis, France
E-mail: {carlos.rismacarletti, claudio.casetti, fulvio.risso}@polito.it {risma, jerome.haerri}@eurecom.fr

***Abstract*—Onboard perception systems found on modern vehicles generate data that are incredibly rich in contextual information, and thanks to the increasing number of vehicles equipped with communication capabilities, the valuable data generated can be shared with nearby vehicles. One of the most studied vehicular applications for autonomous driving is platooning, for which all maneuvers are managed by the Platoon Leader (PL) with the aid of context information available through Vehicle-to-Vehicle (V2V) messages. However, redundant context information from nearby vehicles in the platoon creates an additional workload for the platoon leader resulting in elevated computational costs. To overcome the challenge of redundant context information, the formation of vehicular micro-clouds can offload the platoon leader by enabling vehicles in a cluster to more effectively utilize the available context data through collective data processing and aggregation. The proposed solution, called Platoon Local Dynamic Map (P-LDM), creates a single database of context information, distributing the data aggregation load among all members of the platoon. Simulation results evaluate the collective perception enhancement and the distribution of the computational load, comparing the proposed scheme with usual Cooperative Perception mechanisms.**

***Index Terms*—V2X, Autonomous Vehicles, Platooning, Vehicular Micro Cloud, Local Dynamic Map, LDM, ITS, Vehicular Networks, 5G**

I. Introduction

TOWARDS the wide-spread deployment of automated vehicles in the near future, the capacity for efficient data exchange among these vehicles emerges as a crucial enabler for meeting the requirements of safe and prompt decision-making. This exchange of data between vehicles and between vehicles and the infrastructure is performed through what is called Vehicle-to-Everything (V2X) communications for which several standards have been developed over the years including the Wi-Fi-based IEEE 802.11p and upcoming 802.11bd, as well as the cellular-based counterparts, namely 3GPP's LTE-V2X and NR-V2X.

The European Telecommunication Standards Institute (ETSI) envisages a system where vehicles communicate by broadcasting standardized messages including specific types of data. In particular, Cooperative Awareness Messages (CAMs) have been defined for vehicles to share their own geographical position and dynamic state (e.g. speed, acceleration, drive direction, etc). Recently, ETSI finalized the standardization of the Collective Perception Service (CPS), which is designed to manage the creation of Collective Perception Messages (CPMs). CPMs have been developed to counteract the inherent limitations of on-board sensors, including their necessity of line of sight, susceptibility to adverse weather conditions, and limited perception range. Indeed, CPMs contain information about the objects perceived stored in the sender vehicle's Local Dynamic Map (LDM). Within ETSI's vision, the LDM constitutes a facility that manages a database with both locally gathered and remote information (from messages such as CAMs and CPMs).

Despite the benefits of the extended awareness achieved with CPMs, the fully distributed dissemination scheme tends to result in a lot of redundant information being exchanged, translating into an unnecessary channel load if the inclusion of information is not controlled [1]. Additionally, the computational capacity needed to match all locally sensed objects with the data from other vehicles increases the delay in the perception pipeline, resulting in outdated perceptions [2].

One of the ITS applications most studied by the research community is platooning, which is a cooperative Automated Driving System (ADS) application for which huge efforts have been made to enable its deployment on real highways [3]. In particular, for the European market, the ENSEMBLE project has brought together the main truck manufacturers of the region to realize a communication protocol for multi-brand truck platooning [4]. ENSEMBLE has laid the foundation for standardization bodies, such as ETSI, to create a new facilities layer service for the C-ITS set of standards. Platoons are, in essence, clusters of vehicles that organize themselves to travel close behind one another with the aim of saving fuel, reducing traffic, and increasing road safety. The availability of context information available from V2V messages crucial for the platoon's safely and timely reaction to potential disturbances.

As illustrated in Figure 1, a problem arises for vehicles in a platoon due to their close proximity, resulting in the gathering and sharing of highly redundant data that needs to be processed by each Platoon Member (PM) to align the locally gathered data with the shared data. Although this redundancy allows for an enhanced view of the road, the incurred processing load to match the information together with the necessary communication overhead results in diminishing returns within the platoon. To address this issue, the concept of vehicular micro-cloud can be employed, which allows the distribution of the computing resources of the PMs. Under this micro-cloud approach, a master node (the Platoon Leader in our case) takes on the role of coordinating and distributing the contextual information responsibility among all PMs by assigning a

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

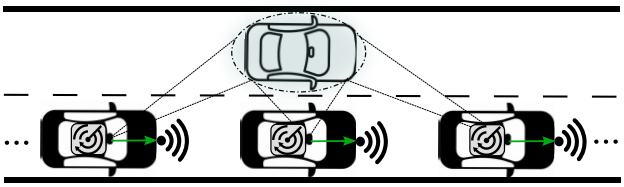


Figure 1. Scheme showing multiple platooning vehicles detecting and broadcasting redundant information about the same non-connected vehicle.

subset of all Platoon Perceived Objects to each.

The service investigated in this paper, the Platoon Local Dynamic Map (P-LDM), aims to reduce redundant perceptions being shared and balance the processing load within the platoon's micro-cloud architecture. The P-LDM creates a single aggregated dataset of context information surrounding a platoon, where the Platoon Leader assigns a subset of all Platoon Perceived Objects to each Platoon Member. Each PM is responsible for updating the portion of data assigned to it by exchanging custom V2V messages, specifically defined for this service. By distributing the processing responsibilities across multiple PMs, the system efficiently balances the computational load within the platoon's micro-cloud. This approach not only minimizes redundant perception sharing but also reduces the overall computational and communication overhead.

The first solution of the P-LDM service has been presented in our conference paper [5]. However, the work presented here further extends the work presented in [5] providing:

- An improved messaging protocol reducing the communication overhead of the service, maintaining the same capabilities.
- The definition of a multi-objective optimization problem for the assignment of platoon perceived objects.
- An evaluation of various heuristic algorithms for the online assignment procedure, providing a performance comparison as well as a scalability analysis.
- A validation of the service in a new open-source simulation framework combining realistic models for communication, sensing and control.

The remainder of the paper is organized as follows: in Section II, a description of related works studying the development of cooperative perception schemes is followed by a detailed description of the system model for the presented service in Section III. In section IV, an optimization problem is formulated for the assignment of Platoon Perceived Objects to each platoon member. Section V showcases the proposed algorithms for our optimization problem with their performance evaluation and scalability analysis. The service is validated in Section VI, for which an implementation has been developed on an open-source simulation framework, outlining the obtained results. Finally, Section VII concludes the paper.

II. Related Works

The concept of LDM has been developed as a powerful facility to represent the dynamic world around a vehicle and has been standardized by ETSI [6], providing guidelines for the interface with other facilities layer services. Indeed, in addition to storing data provided by sensors, ETSI envisions the LDM to be populated with data coming from the Cooperative Awareness Service [7] and the Collective Perception Service [8]. Although ETSI defines a set of redundancy mitigation rules in [8], the authors in [9] argue that these rules fail to effectively reduce the channel overhead created due to redundant data being exchanged.

The process of matching, involving associating new sensor data with previously detected objects, and tracking, i.e., updating these associations over time, are critical components for achieving Collective Perception. Search algorithms for matching can operate in milliseconds for a few objects, but scalability becomes a problem with increasing numbers of data sources and objects to track [10]. In the presence of redundant information within exchanged CPMs, vehicles need to match the objects from received CPMs with the ones already stored by onboard sensors and previous CPMs. The repeated reception of data about the same objects (from different sources) overloads the matching process, introducing high end-to-end delay and rendering the additional information useless [2].

Several works have explored different ways to address information redundancy in collective perception. In [11], the authors investigate and evaluate several filtering rules to mitigate redundancy, proposing new metrics for quantifying redundancy and demonstrating how excessive duplicates can negatively affect overall system performance. Similarly, the authors from [12] analyzed adaptive object filtering strategies to reduce redundancy while retaining sufficient awareness of relevant road users, showing that controlling redundant data updates can improve both resource efficiency and perception reliability in V2X scenarios. While in both [11] and [12] prove that filtering-based approaches can effectively reduce redundant data transmission, our proposed P-LDM instead reduces redundancy without losing context awareness that might be lost by filtering leveraging a hierarchical assignment which also distributes the computational load across multiple platoon members. In [13], the authors propose a Value of Information (VoI)-based method to reduce redundant transmissions among Connected and Automated Vehicles (CAVs) by quantifying the utility of each CPM entry and discarding less useful ones. Compared to the approach on [13], which relies on a centralized scheduler, our service avoids the need for a persistent centralized entity that may be unreachable in remote or highway scenarios. Instead, it distributes the perceptual processing tasks among platoon members in a coordinated yet decentralized way, making the P-LDM more resilient to infrastructure coverage.

In the context of decentralized task allocation, the authors in [14] propose a lightweight algorithm for minimizing task completion time within vehicular micro-clouds with heterogeneous computational resources, showing promising results compared to random allocation. In [15], instead, a multi-objective task allocation approach is presented, aiming to optimize energy consumption, latency, and fairness. The multi-objective optimization problem is solved with a custom genetic algorithm proving to be more effective than well-known methods such as the Non-dominated Sorting Genetic Algorithm II (NSGA-

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

II). Similarly, the authors in [16] present a novel approach to optimizing resource allocation in vehicular cloud computing environments. The authors propose a multi-objective optimization model that seeks to minimize both the blocking probability of tasks and the operational cost for providers solved through an NSGA-II-based algorithm. A task scheduler that enhances fairness in resource usage by balancing the load across vehicular clouds through a probabilistic selection method is proposed in [17]. Even though a considerable body of research has focused on decentralized task allocation in mobile micro-clouds, all these works consider general task without considering specific use cases, such as cooperative perception, which pose their particular constraints.

A resource allocation strategy is proposed in [18] to distribute bandwidth and compute resources based on collaboration quality, considering factors like concurrent tasks and data overlap. Although effective for highly dynamic scenarios, the authors do not delve into the cooperative perception benefits of the proposed approach. In contrast, our work explicitly focuses on cooperative perception, providing a structured mechanism to assign and manage perception tasks for detected objects.

In [19], the authors present a clustering algorithm for assigning perceived objects to each Connected Autonomous Vehicle (CAV). The proposed algorithm, executed by all CAVs, computes an 'importance' score for each locally detected object which then is compared with the score received from other CPMs about the same objects. The CAV with the highest score for a particular object then proceeds to self-assign that object to its cluster, and subsequently includes that object in the following CPMs. Although the concept of assigning perceived objects to CAVs shows promising results in terms of the reduced redundancy in the messages exchanged, the fully distributed execution of such an algorithm necessitates for all CAVs to match and keep track of all received CPMs perceptions and does not foresee a synchronization of the context representation. The P-LDM, instead, favors a hierarchical scheme to achieve the synchronization of all perceived objects' reference IDs for all platoon members, allowing a more lightweight management of the database.

While some works leverage edge servers to aggregate vehicle perceptions into a unified LDM [20] [21], this centralized approach requires high infrastructure availability, which is often lacking in rural areas where vehicular applications such as platooning mostly operate. Several works have investigated the use of platoons as a cluster of computing resources. Works such as [22] and [23] explored the benefits of leveraging platoons compared to general cluster-based schemes for creating vehicular micro-clouds. Indeed, thanks to the long-term close proximity of vehicles and static topology, platoons provide more stable communication links, reduce message overhead, and improve resource utilization efficiency. On the other hand, in works such as the one presented in [24], a new concept is proposed leveraging idle resources present in Platoon Members to aid edge servers when needed, proposing to offload non-platoon vehicles' tasks to platoon members. In [25], the authors proposed a federated learning mechanism leveraging the steady network conditions within a platoon despite the elevated absolute speed of worker nodes.

Various authors have studied the impacts and challenges of non-connected Human Driven Vehicles (HDVs) on safe platoon control and maneuvers [26]. However, there is still a gap in research on the benefits of cooperative perception for platoons in mixed traffic scenarios. The authors in [27] presented a novel framework that harnesses the capabilities of the CARLA simulator to analyze the improvements provided by cooperative perception for the execution of platoon joining maneuvers. Although cooperative perception is shown to be essential for mixed traffic scenarios, the authors analyze a scenario where raw sensor data is exchanged under ideal communication conditions, which is very difficult to achieve when relying on V2V communications. In this work, instead, we propose a lightweight service working with object-level perception sharing, which offloads and distributes the platoon leader's perception tasks needed for executing and planning platooning maneuvers to all Platoon Members.

III. System Model

A. System overview

In the considered system, all vehicles belonging to a platoon are equipped with communication devices for V2X communications and a C-ITS stack for the dissemination of CAMs and CPMs. All Connected Vehicles (CVs) are equipped with a GNSS device for positioning, a set of cameras (facing front, left, back, and right of the vehicle), and a LiDAR sensor. The cameras and LiDAR sensor are used for object detection leveraging sensor fusion. Longitudinal platoon control is performed using the CACC controller defined in [28, Chapter 7]. The set of messages defined by the ENSEMBLE project [4] is used to exchange the data necessary for the controller.

Based on available context information, the Platoon Leader (PL) is responsible for all platoon-wide maneuvers, including allowing new PMs to join or notifying that a PM has left the platoon. As in a near future scenario, our system considers only a given percentage of vehicles to be CVs, defined by the V2X technology penetration rate, establishing a *mixed* traffic scenario. For the environmental awareness of all vehicles, each CV relies on CAMs to perceive other CVs within communication range and on CPMs to perceive non-connected vehicles outside its perception range. For all the vehicles under perception range, the data collected by each CV's sensors is processed to update the Vehicle's LDM, referred to as V-LDM in our system, which includes a list of objects and CVs along with their path history. The V-LDM can be updated with data from onboard sensors and information received from other CVs through CAM and CPM transmissions, as defined in [6]. Each CV uses the information stored in the V-LDM to generate CPMs, as specified in [8].

To address the challenges present within ETSI's standards, where each vehicle independently manages and shares locally perceived objects, the P-LDM unifies the multiple views of all PMs into a single synchronized database of the entire platoon, as depicted in Figure 2. However, the processing of this unified database is not centralized and is instead distributed among the PMs. Thus, in our approach, the entire set of Platoon Perceived Objects (PPOs), i.e., the set of objects under the perception

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

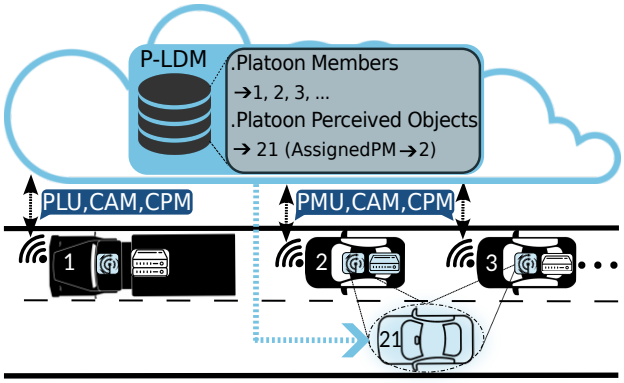


Figure 2. P-LDM architecture and perception aggregation scheme.

range of the platoon as a whole, is aggregated in a coordinated way by all PMs. This coordinated aggregation and distributed processing allows the platoon to efficiently manage the PPOs while reducing redundant computational effort by splitting database management among all PMs. The PL serves as the master node, managing the P-LDM service, overseeing task distribution, and offloading data aggregation to other members.

The P-LDM contains information about all PPOs, i.e., objects currently being perceived by one or more PMs, and information about all CVs in the platoon vicinity. Finally, the P-LDM contains a *PM state map* with information on each PM, including which PPOs are currently under their detection range, which PPOs they are assigned with, and their available free resources. To simplify the management of the P-LDM database, each entry is stored with a Platoon ID representing the platoon-wide identity of each vehicles in the context (PL, PMs, CVs and PPOs). The Platoon ID used for CVs outside the platoon is the ITS-Station (ITS-S) ID, allowing for interoperability with CAMs. Instead, the Platoon ID used for PPOs is the object ID given by the first PM assigned with such PPO.

B. Message Protocols

The data aggregation is achieved by synchronizing the context databases of all PMs leveraging, in addition to CAMs, a set of two newly defined messages, the *Platoon Leader Update* (PLU) and the *Platoon Member Update* (PMU). This message set is used for perception data exchange and management data exchange for the PPO assignment procedure. The PLU is sent periodically by the PL, carrying the following information containers:

- **Platoon Leader Container**: geographical, dynamic information of the PL.
- **Platoon Members**: list of all PMs' ITS Station Identifier (ITS-S ID) and Platoon ID.
- **Connected Vehicles**: list of each CV not belonging to the platoon, containing their ITS-S ID
- **Assigned Perceived Objects**: a list containing geographical and dynamic information about all PPOs that have been self-assigned by the PL.
- **Platoon Perceived Objects Assignments**: a list specifying the PPOs' Platoon ID and the PM assigned with each of the PPOs.

The PMU, instead, is sent by each PM after the reception of a PLU carrying the following information containers:

- **Assigned Perceived Objects**: a list containing geographical and dynamic information about all the PPOs that have been assigned to this PM.
- **New Perceived Objects**: a list with information about all the Perceived Objects 'seen' by this PM (i.e., appearing in its V-LDM) but not yet included in the P-LDM.
- **PM State**: containing the PM's available resources for the P-LDM and the list of PPO IDs of all the PPOs this PM currently detects (assigned to this PM or not).

For the management of the P-LDM service, the PL periodically sends a PLU to all PMs, every 100 ms. At service start-up, the PL issues a PLU with the Platoon ID of all PMs and all CVs outside the platoon, together with the data of PPOs currently self-assigned by the PL. At PLU reception, each PM matches the new PPO data with its locally perceived objects and for all replicas, it updates the local object's ID with the P-LDM platoon ID.

After the first P-LDM update period, i.e., PLU sent by PL and all PMUs received from all PMs, the information about all PPOs stored in the P-LDM is only the one reported on the *Assigned Perceived Objects* of each PMU and PLU. Thus, upon reception of a PLU, all PMs update the assignments on their P-LDM and wait for the rest of the PMUs to complete the P-LDM database update.

C. Object data processing

To match the objects for ID synchronization, each object of the V-LDM and P-LDM is translated into a 2D box defined by 4 corners and a central point. Each object of the V-LDM is compared to each object of the P-LDM computing the Intersection-over-Union (IoU) and the distance between the central points of each object. The IoU metric measures the overlap between two 2D boxes and is defined as the ratio of the area of their intersection to the area of their union. The IoU ranges from 0 to 1, where 0 indicates no overlap and 1 indicates a perfect overlap. In order to account for possible time differences between detections, potential occlusions from obstacles or harsh environments, and the challenges posed by highly dynamic objects, each PM predicts the positions of all objects at the matching time by implementing a Kalman filter with a constant velocity model for robust prediction [29]. For an IoU higher than 0 and a central point distance lower than 2 m, the two objects are considered to be the same, and the local object ID in the V-LDM is synchronized with the Platoon ID. The reason behind the usage of the central point in addition to the IoU as a threshold to match the objects is due to the shape of vehicles, often being long rectangles in their 2D representation, thus resulting in very low or zero IoU values for small lateral misalignments, leading to false negatives during matching.

When locally perceived objects stored in the PM's V-LDM are not matched to any of the PPOs in the P-LDM, the

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

PM appends their information to the *New Perceived Object* container of the PMU, using a newly generated Platoon ID that is not currently in use. Once all new PPOs are matched with its locally Perceived Objects, each PM checks which of the PPOs *in sight* have been assigned to it by the PL, if any, and adds the PPO data to the *Assigned Perceived Objects*. After processing the PLU, each PM sends a PMU with the updated information.

On each PMU reception, the PL updates the P-LDM with the updated perceptions extracted from the *Assigned Perceived Objects* container and checks for new PPOs in the *New Perceived Objects* to be added to the P-LDM. Before storing a new PPO in the P-LDM, the PL checks for duplicates that might be found in other PMUs received during the same update period, storing the perception of the closest PM.

Before starting a new update period, the PL begins the *assignment selection procedure* in which it updates, if necessary, the PPOs assignments. The *assignment selection procedure* seeks to jointly achieve the following platoon-wide objectives:

- **Minimum computational cost**
- **Maximum computational fairness**
- **Maximum perception robustness**

The assignment selection problem is detailed in Section IV for which the proposed algorithms are described in Section V.

D. Quantifying object perception

The P-LDM is designed as a Facilities layer service within ETSI's C-ITS stack, working alongside facilities such as the Cooperative Awareness and Collective Perception services. To maintain compatibility with ETSI's standardized messages and leverage the exchanged information, the P-LDM is updated through these messages.

For the management of information from CVs outside the platoon, the PL stores the incoming information found in their CAMs, using their ITS-S ID as the Platoon ID. On the other hand, for PPOs, each PM updates the PL with the context information of its assigned PPOs, using not only the sensor information of that PM but also the sensor information of other PMs with the exchange of CPMs. By fusing the perceptions of multiple PMs, the overall perception accuracy is enhanced with a higher granularity of perception updates as well as multiple viewpoints from which the objects are sensed. This approach mitigates issues arising from sudden obstacles obstructing the view of a vehicle assigned with an object.

For each assigned PPO, a *path history* is created in the PM's local P-LDM, storing the last 10 perceptions received from either local sensors or CPMs, (corresponding to a 1-second window for a 10Hz update) rate providing the best trade-off between historical context awareness and memory efficiency. To reduce the computational workload and delay from matching and fusing objects in incoming CPMs, PMs process only CPM objects with IDs assigned by the PL. Additionally, since ID synchronization is only guaranteed for CPMs from other PMs, all CPMs from CVs outside the platoon are discarded. This policy aligns with the objective of maintaining a continuous context awareness, relying on information from vehicles consistently within communication range, i.e., Platoon Members.

Each PM continuously aggregates all the perceptions of an assigned PPO by computing a weighted average of the received bounding box $\mathbf{B'}$ compared to the one stored in the P-LDM ($\mathbf{B}$). The weighted average between the 2 bounding boxes is obtained by computing the weighted average of each of the variables defining them: center point (stored in a cartesian format from the ego-vehicle point of reference), width, length, and yaw angle. To achieve a trade-off between prioritizing newer perceptions and prioritizing perceptions perceived from a better viewpoint, the weights are computed by dividing the perception confidence C by the perception age A. The perception confidence of an object in our system is the classification confidence obtained by the computer-vision AI model used, providing a value between 0 and 100. On the other hand, the perception age is the difference between the time at which the object has been detected and the current time, measured in milliseconds, considering values in the range from 1 to 100 ms (i.e., a value of 0 is treated as 1). Thus, each new PPO perception's bounding box $\boldsymbol{\pi}_{PPO}$ is converted into the weighted average of the received perception and the already stored perception according to the following expression:

$$\boldsymbol{\pi}_{PPO} = \frac{\frac{C}{A} \cdot \mathbf{B} + \frac{C'}{A'} \cdot \mathbf{B'}}{\frac{C}{A} + \frac{C'}{A'}} \tag{1}$$

where C and C' represent the confidence values of the stored and new perceptions, and A and A' indicate the detection age of the stored and new perception, respectively.

IV. Assignment Problem Formulation

As described in the previous section, the P-LDM contains a database with the entire context under the platoon perception range. Within this context, each non-connected vehicle stored as PPOs is periodically assigned to a given Platoon Member (PM). For a given N number of PMs and M number of PPOs, we denote as $\mathbf{P}$ a NxM binary matrix whose elements $p_{n,m}$ are set to 1 if PM n is perceiving PPO m, and 0 otherwise, according to the following expression :

$$p_{n,m} = \begin{cases} 1 & \text{if } d_{n,m} \leq r_n(1 - 0.9\omega_o) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $d_{n,m}$ represents the distance between PM n and PPO m, r_n the perception range of PM n and ω_o the conditional occlusion factor of PPO m [30]. As shown in Figure 3, from the perception matrix $\mathbf{P}$ we derive the assignment matrix $\mathbf{X}$, which captures how each PPO is ultimately allocated to one of the PMs. We denote as $\mathbf{X}$ the set of assignments. $\mathbf{X}$ is thus given as a NxM binary matrix whose elements $x_{n,m}$ are set to 1 if a certain PPO m is assigned to PM n [15], 0 otherwise:

$$x_{n,m} = \begin{cases} 1 & \text{if } PM_n \text{ is assigned with } PPO_m \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The following constraints apply:

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

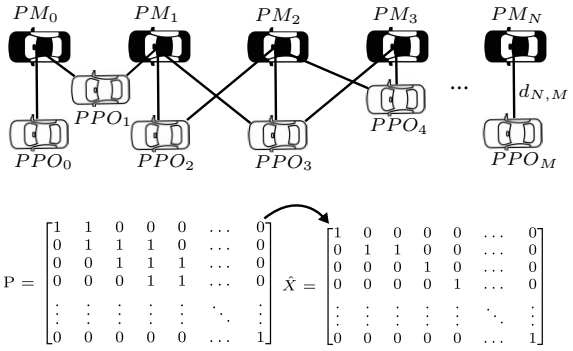


Figure 3. Relationship between the perception matrix $\mathbf{P}$ and the resulting assignment matrix $\mathbf{X}$. Each row n in $\mathbf{P}$ corresponds to Platoon Member n , while each column m corresponds to a Platoon Perceived Object m.

- A PPO can be assigned to only one PM (Responsible PM):

$$\sum_{n \in N} x_{n,m} = 1 \quad (m = 1, ..., M) \tag{4}$$

- The number of PPOs assigned to a given PM n, can be up to M:

$$\sum_{m \in M} x_{n,m} \leq M \quad (n = 1, ..., N) \tag{5}$$

- A PM can only be assigned with a PPO that is currently being perceived by it:

$$x_{n,m} \leq p_{n,m} \quad (n = 1, ..., N), (m = 1, ..., M) \tag{6}$$

  where $p_{n,m}$ is the entry at the at the n^{th} row and m^{th} column of the binary Perception matrix P.

We denote $\mathbf{R}$ as the PPO redundancy vector, whose elements are redundancy factors R_m, each for a given PPO m, representing the number of PM n currently perceiving it:

$$R_m = \sum_{n \in N} p_{n,m} \tag{7}$$

The redundancy factor of a given PPO represents the number of available sources that can provide information about it to a PM, which, in our architecture, translates into the number of detections received within an update period from CPMs. The computational cost of processing a given PPO m by a given PM n is described by:

$$C_{n,m}(X) = [\alpha_n + \gamma_n \cdot R_m] \cdot x_{n,m} \tag{8}$$

Thus, the total computational cost of handling assignment X, i.e., the cost of processing all assigned PPOs, for each PM n is obtained by:

$$C_n(X) = \sum_{m \in M} \alpha_n \cdot x_{n,m} + \sum_{m \in M} \gamma_n \cdot x_{n,m} \cdot R_m \tag{9}$$

$$C_n(X) = \sum_{m \in M} [\alpha_n + \gamma_n \cdot R_m] \cdot x_{n,m} \tag{10}$$

where α_n is the computational load factor of PM n due to a PPO m assigned to it and γ_n the additional computational load factor due to the detection of a PPO m received through CPMs (thus multiplied by the redundancy factor R_m).

Thus, the total computational cost $C(X)$ of handling the X assignment is given by:

$$C(X) = \sum_{n \in N} C_n(X) \tag{11}$$

For which the following constraint applies:

$$C_n(X) \leq \epsilon_n^T \tag{12}$$

where ϵ_n^T the total available computational resources that PM n can allocate.

The computational fairness cost $L(X)$, can be described by the difference between the computational cost of each PM n and the average computational cost of the other $PM_{i \neq n}$:

$$L(X) = \sum_{n \in N} \left| C_n(X) - \frac{1}{1 - N} \sum_{i \neq n} C_i(X) \right| \tag{13}$$

The probability of a given PM n to lose track of a given PPO m assigned to it, before the following assignment period $t + \Delta t$, is assumed to be proportional to the distance between PM n and PPO m at $t + \Delta t$:

$$P(p_{n,m}(t + \Delta t) = 0 | p_{n,m}(t) = 1) \propto d_{n,m}(t + \Delta t) \tag{14}$$

where

$$d_{n,m}(t + \Delta t) = d_{n,m} + v_{n,m} \cdot \Delta t + a_{n,m} \cdot \Delta t \tag{15}$$

with $v_{n,m}$ and $a_{n,m}$ denoting the speed and acceleration difference between PM n and PPO m, respectively.

The perception robustness cost $D(X)$ of using assignment X is defined as the sum of the probabilities of each PPO m going out of perception range of the PM n assigned with it, before a new assignment is made:

$$D(X) = \sum_{n \in N} D_n(X) \tag{16}$$

where

$$D_n(X) = \sum_{m \in M} P(p_{n,m}(t + \Delta t) = 0) \cdot x_{n,m} \tag{17}$$

$$= \sum_{m \in M} d_{n,m}(t + \Delta t) \cdot x_{n,m} \tag{18}$$

In our system, the optimal assignment $\hat{X}$ is the one that jointly minimizes the computational load, the computational fairness and perception robustness costs:

$$\min \quad \{C(X), L(X), D(X)\} \tag{19}$$

$$\text{subject to:} (4), (5), (6), (12) \tag{20}$$

In the following section, a detailed description of the proposed algorithms to perform the online assignment computation.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

V. Proposed Heuristic Algorithms

Due to the highly dynamic nature of the information stored in the P-LDM, the optimal PPO assignment changes after every new P-LDM update period, requiring a quick algorithm to compute it. Furthermore, because conflicting objectives must be jointly optimized, the solution of the optimization problem is represented by a pareto-optimal set of assignments instead of a single one. The pareto-optimal solutions are the ones where no individual solution can be considered better than another when considering all objectives simultaneously.

In order to achieve an online computation of the Platoon Perceived Object assignment, the challenge of dealing with integer decision variables necessitates the use of heuristic algorithms due to the inefficiency of exhaustive methods. Many works on similar multi-objective problems have highlighted the efficacy of Genetic Algorithms (GAs) in finding optimal or near-optimal solutions. Genetic Algorithms are adaptive heuristic search algorithms based on the evolutionary behavior of natural selection and genetics. Their ability to handle multiple objectives and constraints makes them well-suited for complex optimization tasks like ours [15], [16], [31], [32].

Among Genetic Algorithms, the NSGA-II is one of the most prominent and widely utilized. NSGA-II is popular for its efficiency in handling multi-objective problems through its fast non-dominated sorting approach and its ability to maintain a diverse set of solutions. This algorithm enhances the search process by prioritizing non-dominated solutions (pareto-optimal), thereby effectively balancing the trade-offs between competing objectives [33]. Even though the NSGA-II provides a close to optimal pareto-front of PPO assignments, the convergence time becomes unsuitable for our service due to the high number of possible solutions when the number of PPOs is at least twice that of the platoon members (for platoons with more than 10 vehicles in congested traffic). Despite its long convergence times, the NSGA-II reliably generates high-quality Pareto fronts through its non-dominated sorting and crowding-distance techniques. In contrast, while nature-inspired methods such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC) offer fast convergence, they generally require extra mechanisms to ensure solution diversity. To update the PPO assignment after every P-LDM update period, there is a need for an algorithm able to compute a solution of the pareto-optimal front in less than 100ms. To this end, we have devised a greedy algorithm for the online PPO assignment selection. The idea behind the algorithm is that, instead of assigning PMs to PPOs, PPOs are assigned to PMs in order of assignment cost. Two algorithms have been devised that differ in how the PPOs are ordered before assigning them. The greedy algorithm, depicted in Algorithm 1, orders all PPOs from the least detected (i.e., the one that is detected by the least number of PMs) to the most detected.

The algorithm takes as input the current perception matrix $\mathbf{P}$ as defined in (2), the distance matrix $\mathbf{\Delta}$ with the values computed as defined in (15), the vectors with computational load factor α and computational cost coefficient due to redundancy γ of each PM. Additionally, weight coefficients ω_c and ω_d are taken as input for the computation of the combined computational and perception robustness costs of a given assignment. As the PPOs get assigned to PMs, a vector $\mathbf{Cv}$ with the accumulated CPU costs of each PM is updated to compute the fairness cost.

Algorithm 1 Greedy Algorithm

1: **procedure** GREEDY($\mathbf{P}, \mathbf{\Delta}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, w_c, w_d, \epsilon_n^T$)
2: $(N, M) \leftarrow$ dimensions of P
3: $\mathbf{R} \leftarrow [m, \sum_{n \in N} P_{n,m}]$ for $m \in M$
4: $\mathbf{C} \leftarrow$ CPU cost matrix from $(\mathbf{P}, \alpha, \gamma)$
5: $\mathbf{D} \leftarrow$ Perception robustness cost matrix from $(\mathbf{P}, \mathbf{\Delta})$
6: $\mathbf{Cv} \leftarrow$ initialize CPU cost vector of size N
7: $\mathbf{X} \leftarrow$ initialize assignment matrix $N \times M$
8: $\mathbf{R}' \leftarrow Sort(R)$ – Sorted by $\sum_{n \in N} P_{n,m}$
9: **for** each object $m \in R'$ **do**
10: $\mathbf{N}' \leftarrow [n$ for n in N if $P_{n,m} = 1]$
11: $\mathbf{Z} \leftarrow [w_C.C_{n',m} + w_d.D_{n',m}]$ for $n' \in$ N'
12: $\mathbf{\Upsilon} \leftarrow [N', Z]$
13: $\mathbf{\Upsilon}' \leftarrow Sort(\mathbf{\Upsilon})$ ▷ Sorted by values of Z
14: $c \leftarrow n'$ of $\mathbf{\Upsilon}'[0]$ ▷ PM with lowest $C(X)$,$D(X)$
15: **if** $\sum_{m \in M} X_{c,m} = 0$ **then**
16: $X_{c,m} = 1$ ▷ Assign object
17: $Cv_c \leftarrow C_{c,m}$
18: **else**
19: $\mathbf{L} \leftarrow$ additional fairness costs for each n' in $\mathbf{\Upsilon}'$
20: $\mathbf{L}' \leftarrow Sort(\mathbf{L})$
21: $l \leftarrow n'$ of $\mathbf{L}'[0]$ ▷ PM with lowest $L(X)$
22: $L_c \leftarrow \mathbf{L}'$ of member c ▷ $L(X)$ of PM from $\mathbf{\Upsilon}'[0]$
23: **if** $(Z_l - \mathbf{\Upsilon}'[0]) > (L_c - \mathbf{L}'[0])$ **then**
24: $X_{c,m} = 1$ ▷ Assign to lowest $C(X), D(X)$
25: $\mathbf{Cv_c} \leftarrow C_{c,m}$
26: **else**
27: $X_{l,m} = 1$ ▷ Assign to lowest $L(X)$
28: $\mathbf{Cv_l} \leftarrow C_{l,m}$
29: **end if**
30: **end if**
31: **end for**
32: **return** X
33: **end procedure**

At the beginning of the assignment process, a matrix $\mathbf{R}$ of size $2\text{x}M$ where each row contains the index of each PPO with the number of PMs currently perceiving it. Once the matrix $\mathbf{R}$ is created, its rows are sorted according to the number of PMs perceiving each PPO obtaining $\mathbf{R}'$. For every PPO m in $\mathbf{R}'$, two vectors are defined $\mathbf{Z}$ and $\mathbf{N}'$. The vector $\mathbf{N}'$ is a subset of N, containing the indexes n', which correspond to the PMs that perceive PPO m. On the other hand, $\mathbf{Z}$ is the vector whose elements $\mathbf{Z_{n'}}$ represent the combined computational and robustness cost of the PPO m seen by PM n'. Vectors $\mathbf{Z}$ and $\mathbf{A}$ (which have the same dimension) are then combined to create a matrix $\mathbf{\Upsilon}$,i.e. each row $\mathbf{\Upsilon_{n'}}$ contains the index of the PM and its associated cost. $\mathbf{\Upsilon}$ is computed in Line 12 and then sorted in ascending order ($\mathbf{\Upsilon}'$) according to the cost values in Line 13. The first PPO is assigned to the PM with the lowest cost (i.e., $\mathbf{\Upsilon}'[0]$). For the following assignments, the fairness cost is computed according to the current computational cost vector ($\mathbf{Cv}$). The fairness cost of each potential assignment is stored in a vector $\mathbf{L}$ and sorted in ascending order ($\mathbf{L}'$) in Line 20. In line 23, the algorithm compares the Computational and Robustness cost ($C(X), D(X)$) difference with the Fairness cost ($L(X)$)

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

difference. The former is the difference in assignment cost if the object is assigned to the node with the lowest $C(X), D(X)$ cost versus assigning it to the PM that achieves the best $L(X)$, whilst the latter is the difference in $L(X)$ if the object is assigned to the PM achieving the best fairness versus the node with the lowest $C(X), D(X)$ cost. If the difference in $C(X), D(X)$ cost for the PM with best $L(X)$ is greater than the difference in fairness cost for the PM with the best $C(X), D(X)$ cost, then the PPO is assigned to the PM with the lowest $C(X), D(X)$. Otherwise, it is assigned to the PM providing the lowest $L(X)$ (i.e., $\mathbf{L}'[0]$).

The idea of sorting PPOs according to the number of PMs currently detecting them, starting from the least perceived, comes from the fact that the least detected object has fewer candidates to assign it to. Hence, this strategy avoids assigning objects to PMs that are not close enough to the object and cannot ensure a robust perception. However, this strategy has one downside, namely that the most perceived object will yield the highest computational cost due to its high redundancy factor, as explained in Section IV. Indeed, assigning the objects with the highest redundancy factor as last, results in poor granularity in the computational cost of assignments, thus lowering the possible achievable fairness. In order to analyze the trade-off between the different approaches, we have considered two possible versions of our greedy algorithm for the performance evaluation presented in this work, which we have called *least2most* and *most2least*. They only differ by the sorting order performed in Line 8 of Algorithm 1, with the *least2most* sorting them from least detected to most detected whereas the *most2least* sorts them from the most detected to the least detected.

For what regards the computational complexity of both *least2most* and *most2least* can be analyzed as follows. The outer loop, iterating over each PPO in $\mathbf{R}$, thus being performed M times. Within this loop, the process for computing $\boldsymbol{\Upsilon}$ has a complexity of $\mathbf{O}(N)$. Additionally, sorting assignable PMs incurs a complexity of $\mathbf{O}(N \log(N))$. Lastly, the computation of the fairness costs vector $\mathbf{L}$, a comparison between all PMs' computational and distance cost needs to be performed, rendering a complexity of $\mathbf{O}(N^2)$. Finally, the overall computational complexity for both algorithms is $\mathbf{O}(MN^2)$.

A. Performance Evaluation

Even though the NSGA-II is not suitable for the time requirements of our system, it serves as a reliable benchmark, enabling us to clearly investigate the trade-off between solution optimality and the computational efficiency achieved by our custom greedy algorithms. Thus, to evaluate the performance of our proposed algorithms, we have developed a custom python implementation of the NSGA-II algorithm as well as the greedy algorithm described. Our custom implementation includes all the functionalities of the NSGA-II algorithm with tailored implementations for population initialization, crossover, and mutation to ensure our constraints are met. In particular, in our implementation, all members of the population (i.e. matrix X defined in Section IV) are such that every PPO is assigned to a single PM. For the initial simulation campaign, we considered a scenario with a platoon size of 10 Platoon Members and 25 Platoon Perceived Objects. For each simulation, the position of PPOs is randomly generated and then arranged so that each one is perceived by at least one PM. Each computational load factor associated with assigning a PPO to a PM (i.e., the assigned values of α_n defined in (8) has been randomly selected from the set [0.05, 0.1, 0.2] to account for different computational capabilities amongst PMs. In a similar manner, the cost factor due to the redundancy of a PPO (i.e., the assigned values of γ_n defined in (8) has been randomly selected from the set [0.01, 0.05, 0.1]. For the configuration of the NSGA-II algorithm,the parameters have been determined through a series of preliminary experiments on these base scenario. Initially adopting the common default settings reported in the literature and tested various combinations of these parameters for a trade-off between convergence speed and the stability of the Pareto front. Ultimately, a population of 100 assignments was used with a mutation rate of 0.1, a crossover probability of 0.9 and a total of 100 generations. Being the output of the NSGA-II, a Pareto front of non-dominated solutions, for each simulation, only the solution with the lowest weighted average (with equal weights) is considered.

All algorithms have been implemented as Python functions to be called for each randomly generated PPO position, α_n, γ_n, and perception matrix. A wrapping Python script allowed us to measure both the optimization performance and execution time under identical conditions for all algorithms. The simulations have been performed on an off-the-shelf AMD Ryzen 7 5800H processor. The results in terms of optimization performance are shown in Table I. As our objective is to minimize all the costs defined in (19), the lowest values reported represent the best assignments. The NSGA-II provides the best assignment compared to the greedy algorithms across all the objectives of interest, providing the best assignment of the three algorithms. As expected, the *most2least* algorithm achieves a lower fairness cost compared to the *least2most* counterpart. These results showcase the benefits of assigning the objects that are perceived by most PMs first, as those are the more 'expensive' ones, hence, the ones requiring the best possible assignment. When considering the perception robustness cost, the *least2most* algorithm archives a slightly better performance compared to the *most2least*. As anticipated, the *least2most* achieves a better assignment for the PPOs perceived by the least PMs, statistically further away than the rest of PPOs. As in our assignment problem, the objective is to jointly minimize all three computational, robustness, and fairness costs; the resulting weighted average is reported in Table I as well. Intuitively, the NSGA-II algorithm provides the best performance, with the *least2most* being the best of the greedy algorithm solutions.

As hinted at the beginning of the section, even though the NSGA-II provides the best results of the considered heuristics, the performance showcased is achieved at the expense of a considerable number of iterations. As can be seen in Table I, the NSGA-II takes over 30 seconds on average to converge, which renders it unsuitable for the target assignment period of 100ms. On the other hand, both greedy algorithms produced

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

Table I
PERFORMANCE COMPARISON OF EACH ALGORITHM IN TERMS OF OPTIMIZATION COSTS AND TOTAL EXECUTION TIME.

Performance metric	NSGA-II	least2most	most2least
Computational Cost	**0.188**	0.206	0.232
Perception Robustness Cost	**0.312**	0.415	0.442
Fairness Cost	**0.304**	0.320	0.306
Weighted Average Cost	**0.268**	0.312	0.318
Execution time [ms]	34956	1.977	**1.863**

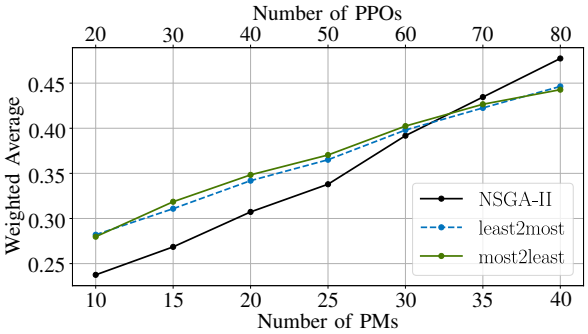


Figure 4. Weighted average of PPO assignment cost of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members.

an assignment selection in under 2ms, on average.

B. Scalability

Based on the performance evaluation, NSGA-II outperformed the two greedy algorithms. However, as the computation time for the NSGA is over a full minute on average for a scenario with 10 PMs and 25 PPOs, the greedy algorithms become the only viable solution for the online assignment of PPOs. We have performed a scalability analysis to evaluate the performance of all algorithms in terms of optimization performance and execution time under more demanding scenarios.

The optimization cost outcomes of doubling the number of PPOs along with increasing the number of PMs are depicted in Figure 4. As can be seen, the NSGA-II outperforms both greedy algorithms up to a scenario with 30 PMs and 60 PPOs, after which both greedy algorithms provide better results. These results reflect the limitation of the NSGA-II for extensive search spaces, which require a bigger population and more generations to achieve good enough results. Indeed, when both the number of PM and PPOs are increased, the search space grows exponentially, reflected in the execution time for all the algorithms, depicted in Figure 5.

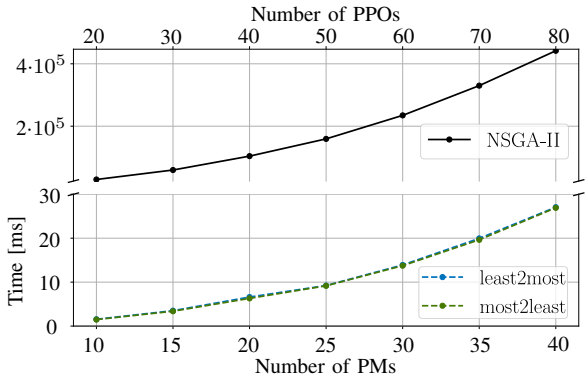


Figure 5. Average execution time of each algorithm for an increasing number of Platoon Members, where the number of Platoon Perceived Objects is twice the number of Platoon Members.

The scalability results for the NSGA-II algorithm are shown here only as a reference, since its execution time was prohibitively high even for the base scenario described in Section V-A. As pointed out, the Genetic Algorithm needs different configurations for different search spaces to obtain the best trade-off between execution time and performance. To simplify the comparison, we have always considered the same population size, number of generations, and mutation rate, which have resulted in the Genetic Algorithm being over-dimensioned for simple cases and under-dimensioned for the more complex ones. Indeed, the results of our scalability analysis show that for complex scenarios with 40 PMs and 80 PPOs to be assigned, our custom greedy algorithms not only outperform the NSGA-II algorithm but also achieve execution times lower than 30ms, below the target of 100ms needed by the P-LDM.

VI. SERVICE VALIDATION

In addition to the presented service, we have implemented the platooning protocol presented in [4] for the ENSEMBLE project, leveraging 2 newly defined messages:

- Platoon Management Messages (PMM): messages for handling join requests and join responses for the management of a given platoon.
- Platoon Control Messages (PCM): messages containing all necessary information for the Platoon Members' Cooperative Adaptive Cruise Control (CACC) controller. This message is transmitted with a frequency of 20Hz.

All the Platoon Members in our simulations are controlled by the CACC controller implemented in ms-van3t based on the one defined in [28], commonly leveraged in other simulation frameworks such as Plexe [34]. The CACC controller works under a constant spacing policy to secure string stability along the platoon, relying on the dynamic information of the platoon leader and the preceding platoon member. According to the information available about other PMs received through PCMs, and the information about PPOs found in the LDM (or P-LDM) database, each PM_i applies the desired acceleration given by the controller. Once the desired acceleration is computed, it is then passed to the OpenCDACI module where it gets translated into a control input (throttle, brake and steering) leveraging OpenCDA's PID controller.

A. Simulation setup

A simulation campaign has been performed to jointly evaluate the key aspects of the cooperative perception concept that our system aims to improve:

- Computational load

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

- Perception Robustness
- Platoon awareness

To this end, we have designed a custom CARLA map consisting of a 4-lane 3-km strip of highway with an entrance ramp merging with the main highway at the 1.2 km mark. In addition to platooning vehicles, a total of 60 background vehicles are evenly distributed on the strip to achieve a 20 vehicle/km density. Lastly, an intruder vehicle is spawned on the entrance ramp set to interfere with the platoon execution. In terms of the cooperative perception scheme, the two scenarios considered for the simulations were:

- P-LDM: representing the scenario on which each PM runs the P-LDM service and attempts to match only the objects with the IDs assigned to it.
- Platoon Cooperative Perception (Platoon CP): representing a scenario in which the PL and PMs process the information of all objects detected by other PMs.

To evaluate the trade-offs on both schemes, two different ITS penetration rates have been considered for the simulated background vehicles, namely, 0.1 and 0.5. The ITS penetration rate determines the ratio of background vehicles equipped with a communication device for the exchange of V2X messages. Hence, a penetration rate of 0.1 yields a scenario in which only 10% of vehicles traveling on the highway are V2X-enabled, representing a near-future case. The remaining background vehicles are potential PPOs for the platoon. Regarding the V2X-enabled vehicles communication setup, they are equipped with 802.11p OBUs on a 20 MHz channel at a central frequency of 5.9 GHz, configured with a physical data rate of 12 Mbit/s, and a transmission power of 26 dBm. As for the perception configuration, all V2X-enabled vehicles are equipped with four RGB cameras, whose feeds are processed by the YOLOv5 object detection model. Additionally, a LiDAR sensor is configured, and its output is used for late fusion to achieve 3D object detection. Furthermore, the platoon sizes were set to 10, 15, and 20, respectively, to analyze the effect of the platoon size on the perception range and platoon reaction to an intruder vehicle. For this evaluation we have focused on platoons of at least 10 vehicles due to the fact that in smaller platoons (e.g., 5 vehicles), the perception range of each member would normally cover the platoon's area of interest. As a result, the added benefits of cooperative perception become less relevant, and the number of perceived objects would be too small to demand for a sophisticated assignment mechanism such as the P-LDM. On the other hand, although we have considered platoons of up to 40 PM in the scalability analysis of Section V-B, for the service validation presented here, the largest platoon size we have considered is 20. Even if our assignment algorithm achieved execution times well below our target of 100 ms for 40 PMs, allowing the service to run without loss of performance, the deployment of platoons of such dimensions has limited practical value. Constraints due to latency and packet loss [35], as well as reduced traffic flow benefits [36] [37], keep platoons below 20 vehicles in real-world conditions. All results presented in this section are averaged over 100 simulations, each lasting 60 seconds, for every combination of penetration rate and platoon size. To capture diverse traffic dynamics, the initial positions of background vehicles were randomized across simulations. The intruder vehicle, representing a human-driven vehicle, has been simulated to systematically interfere in the middle of the platoon by cutting in between two members (depicted in phase 3 of Figure 6), relying on the ground truth information from the CARLA simulation for the maneuver. Instead, all platoon members rely on the information from platoon control messages and the LDM or P-LDM databases, with which, at every CACC controller step, a check is performed to verify the vehicle or perceived object in front to adjust its acceleration accordingly. Lastly, the PPO assignment selection for the P-LDM scenario has been performed with the *least2most* algorithm detailed in Section V.

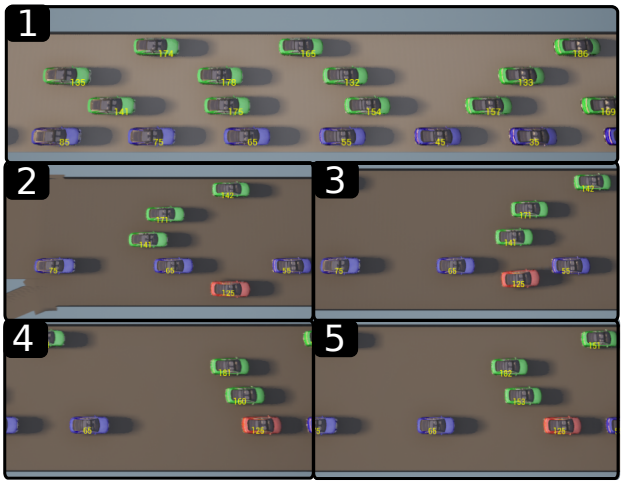


Figure 6. Scenario timeline showing the different phases of the considered simulation scenario: 1. Normal Platooning state (PMs colored in blue and background vehicles colored in green). 2. Intruder vehicle (colored in red) approaches the platoon. 3. Intruder vehicle starts the cut-in maneuver between two PMs. 4. PMs react to the intruder interference by slowing down to create a safe gap. 5. The platoon adjusted to the intruder's presence.

B. Perception computational load analysis

One of the main purposes of the P-LDM is to distribute the processing power dedicated to cooperative perception among all the members of a platoon of vehicles. Indeed, upon reception of a CPM, each PM attempts to match each received object with the ones stored in its own LDM database. As the number of objects stored in the LDM and the number of objects included on each CPM increases, so does the processing time needed to perform such matching. The perception matching task in our framework is formulated as an assignment problem, where the goal is to assign CPM objects to the existing entries of the LDM in an optimal way, creating new entries for the cases on which there is no match. To perform the matching, we leverage the algorithm provided in the SciPy Python package based on the algorithm devised in [38], which creates a matching matrix by evaluating each received object against existing entries according to the IoU and the distance as described in Section III-C.

In our framework, each simulation utilizes two primary processes in addition to the CARLA server: one for ns-3 and one for OpenCDA. Consequently, all vehicle perception

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

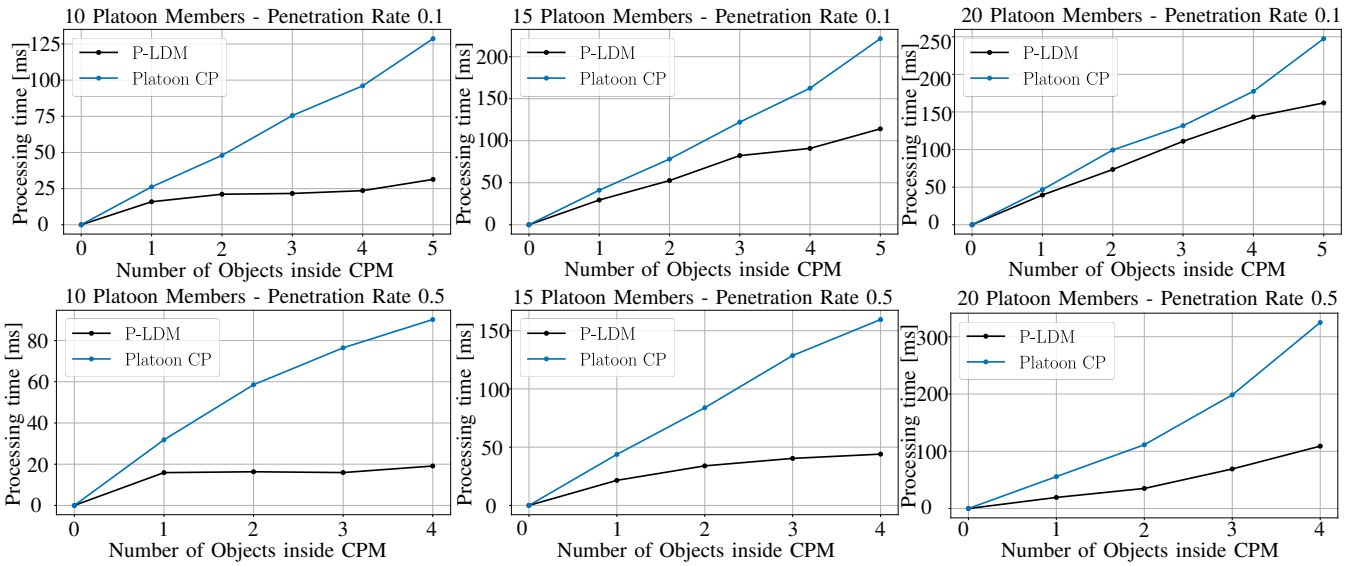


Figure 7. Average processing time of each received message (CPMs) for each number of Platoon Perceived Objects included in each message.

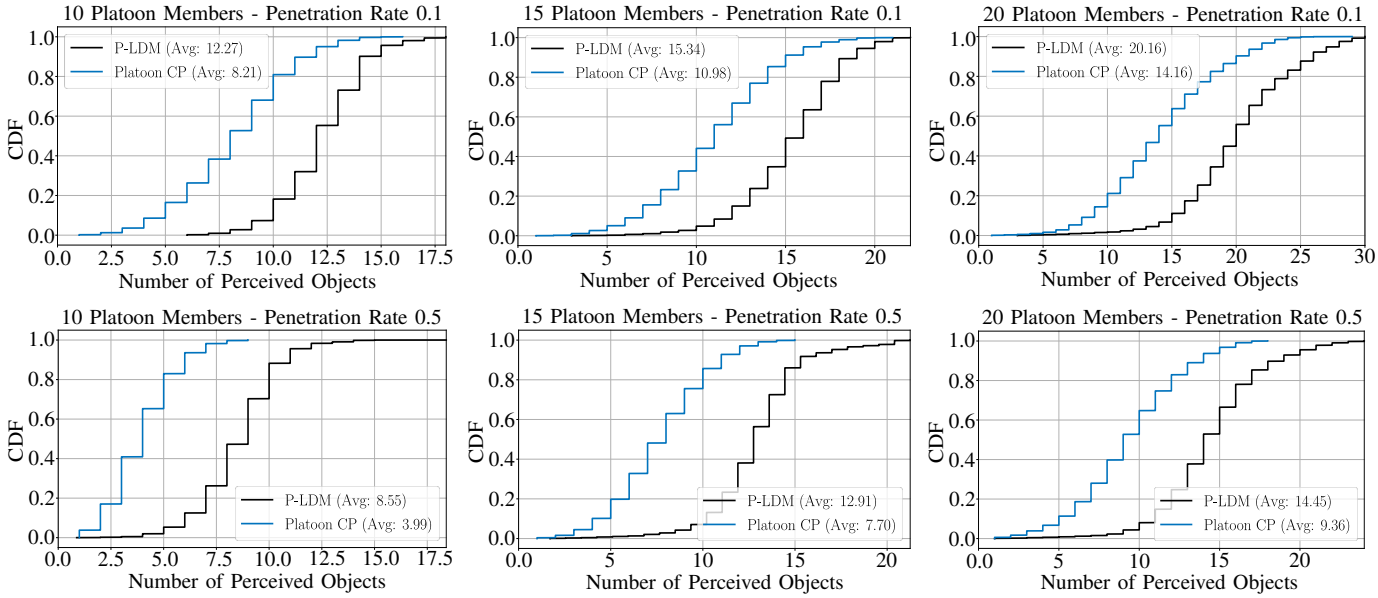


Figure 8. Cumulative Distribution Function of the total number of Platoon Perceived Objects stored in the database at a given point in time.

processing within the simulation is performed sequentially by OpenCDA, using the same hardware resources. This sequential processing approach prevents real-time simulation and makes CPU usage an unsuitable metric for evaluating our service. Therefore, we have chosen processing time as the metric to evaluate the computational load of each scenario. Specifically, we focus on the processing time of CPMs, as they are the most demanding task in our cooperative perception pipeline, apart from local perception. For each simulation in our campaign, the processing time of each CPM is measured from the moment the message is received and passed on to the message handler running in OpenCDA until all objects have been updated in the database. The time measured includes the creation of the bounding box, the matching process according to the intersection over union, and the update or creation of the Kalman filter for each PPO received.

The processing time of a CPM at time t depends on the number of objects stored in the vehicle's database at that time instant and the number of objects included on the received CPM, translating into the dimensions of the matching matrix. However, in the case of the P-LDM, thanks to the ID synchronization of objects and the assignment scheme used, only the objects assigned by the Platoon Leader to the PM receiving the CPM are processed to be matched, while the rest are discarded. As can be seen from Figure 7, in the Platoon CP scenario, the processing time of CPMs increases linearly with the number of objects included in each CPM and the number of objects being stored and maintained by the receiving PM. On the other hand, when leveraging the P-LDM, the processing time increases linearly up to the average number of assigned objects, reported in Table II for each platoon size considered and a penetration rate of 0.1 (representing the most demanding

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

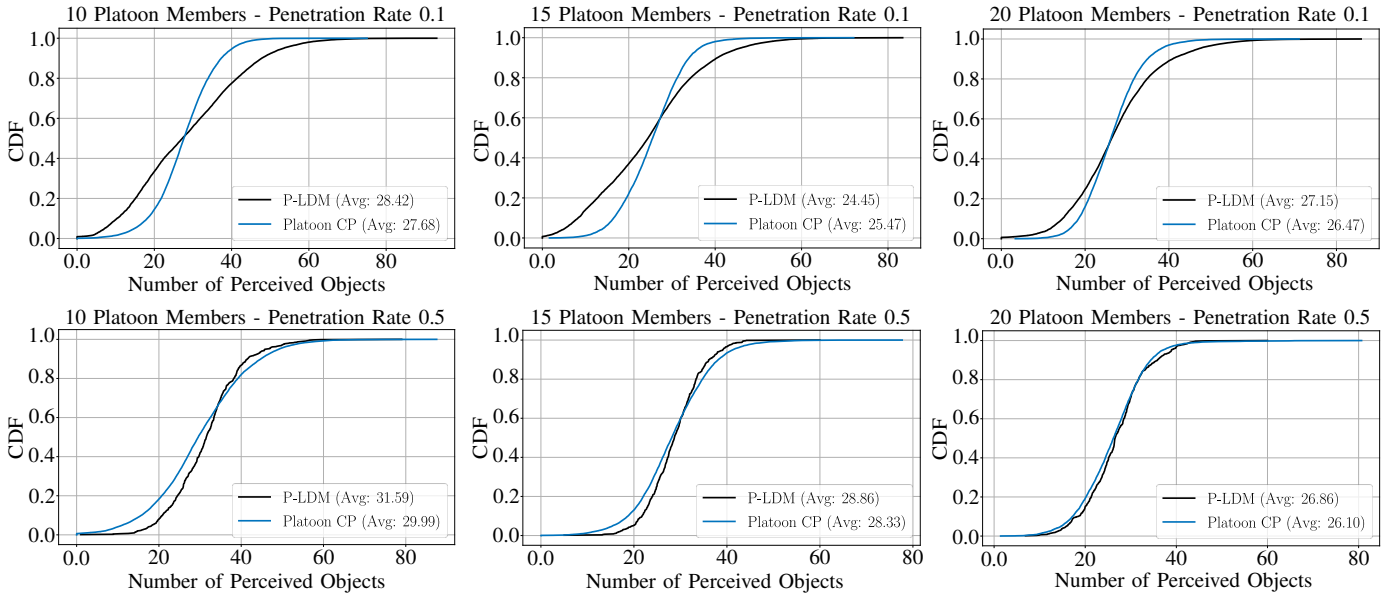


Figure 9. Cumulative Distribution Function of the average IoU of all objects stored in the database at a given point in time.

case). Indeed, for CPMs with more objects than the ones assigned to each PM, the processing time increases with a flatter behavior compared to the Platoon CP counterpart, where all objects are processed. As the number of Platoon Members increases, the processing time for each CPM becomes more dependent on the total number of objects stored in the database as opposed to the number of objects present on each CPM.

Intuitively, for a higher penetration rate, the number of objects stored in the database decreases for both scenarios, as showcased in Figure 8. As a result, when considering the P-LDM, each PM is assigned with fewer PPOs, achieving lower processing times for the same number of objects included in CPMs. On the other hand, for the Full CP scenario, there is no significant difference in the processing time for equally sized CPMs when the penetration rate is increased. It is worth noting that even if there are fewer perceived objects for higher penetration rates, both connected and detected vehicles are considered for the matching process. Indeed, when changing the penetration rate, the combined sizes of PPOs and CVs stored in the databases remain stable.

Table II
AVERAGE NUMBER OF PLATOON PERCEIVED OBJECTS BEING HANDLED, I.E., BEING PROCESSED FOR EACH CPM RECEPTION.

Penetration Rate	Platoon Members	P-LDM	Platoon CP
0.1	10	1.83	8.21
	15	2.01	10.98
	20	2.28	14.16
0.5	10	1.55	3.99
	15	1.78	7.70
	20	1.96	9.36

C. Perception robustness

The cooperative perception concept has been devised to expand the local view of each CV for the execution of the so-called Day-2 and beyond [39] set of applications, such as platooning. Indeed, as the vehicle 'view' of the road broadens, maneuvers are executed in a more timely and safe manner. As outlined in Section II, many challenges arise in keeping an updated view of the road through the distributed exchange of CPMs without creating unnecessary redundancy, resulting in both communication and computational overhead.

For our simulation campaign, in addition to measuring the performance of the P-LDM in terms of computational load, we have measured the achieved robustness of the cooperative perception of the road. The two Key Performance Indicators (KPIs) we have chosen for perception robustness are the total number of PPOs stored in the database and the average IoU of all PPOs. For each simulation step (50 milliseconds), the database of each Platoon Member has been compared to the ground truth extracted from CARLA to compute the average IoU of all perceived objects stored. For both P-LDM and Platoon CP, each entry in the database is retained for up to 2 seconds. If not updated by a V2X message or local perception, it is deleted. Such criterion has been taken considering that all perceived objects in both scenarios are, in fact, highly dynamic vehicles. Indeed, outdated perceptions can hinder the execution of applications relying on them, such as platoon control.

The results for the different numbers of platoon members and penetration rates are shown in Figure 9, where the Cumulative Distribution Function of the measured average IoU is depicted for both P-LDM and Platoon CP scenarios. When considering a penetration rate of 0.1, the distribution of Platoon CP initially provides higher IoU values than the P-LDM for all platoon sizes. This behavior is obtained due to local perceptions being updated with a higher frequency compared to the P-LDM messages, resulting in 'fresher' (more accurate) perceptions. In the Platoon CP scenario, the average

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

perception accuracy is biased by the high accuracy of local perception, as most stored perceptions are indeed local at any given time. On the other hand, the P-LDM entries are updated only by the assigned PM. As a result, objects are updated solely through P-LDM messages, even if they are perceived locally. Thus, the average of all the perceptions at a given point in time does not contain the bias from local perception, resulting in a flatter distribution. Moreover, to ensure computational fairness alongside perception robustness, the assignment algorithm might allocate objects to PMs that are not the closest. This is particularly noticeable for objects located at the edges of the platoon, which, together with the skipped updates when an object assignment is changed from one PM to another, results in lower average accuracy measurements by the P-LDM for 50% of the time. Nevertheless, for all the considered cases, except for 15 Platoon Members and a penetration rate of 0.1, the P-LDM achieves a slightly higher perception accuracy overall. It is important to note that while the P-LDM does not achieve a significant increase in perception accuracy, it does significantly increase the number of perceived objects stored in the database, as demonstrated in Figure 8, without sacrificing accuracy.

When analyzing the cases with a penetration rate of 0.5, the overall accuracy shows similar values for both scenarios. In the case of Platoon CP, the local perception bias fades as there are fewer local perceived objects to generate it. Indeed, both distributions show similar trends, with even the P-LDM showing a sharper distribution product of fewer objects to be assigned to PMs, allowing for a better assignment in terms of perception robustness.

D. Platooning awareness

After analyzing the benefits of leveraging the P-LDM for computational load and perception robustness, we now evaluate how they translate into the effective execution of platooning maneuvers. To this end, we have configured our simulations to include an intruder vehicle to intervene with platoon control execution by cutting in between two members. The awareness of this intruder representing a legacy, non-connected, and human-driven vehicle relies solely on the sensor perception of PMs. Thus, the platoon's timely and safe reaction will depend on the achieved cooperative perception.

The results for the platoon speed profile is shown in Figure 10 for a platoon size of 10 members leveraging the P-LDM on the left and leveraging the Platoon CP scheme on the right. At the beginning of each 60-second simulation, the platoon is already formed, and all vehicles are stationary. The PL, programmed to reach a desired platoon speed of 20 m/s, starts speeding up, and all PMs adjust their control inputs accordingly to match the PL's speed profile, as depicted in Figure 10 from 0 to 30 seconds of simulation runtime. Once the entire platoon reaches the desired speed (phase 1 in Figure 6), the intruder vehicle traveling along the merging entrance ramp adjusts its speed to arrive at the merge lane after 35 seconds (phase 2 in Figure 6), simultaneously with the platoon. After an adjusting period, in which the intruder reaches a suitable spot between 2 PMs, the cut-in maneuver is performed (phase 3 in Figure 6), highlighted with a grey background in Figure 10. As hinted at the beginning of the section, all PMs perform an obstacle check on each controller step. For this check, either the PM's LDM (in the Platoon CP scenario) or P-LDM is queried to provide all potential vehicles to be found in front of it. When a vehicle different from the preceding PM is found to be the immediate vehicle in front, all the information needed for the CACC controller calculations is taken from the LDM/P-LDM instead of the dedicated Platoon Control Messages. After the intruder has started interfering with the platoon, and as soon as the PM behind is aware of it, the platoon performs the necessary adjustments to restore the original distance between members as if the intruder were a new member (phases 4 and 5 in Figure 6). It is worth mentioning that the intruder is configured to follow the platoon until the end of the simulation, locking its speed onto the one of the Platoon Leader. As can be seen, when the intruder starts interfering with the platoon, the preceding vehicle and the rest of the platoon members reduce their speed until the desired inter-distance of 10 meters with the intruder is restored. Since the desired acceleration computed by the CACC controller depends not only on the distance with the vehicle in front but also on the speed, the speed estimation of the intruder perception is key for the correct platoon reaction. When comparing the behavior relying on the P-LDM with the one of the Platoon CP, it can be observed that the initial speed reduction by the PMs is much more abrupt in the Platoon CP. The reason behind these different behaviors is the different speed estimation achieved in the two scenarios, with the P-LDM deriving a more accurate value by keeping track of the intruder longer than the Platoon CP does. In the Platoon CP scenario the intruder's speed estimation is mostly computed through local perceptions with sporadic updates from received CPMs. Given that the speed estimation in our system is done using a Kalman filter with position samples, the longer the object is tracked, the more accurate the speed estimation will be. Thus, since the P-LDM perception of the intruder is tracked for a longer time, the speed estimation is more accurate when the cut-in maneuver is performed by the intruder, and the PM does not overreact.

It is worth mentioning that the results shown in Figure 10 correspond to a single simulation, and it is shown here for descriptive purposes. To provide a more comprehensive view of the results, we have aggregated the results into three KPIs comparing the platoon reaction to an intruder: the average speed difference between PMs and PL, the minimum distance measured between PMs and the minimum Time To Collision between PMs. All the KPIs are computed with the measurements from the moment the cut-in maneuver begins (at 40 seconds) until the end of the simulation. Intuitively, the results for the mentioned KPIs are more influenced by the platoon size compared to the penetration rate, with the latter affecting the distribution of vehicles to be detected and not affecting the perception of a single vehicle (the intruder). For this reason, we only present results for a penetration rate of 0.1.

The average speed difference between PMs and PL has been computed with the values of all speed differences between each PM and the PL along the duration of the intrusion,

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

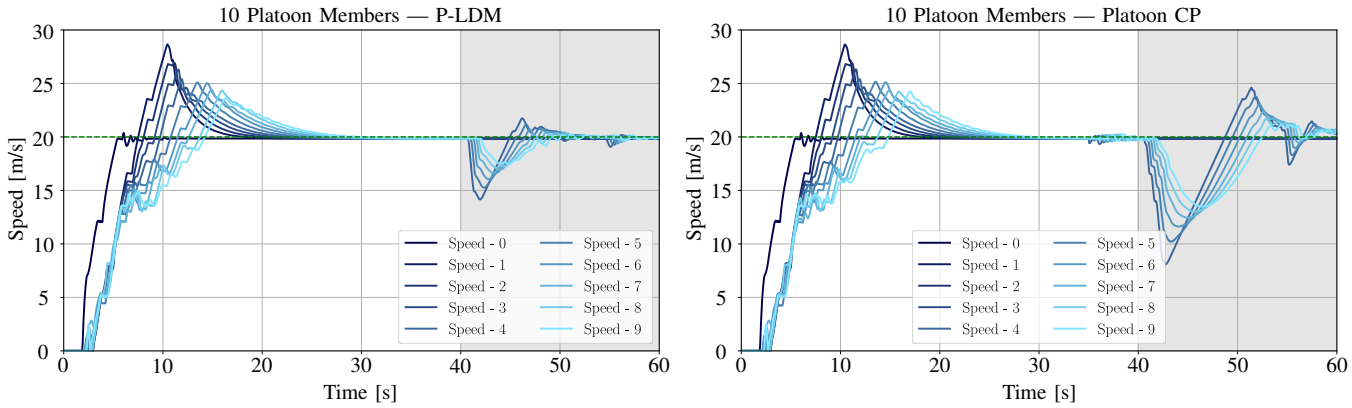


Figure 10. Instantaneous speed of each Platoon Member over time in m/s. The green dashed line at 20 m/s represents the desired platoon speed. The grey background signals the presence of the intruder within the platoon.

capturing both the speed adjustment when the intruder cuts in as well as the recovery to nominal speed. As showcased in Figure 11, for the case of 10 platoon members, the P-LDM achieves a much lower speed difference, resulting in a safer and faster reaction. Indeed, the minimum measured distance between PMs is not lower than 7 meters on average, as opposed to 3.5 meters for the Platoon CP scenario. However, as the platoon size increased, the P-LDM scenario showed a remarkable performance deficit compared to Platoon CP for the same number of members. Instead, since local perception is the main source for the Platoon CP, very similar results are obtained for all platoon sizes. The reason behind the worse performance of the P-LDM with a higher number of PMs lies in the higher number of candidates for the assignment of the intruder object, resulting in many more assignment changes during the considered intrusion. Such changes in assignment, from one PM to another, result in the control process using delayed perceptions of the intruder. Although an abrupt change in performance is obtained for the average speed difference, which is affected by the duration of the platoon reaction, the minimum distance and minimum time to collision measurements show a more linear performance decrease. Thus, with the P-LDM, a safer platoon reaction is achieved compared to the Platoon CP baseline for all platoon sizes, albeit with slower re-adjustment when platoon sizes are increased.

In summary, we have shown how the performance of the P-LDM service is sensitive to platoon size and penetration rate. As the platoon size increases, each PM's database grows, leading to longer processing times due to the higher number of objects that must be handled. However, the P-LDM's assignment strategy limits the processing load per PM, effectively mitigating this effect. Meanwhile, a higher penetration rate reduces the overall number of perceived objects, thereby decreasing the computational load and processing time. Regarding perception robustness, as the penetration rate increases, the P-LDM offers an improved performance compared to the CP counterpart due to less local perception bias. For platooning awareness, smaller platoon sizes yield more responsive adjustments and safer inter-vehicle distances, while larger platoons may experience increased assignment changes that slightly delay recovery. Overall, the sensitivity analysis confirms that the P-LDM service remains robust under varying conditions, with performance improvements particularly notable in scenarios with lower penetration rates and smaller platoon sizes.

VII. Conclusion

In this paper, we have proposed and evaluated the Platoon Local Dynamic Map (P-LDM) service, featuring a hierarchical cooperative perception scheme to efficiently and effectively achieve enhanced context awareness for platooning vehicles. Our system addressed the challenges introduced by the presence of highly redundant context information by distributing the context data aggregation among all the Platoon Members (PMs), thus avoiding overloading them with perception tasks.

The P-LDM creates a unified database of context data for the entire platoon, orchestrating the data aggregation process by assigning specific portions of the context to each PM. Each PM is responsible for updating the portion of the context data that has been assigned by the PL, who is in charge of database management. A set of V2V messages has been devised for periodically updating the portion of context information assigned to each PM and for periodically updating the assignment of PPOs by the PL.

The PPO assignment selection problem has been formulated and solved as a multi-objective optimization problem seeking to optimize the total computational load, the computational fairness among PMs and the context's perception robustness. We have developed custom heuristic solutions and compared them with the well-established Non-dominated Sorting Genetic Algorithm II (NSGA-II). Our performance and scalability evaluation for the considered heuristic algorithms showcased the superior efficiency of our proposed algorithms compared to NSGA-II while obtaining comparable optimization results.

We validated our service using a novel simulation framework that combines CARLA and ns-3, allowing for accurate modeling of perception processing, vehicle control, and wireless communication. In addition to a new enhanced implementation of our proposed service, first presented in [5], we

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

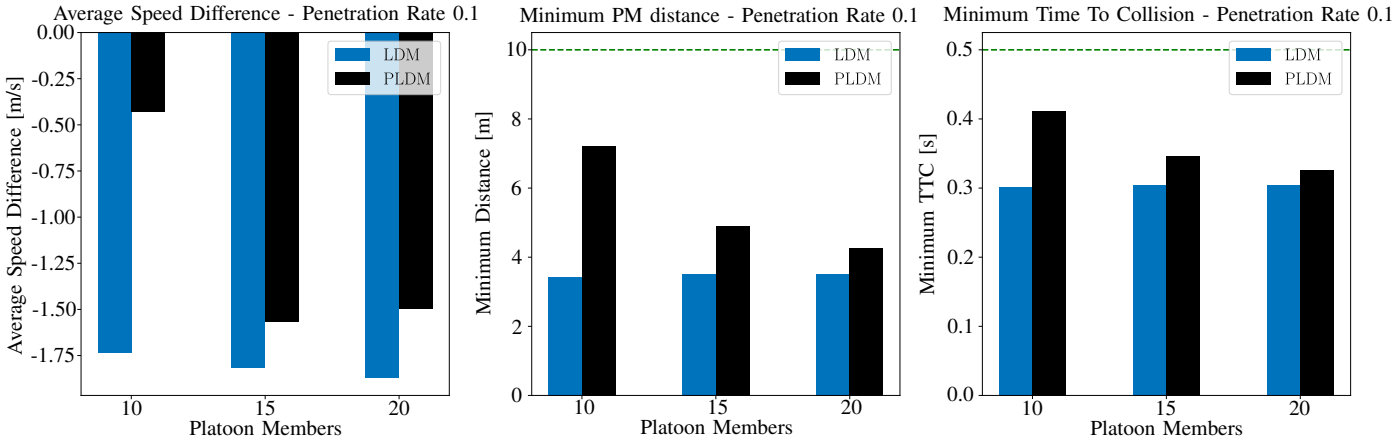


Figure 11. Average member speed difference with platoon leader at intruder cut-in, average member inter-distance at intruder cut-in and average Time To Collision between members at intruder cut-in.

have implemented the field-tested platooning protocol defined in [4]. Confronting our solution to ETSI's baseline scheme for cooperative perception, in which all PMs process received messages in a fully decentralized manner, the P-LDM achieves superior efficiency in terms of message processing time and an extended context perception without loss of accuracy. Moreover, thanks to the more stable context awareness of the P-LDM, a more effective and safer platoon reaction to interfering human-driven vehicles is obtained.

Future work will include integrating advanced machine learning techniques for the object assignment procedure, where data-driven models could adaptively learn optimal task allocation strategies and improve perception robustness under varying traffic conditions. We also foresee extending the P-LDM architecture to other vehicular applications beyond platooning, such as Maneuver Coordination use cases [40], where the hierarchical perception could provide a more scalable and reliable view of the shared environment. By broadening the scope of P-LDM and leveraging contemporary AI models, it will be possible to handle an even larger number of perceived objects and more complex application requirements, ultimately enhancing overall safety and efficiency in future connected and automated transportation systems.

ACKNOWLEDGMENTS

This work was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program RESTART) and through the project CONNECT under grant agreement no. 101069688.

REFERENCES

[1] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Generation of cooperative perception messages for connected and automated vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 336–16 341, 2020.

[2] G. Volk, Q. Delooz, F. A. Schiegg, A. Von Bernuth, A. Festag, and O. Bringmann, "Towards realistic evaluation of collective perception for connected and automated driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 1049–1056.

[3] ITS Deployment Evaluation Executive briefing. [Online]. Available: https://www.connectedautomateddriving.eu/blog/platooning-becomes-a-reality-in-europe/

[4] J. van de Sluis and J. F. de Jongh, "V2x communication test tool for scenario-based assessment of truck platooning." in *VEHITS*, 2023, pp. 135–143.

[5] C. M. R. Carletti, C. Casetti, J. Härri, and F. Risso, "Platoon-local dynamic map: Micro cloud support for platooning cooperative perception," in *2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2023, pp. 405–410.

[6] ETSI, "ETSI EN 302 895 V1.1.1 (2014-09) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)," European Telecommunications Standards Institute, Standard ETSI EN 302 895 V1.1.1, 2014.

[7] ——, "ETSI EN 302 637-2 V1.4.1 (2019-04) - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," European Telecommunications Standards Institute, Standard ETSI EN 302 637-2 V1.4.1, 2019.

[8] ——, "ETSI TS 103 324 V2.1.1 (2023-06) - IIntelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2," European Telecommunications Standards Institute, Standard ETSI TS 103 324 V2.1.1, 2023.

[9] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Redundancy mitigation in cooperative perception for connected and automated vehicles," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[10] C. Pilz, P. Sammer, E. Piri, U. Grossschedl, G. Steinbauer-Wagner, L. Kuschnig, A. Steinberger, and M. Schratter, "Collective perception: A delay evaluation with a short discussion on channel load," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, pp. 506–526, 2023.

[11] Q. Delooz, A. Willecke, K. Garlichs, A.-C. Hagau, L. Wolf, A. Vinel, and A. Festag, "Analysis and evaluation of information redundancy mitigation for v2x collective perception," *IEEE Access*, vol. 10, pp. 47 076–47 093, 2022.

[12] Q. Delooz, A. Festag, A. Vinel, and S. C. Lobo, "Simulation-based performance optimization of v2x collective perception by adaptive object filtering," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–8.

[13] Z. Shen, L. Zhu, W. Qin, Y. Zhi, and C. Li, "A voi based redundant mitigation strategy in cooperative perception services of cavs," in *2024 IEEE/CIC International Conference on Communications in China (ICCC)*, 2024, pp. 985–990.

[14] G. Hattab, S. Ucar, T. Higuchi, O. Altintas, F. Dressler, and D. Cabric, "Optimized assignment of computational tasks in vehicular micro clouds," in *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, ser. EdgeSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–6. [Online]. Available: https://doi.org/10.1145/3301418.3313937

[15] J. Jijin, B.-C. Seet, and P. Chong, "Multi-objective optimization of task-

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2025.3569742

to-node assignment in opportunistic fog ran," *Electronics*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/3/474

[16] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 536–25 545, 2022.

[17] J. B. D. da Costa, A. M. de Souza, R. I. Meneguette, E. Cerqueira, D. Rosário, C. Sommer, and L. Villas, "Mobility and deadline-aware task scheduling mechanism for vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 11 345–11 359, 2023.

[18] L. Zhu, X. Li, and H. Zhang, "Resource allocation based on vehicle collaboration quality and delay tolerance in vehicular networks," in *2024 IEEE/CIC International Conference on Communications in China (ICCC)*, 2024, pp. 377–381.

[19] B. Hakim, A. A. Elbery, M. Hefeida, W. S. Alasmary, K. H. Almotairi, and A. Noureldin, "Stc: Spatial and temporal clustering for cooperative perception system," *IEEE Transactions on Vehicular Technology*, pp. 1–11, 2024.

[20] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 953–959.

[21] C. Risma Carletti, F. Raviglione, C. Casetti, F. Stoffella, G. Yilma, and F. Visintainer, "S-ldm: Server local dynamic map for 5g-based centralized enhanced collective perception," *Vehicular Communications*, vol. 49, p. 100819, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209624000949

[22] S. Ucar, T. Higuchi, and O. Altintas, "Platoon as a mobile vehicular cloud," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[23] T. Xiao, C. Chen, Q. Pei, and H. H. Song, "Consortium blockchain-based computation offloading using mobile edge platoon cloud in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 769–17 783, 2022.

[24] M. Nasimi, M. A. Habibi, and H. D. Schotten, "Platoon–assisted vehicular cloud in vanet: Vision and challenges," 2020. [Online]. Available: https://arxiv.org/abs/2008.10928

[25] X. Ma, J. Zhao, and Y. Gong, "Joint scheduling and resource allocation for efficiency-oriented distributed learning over vehicle platooning networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 894–10 908, 2021.

[26] Y. Ma, Q. Liu, J. Fu, K. Liufu, and Q. Li, "Collision-avoidance lane change control method for enhancing safety for connected vehicle platoon in mixed traffic environment," *Accident Analysis & Prevention*, vol. 184, p. 106999, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457523000465

[27] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Opencda:an open cooperative driving automation framework integrated with co-simulation," 2021. [Online]. Available: https://arxiv.org/abs/2107.06260

[28] R. Rajamani, *Vehicle Dynamics and Control*, 01 2006.

[29] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2016.7533003

[30] L. Strand, J. Honer, and A. Knoll, "Modeling inter-vehicle occlusion scenarios in multi-camera traffic surveillance systems," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023, pp. 1–8.

[31] A. A. Al-Habob, O. A. Dobre, A. G. Armada, and S. Muhaidat, "Task scheduling for mobile edge computing using genetic algorithm and conflict graphs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8805–8819, 2020.

[32] W. Rahman and E.-n. Huh, "Content-aware qoe optimization in mec-assisted mobile video streaming," *Multimedia Tools and Applications*, vol. 82, pp. 1–33, 04 2023.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[34] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, 2014, pp. 53–60.

[35] A. Papadakis, L. Mamatas, and S. Petridou, "Investigating the latency dynamics in vehicular platooning networks," in *2024 IEEE International Mediterranean Conference on Communications and Networking (MediterCom)*, 2024, pp. 371–376.

[36] J. Zhou and F. Zhu, "Analytical analysis of the effect of maximum platoon size of connected and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102882, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307828

[37] J. Zeng, Y. Qian, W. Wang, D. Xu, and H. Li, "The impact of connected automated vehicles and platoons on the traffic safety and stability in complex heterogeneous traffic systems," *Physica A: Statistical Mechanics and its Applications*, vol. 629, p. 129195, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437123007501

[38] D. F. Crouse, "On implementing 2d rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[39] Car 2 Car Communication Consortium, "Europe's path to connected, cooperative, and automated mobility," 2022. [Online]. Available: https://www.car-2-car.org/fileadmin/documents/General_Documents/Car_2_Car_Communication_Consortium_-_Europe_s_Path_to_Connected__Cooperative_and_Automated_Mobility_.pdf

[40] B. Häfner, V. Bajpai, J. Ott, and G. Schmitt, "A survey on cooperative architectures and maneuvers for connected and automated vehicles," *IEEE Communications Surveys Tutorials*, vol. 24, pp. 380–403, 2022.

Carlos Mateo Risma Carletti is a Ph.D. student at the Department of Control and Computer Engineering in Politecnico di Torino. He is working on the evaluation of vehicular networking technologies for automated driving and on the development of flexible and open source message encoders and decoders for vehicular messages.He is also dealing with the development and testing of innovative vehicular micro-cloud applications.

Claudio Casetti Claudio Casetti is a Full Professor at the Department of Control and Computer Engineering, Politecnico di Torino, Italy. He has published over 250 papers in peerrefereed international journals and conferences on the following topics: vehicular networks, Intelligent Transportation Systems, 5G/6G networks, IoT systems. According to Google Scholar, his H-index is 42. He is a Senior Member of IEEE and a Senior Editor for Mobile Radio of IEEE Vehicular Technology Magazine.

Jérôme Härri Jérôme Härri (Member, IEEE) is an Associate Professor with the Communication Systems Department, EURECOM, Sophia Antipolis, France, where he leads the Connected Automated Transport System (CATS) Team. He has authored or coauthored over 80 international journal articles and conference papers articles and is involved in various national and European research projects related to connected and automated mobility. His H-index is 31. His research interests are related to wireless vehicular communication and networking, traffic flow modeling, positioning and localization, or control system optimization, in particular their mutual interactions in future automated vehicles. Prof. Härri is a member of the Editorial Board of Sensors (MDPI journal) and a Section Editor of Electronics. He has served as the chair, the co-chair, and a TPC member in a large number of international conferences. He is an Associate Editor of Frontier in Future Transportation and the Journal of Advanced Transportation (Hindawi).

Fulvio Risso Fulvio Risso received the Ph.D. degree in computer engineering from the Politecnico di Torino, Italy, where he is an Associate Professor. His research interests include high-speed and flexible network processing, edge/fog computing, and network functions virtualization.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on September 03,2025 at 09:59:41 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.