

Where to Decide? Centralized vs. Distributed Vehicle Assignment for Platoon Formation

Julian Heinovski *Student Member, IEEE* and Falko Dressler *Fellow, IEEE*

Abstract—Platooning is a promising cooperative driving application for future intelligent transportation systems. In order to assign vehicles to platoons, some algorithm for platoon formation is required. Such vehicle-to-platoon assignments have to be computed on-demand, e.g., when vehicles join or leave the freeways. In order to get best results from platooning, individual properties of involved vehicles have to be considered during the assignment computation. In this paper, we explore the computation of vehicle-to-platoon assignments as an optimization problem based on similarity between vehicles. We define the similarity and, vice versa, the deviation among vehicles based on the desired driving speed of vehicles and their position on the road. We create three approaches to solve this assignment problem: *centralized solver*, *centralized greedy*, and *distributed greedy*, using a Mixed Integer Programming (MIP) solver and greedy heuristics, respectively. Conceptually, the approaches differ in both knowledge about vehicles as well as methodology. We perform a large-scale simulation study using PlaFoSim to compare all approaches. While the *distributed greedy* approach seems to have disadvantages due to the limited local knowledge, it performs as good as the *centralized solver* approach across most metrics. Both outperform the *centralized greedy* approach, which suffers from synchronization and greedy selection effects. The *centralized solver* approach however assumes global knowledge and requires a complex MIP solver to compute vehicle-to-platoon assignments. Overall, the *distributed greedy* approach achieves close to optimal results but requires the least assumptions and complexity. Therefore, we consider the *distributed greedy* approach the best approach among all presented approaches.

Index Terms—Intelligent Transportation Systems, Platoon Formation, Vehicle-to-Platoon Assignment.

I. INTRODUCTION

THE amount of road traffic has been constantly growing in recent years, leading to more and more congestion of the roads and environmental pollution. To cope with these negative effects, vehicle manufacturers and researchers are striving to improve today's driving to be more efficient and comfortable. Vehicles are equipped with more and more technology to assist the driver in his tasks in order to make driving more efficient and safe, transforming them into Intelligent Transportation Systems (ITS). Using Inter-Vehicle Communication (IVC) technologies like 5G-based Cellular Vehicle-to-Everything (C-V2X) communication or Distributed Short-Range Communication (DSRC), vehicles are now able to cooperate with each other. This allows applications such as cooperative driving [1, 2].

One such application is vehicular platooning where multiple vehicles are grouped into a convoy and electronically coupled via Cooperative Adaptive Cruise Control (CACC or Cooperative

Adaptive Cruise Controller) [3–5]. This coupling allows small safety gaps (e.g., 5 m) between platoon members while still enabling string-stable and safe operation. Platooning thus promises to enhance today's driving by improving traffic flow [6]. The technical feasibility of string-stable platooning has been investigated in depth in the literature [3, 5, 7–9]. Most studies typically consider pre-configured and well-defined platoons. This assumption, however, is somewhat unrealistic as some form of bootstrapping a platoon is required: Vehicles will initially drive individually until they encounter an appropriate (existing) platoon to join or another individual vehicle to form a new platoon with. In order to perform the necessary driving tasks (e.g., approaching, lane changing, switching to CACC) to become a platoon member, cooperative maneuvers need to be executed [10, 11]. Deciding which vehicles should form a platoon together is part of the general challenge of assigning vehicles to platoons, which requires computing so called *vehicle-to-platoon assignments*.

While simple ad-hoc approaches enable fast setup of platooning [12], vehicle-to-platoon assignments typically require more complex computation in order to optimize for certain factors [13]. Some first solutions for computing such assignments in an optimal way are limited to, e.g., a certain set of vehicles with known trips and requirements, allowing a priori computations [14]. However, the trips and requirements of vehicles are not always known, which requires on-demand and en route computation of assignments based on the current situation. Thus, new vehicles can only be considered for platooning after entering the freeway and announcing their interest in platooning. Only now, vehicles or other entities can start searching for appropriate platoon candidates and compute corresponding assignments. While there are ideas to wait for other vehicles to form platoons before entering the actual freeway [15], doing so will unnecessarily delay the trips. Therefore, the entire platoon formation process including the computation of vehicle-to-platoon assignments should happen during the vehicles' trips on the freeway. To avoid assigning vehicles to platoons with heterogeneous properties, the individual vehicles properties should be considered during the assignment computation [16, 17]. Thus, the platooning benefits for the individual vehicles can be optimized, without giving up too much of the own requirements, instead of system-level aspects such as traffic flow. Therefore, solving the challenge of vehicle-to-platoon assignments with respect to these aspects is the next important step towards large-scale deployment of platooning [18].

In this paper, inspired by our earlier work [17], we explore the computation of vehicle-to-platoon assignments as an optimization problem based on similarity between vehicles.

Julian Heinovski and Falko Dressler are with the School of Electrical Engineering and Computer Science, TU Berlin, Germany; E-Mails: {heinovski,dressler}@ccs-labs.org).

We define the similarity and, vice versa, the deviation among vehicles based on the desired driving speed of vehicles and their position on the road, thereby considering their individual requirements. We aim to increase vehicles' similarity, thereby minimizing their deviation in desired driving speed and position. We create three approaches to solve this assignment problem: *centralized solver*, *centralized greedy*, and *distributed greedy*, using a Mixed Integer Programming (MIP) solver and greedy heuristics, respectively. Conceptually, the approaches differ in both knowledge about vehicles as well as methodology. We assume that such coordination only requires limited network access, which can easily be provided by modern technologies such as 5G-based C-V2X or IEEE 802.11p-based DSRC. We perform a large-scale simulation study with PlaFoSim to compare both greedy approaches to the optimal solution. We report extensive simulation results to evaluate the impact of the knowledge as well as the performance of the approximation by the heuristics.

Our main contributions can be summarized as follows:

- We explore the problem of vehicle-to-platoon assignments as an optimization problem based on similarity between vehicles.
- We developed three approaches to solve this optimization problem: *centralized solver*, *centralized greedy*, and *distributed greedy*, using a MIP solver and greedy heuristics, respectively.
- We perform a large-scale simulation study with PlaFoSim to compare both greedy approaches to the optimal solution.
- We show that the *distributed greedy* approach leads to the best results while requiring the least assumptions and complexity among all presented approaches.

The remainder of this paper is structured as follows: First, we review related work from the literature in Section II. We then propose our idea of computing vehicle-to-platoon assignments based on similarity between vehicles in Section III. Afterwards, we illustrate our methodology for evaluating the proposed idea and report corresponding results in Section IV. Finally, we summarize and conclude our findings in Section VI.

II. RELATED WORK

A simple approach of forming platoons is to perform spontaneous (ad-hoc) platoon formation with other vehicles, using only limited consideration of vehicles' properties and avoiding complex decision making [12]. Vehicles can join close platoons based on their current position without considering a particular constraint or properties of other vehicles [12]. Adding a little more complexity, vehicles can evaluate whether the estimated benefit of catching up or slowing down and joining a platoon is more fuel-efficient than driving alone [19]. Combining catch-up and slow-down, Saeednia and Menendez [20] propose a hybrid platooning strategy targeting the formation of truck platoons with the highest possible platooning speed. Woo and Skabardonis [21] propose a flow-aware strategy for platoon organization, which performs formation conditionally on the local traffic state (i.e., flow and speed) in order to avoid degrading its performance.

Ad-hoc approaches often form platoons under the assumption that platooning is desired and to improve macroscopic

metrics such as lane capacity and traffic flow. However, when considering a more microscopic level, i.e., metrics that directly influence and are of interest to a driver, this might not be optimal as individual properties and capabilities of the trips and vehicles, i.e., destination, desired driving speed, trip duration, fuel consumption, are not considered. In the following, we separately report on centralized, decentralized, and distributed platoon formation solutions in the literature.

A. Centralized Coordination

In *centralized* coordination, vehicle-to-platoon assignments are computed from a global perspective for all vehicles. Thus, it is assumed that details such as trip information are known to the coordination system.

Many studies consider platooning of trucks, where trips and schedules or deadlines are known beforehand. This allows to use static planning models, where a central coordination instance computes vehicle-to-platoon assignments based on trucks' travel information a priori [14]. Considering trucks from a single fleet, many studies propose approaches for offline optimization of the trucks' fuel consumption by using platooning [22–25]. In addition to the vehicle-to-platoon assignments, the algorithms compute speed profiles, routes, and departure times [26] for the trucks in order to fulfill the desired platoons. Recently, such optimization was proposed also for trucks from multiple fleets with different objectives, utilizing the trucks' computational resources [27]. The complexity of such centralized optimization has been shown to be NP-hard [22].

Since these offline approaches are limited to known sets of vehicles including their properties and trip data, they do not work well for general traffic. Thus, on-demand and en route computation of vehicle-to-platoon assignments based on the current situation on the freeway is required. Liang et al. [28] study fuel-efficient platooning for trucks and form platoons on the fly. They consider an optimization problem for pairwise coordination of vehicles in a centralized manner, where the leading vehicle slows down and the trailing vehicle speeds up. Hoef et al. [29] define a combinatorial optimization problem that uses the transport assignments of trucks for en route formation of truck platoons. The process is repeated whenever assignments change or deviation from the plans are detected.

Considering normal passenger vehicles, Krupitzer et al. [30] propose a centralized platoon coordination system that receives information (i.e., destination) from drivers via Vehicle-to-Everything (V2X) communication, searches for a feasible platoon, and performs the corresponding join maneuver. Liu et al. [31] split the highway into zones of 2 km and propose an optimization problem that decides whether or not each single vehicle should join a specific platoon within each zone.

B. Decentralized Coordination

In *decentralized* coordination, vehicle-to-platoon assignments are computed in multiple locations (e.g., Roadside Units) from limited global perspectives for a subset of vehicles. Thus, decentralized approaches have only limited global knowledge about an area (e.g., a road section) or other vehicles in V2X

communication proximity. Several studies used a decentralized approach that sorts vehicles into platoons at the entrance ramp of the freeway [15, 32]. These approaches group vehicles according to no or limited constraints (e.g., their destination) and let them enter the freeway as an already constructed platoon at a certain time. Like ramp-grouping, platoons can be formed with trucks waiting at hubs along the freeway. Larsen et al. [33] and Johansson et al. [34] propose to use local coordinators at hubs for dispatching multiple trucks at the same time in order to form a platoon. The goal is to optimize fuel savings and minimize the cost of waiting at the hub, while adhering to the waiting time windows of trucks.

Larson et al. [35, 36] deploy a distributed network of local controllers at junctions in a road network. The controllers monitor approaching trucks and coordinate platoon formation among all vehicles in proximity. Using vehicles' information such as speed, position, and destination, an optimization problem for required speed adjustments and expected fuel savings is solved via heuristics. Other studies divide the highway into sections and coordinate platooning within those sections with local centralized controllers. Krupitzer et al. [37] extend their centralized platoon coordination system with sub-systems that individually coordinate vehicles within sections in a regional planner-like approach. While only limited details are provided, their approach uses Vehicle-to-Infrastructure (V2I) communication for exchanging vehicles' desired driving speed and route data with the coordination system, which forms platoons among vehicles with a similar route or destination. Similarly, Burov et al. [38] use local centralized controllers for highway sections of 300 m to coordinate platooning. The controllers collect and transmit data between platoons and vehicles via V2I communication. Going one step further, Zhu et al. [39] use local controllers to provide general driving guidance and platoon coordination for vehicles within a controlling zone (i.e., a freeway section). They create an optimization problem for jointly considering speed, safety, and energy efficiency, in order to maximize overall traffic velocity and minimize collision risk and fuel consumption.

C. Distributed Coordination

In *distributed* coordination, vehicle-to-platoon assignments are computed within the individual vehicles. Here, the vehicle itself is the actor taking the decision about forming a platoon with other vehicles. Thus, distributed approaches have only limited local knowledge about other vehicles in V2X communication proximity. Khan and Bölöni [40] develop a system which evaluates the cost and benefit of forming a platoon with other vehicles in proximity. The system continuously evaluates the utility of platooning in terms of deviation from desired driving speed and fuel consumption; if successful, it indicates the decision to the driver using an LED to adjust the Adaptive Cruise Control (ACC) accordingly. Since vehicles' communication range is limited to 20 m, vehicles continue evaluation after joining a platoon, allowing them to switch to a platoon with higher utility if encountered.

One way of grouping vehicles is based on their destination, thereby maximizing the distance a platoon stays intact and

vehicles can share platoon benefits [41, 42]. Vehicles on entrance ramps of a freeway use V2X to communicate with other vehicles and platoons in range to find feasible platooning opportunities. They select a platoon with members that have similar destinations (within a certain range), thereby increasing lane capacity and traffic throughput [41] as well as fuel consumption [42]. If vehicles cannot immediately find a feasible platoon, they can temporarily join a non feasible one and perform a one-time change to a better fitting platoon later [43]. Also using vehicles' destination and route, Dokur et al. [44] propose a system that exchanges this data with broadcasts via DSRC. If vehicles are heading to the same destination or to different destinations but share a common path, they start negotiations for forming a platoon via messages. A negotiation resolver running on the vehicles settles the negotiations between the vehicles to initiate platooning, also deciding about the platoon leader.

In contrast, many studies use vehicles' speed and position for forming platoons. Su and Ahn [45] propose a distributed algorithm that uses other vehicles' speed and position data from IEEE 802.11p broadcasts. Vehicles calculate the difference between their own value and the average of all neighbors in speed and position. They also choose suitable platoon leaders for new platoons based on an exponential distribution of the speed difference and announce their result using a slotted persistence approach.

D. Open Research Problems

The previous strategies for platoon formation show that optimal groupings substantially improve the performance gain. Many studies are targeted towards trucks using known transport assignments and deadlines. We target individual vehicles, where vehicles' properties are not known beforehand and are quite diverse (e.g., variation in desired driving speed). Thus, vehicle-to-platoon assignments need to be computed on-demand and en route, based on similarity (e.g., in driving speed) among individual vehicles instead of total cost (such as fuel consumption). Since a vehicles' platoon choice can have a huge impact on its platooning benefits, assignments should be optimal, resulting in an NP-hard optimization problem. Existing studies consider many different (sets of) optimization objectives and considered properties, often focusing on system-level aspects instead of vehicles' individual requirements. While some studies perform a comparison of their centralized optimal solution with a centralized heuristic, a proper comparison between centralized and distributed approaches as well as optimal and greedy solutions is still missing from the literature. We summarize the existing studies with on-demand vehicle-to-platoon assignment computation from this section in Table I.

In earlier work [17], we studied en route platoon formation for individual passenger vehicles based on their similarity in desired driving speed and position on the road, thereby optimizing vehicle-to-platoon assignments regarding individual properties. Besides formally describing the platoon formation problem, we introduced both a centralized and a distributed greedy heuristic for solving the NP-hard assignment problem in an efficient but simple way. Simulations already indicated

Table I
CATEGORIZATION OF RELATED WORK WITH ON-DEMAND VEHICLE-TO-PLATOON ASSIGNMENT COMPUTATION.

Related Work	assignment time		vehicle knowledge		coordination		solution		considered properties
	before trip	en route	global	local	centralized	distributed	optimal	heuristic	
Liang et al. [28]		x	x		x		x		route, deadline
Hoef et al. [29]		x	x		x		x		route, deadline
Krupitzer et al. [30]		x	x		x				destination
Liu et al. [31]		x	x		x		x		speed, destination
Hall and Chin [15]	x			x	x			x	destination
Baskar et al. [32]	x			x	x			x	destination
Larsen et al. [33]	x			x	x			x	route, deadline
Johansson et al. [34]	x			x	x			x	route, deadline
Larson et al. [35, 36]		x		x	x			x	speed, position, destination
Krupitzer et al. [37]		x	x		x			x	route, speed
Burov et al. [38]		x		x	x		x		speed, position, platoon size
Zhu et al. [39]		x		x	x		x		speed, fuel, safety
Khan and Bölöni [40]		x		x		x		x	speed, fuel
Dao et al. [41, 42]	x			x		x		x	destination
Hobert [43]		x		x		x		x	destination
Sharma et al. [46]		x		x		x			route
Dokur et al. [44]		x		x		x		x	route, destination
Su and Ahn [45]		x		x		x		x	speed, position
our earlier work [17]		x	x	x	x	x		x	speed, position
this work	(possible)	x	x	x	x	x	x	x	speed, position, preferred speed window

that the choice of the approach, and the willingness to deviate from individual objectives, have a huge impact on the resulting assignments. However, the absolute performance of the approaches in comparison to an optimal solution for all vehicles was still unclear.

In this paper, we go one step further and add an approach to optimally solve our similarity-based optimization problem using a MIP solver. For this, we refined the entire problem formulation to make it more applicable for centralized optimization. We compare both greedy approaches to the optimal solution in detail using a wide range of performance metrics in a large-scale simulation study.

III. COMPUTING VEHICLE-TO-PLATOON ASSIGNMENTS

In this section, we define the problem of computing vehicle-to-platoon assignments, which has to be solved as part of the platoon formation process. We describe our perspective on this problem as well as our approach for solving the problem using the concept of similarity among vehicles, which we base on their individual properties.

A. Assumptions

We focus on individual traffic, where vehicles have different properties (such as driving speed) and start their trips unsynchronized.¹

Thus, the trips and requirements of vehicles are not known beforehand, which requires dynamic on-demand computation of assignments based on the current situation. Vehicles will initially drive individually until they encounter an appropriate (existing) platoon to join or an individual vehicle to form a new platoon with. We generally assume that driving in a platoon is preferred because of the expected benefits such as driving efficiency as well as increased safety and traffic throughput.

¹We assume drivers can freely customize their trip based on an individually preferred traveling speed during trip planning. This can be compared with choosing a route from multiple options such as *fast* or *economic* that are provided by the navigational systems already today.

When vehicles start their trip (outside of the freeway), they have no knowledge about other already existing vehicles and vice versa. We assume that the entire platoon formation process including the computation of vehicle-to-platoon assignments happens during the vehicles' trips on the freeway (en route). After assignment, vehicles create a new platoon or join an existing one by performing a maneuver. Vehicles can communicate with each other by means of 5G-based C-V2X or DSRC to exchange platooning information and maneuver control. As platoon coordination has low requirements on the communication, we neglect this part in the evaluation to make the approach also technology agnostic. After successful competition of the join maneuver, vehicles stay within the platoon until they reach their destination, at which a leave maneuver is performed. Assuming a fully operational Cooperative Adaptive Cruise Controller, vehicles in a platoon always mirror the behavior of the platoon leader and keep a constant gap of 5 m [9, 47].

B. Problem Formulation

Early approaches to find candidate vehicles to construct platoons consider different constraints and optimization goals, such as grouping by destination or route, or (pairwise) by fuel efficiency. To avoid assigning vehicles to platoons with heterogeneous properties, we aim to consider the individual vehicles' properties during the assignment computation. Thus, the platooning benefits for the individual vehicles can be optimized without giving up too much of the own requirements instead of system-level aspects such as traffic flow. Assigning vehicles to platoons based on their properties is like clustering vehicles according to some similarity metric corresponding to the constraints and goals introduced by a formation strategy.

Inspired by our earlier work [17], we are using the desired driving speed as a primary similarity metric. Additionally, we also take the position of the vehicles on the freeway into account. Since joining a platoon which is far away can add a lot of overhead in fuel-consumption, it not useful to consider such cases as possible options.

In order to come up with a formation strategy, we formalize the problem as follows. By considering individual vehicles as platoons of size 1, we can model both similarly by the set

$$\{n, D, p, l\}, \quad (1)$$

where n is a unique identifier, D is the desired driving speed, p is the current position, and l is the current position of the last vehicle in the platoon or $l := p$ if vehicle n is driving individually. The position of a vehicle is defined as the location of its front bumper on the freeway. In case of a platoon, we consider its lead vehicle (leader) as the platoon's representative. Thus, the leader's values are used for n , D , and p .

We define C as the set of all individual vehicles that need to be assigned to a platoon. Additionally, we define T as the set of all already existing platoons (consisting of multiple vehicles) as well as individual vehicles, representing all potential assignment targets. Thus, all vehicles in C are also included in T . We define I and J as index sets of C and T , thus $c_i \in C$ and $t_j \in T$ define the i -th and j -th element of C and T , respectively.

We define A as a $|C| \times |T|$ matrix that contains a decision variable a_{ij} for every possible vehicle-to-platoon assignment between elements of C and T . The decision variable is used to determine whether an arbitrary $c_i \in C$ is assigned to an arbitrary $t_j \in T$, using the definition of a_{ij} as

$$a_{ij} = \begin{cases} 1, & \text{if } c_i \text{ is assigned to } t_j \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

If a searching vehicle c_i cannot be assigned to any other vehicle or platoon t_j (with $c_i \neq t_j$), it will need to keep driving individually. We model such a case by $a_{ij} = 1$ with $c_i = t_j$ and call the assignment a *self-assignment* of vehicle c_i . Thereby, all searching vehicles c_i within the assignment matrix A have an assignment. It is important to note that A in general is not symmetric and not all assignments of c_i and t_j are technically possible (see below).

We also define F as a $|C| \times |T|$ matrix that contains the similarity (or rather the deviation) $f_{ij} = f(c_i, t_j)$ in their properties between elements of C and T , using the definition of $f(c, t)$ as

$$f(c, t) = \alpha \cdot d_s(c, t) + (1 - \alpha) \cdot d_p(c, t), \quad (3)$$

where, $\alpha \in [0, 1]$ is a weighting coefficient. The deviation between c_i and t_j in desired speed $d_s(c_i, t_j)$ as well as in position on the freeway $d_p(c_i, t_j)$ are defined as

$$d_s(c, t) = \frac{\|D_c - D_t\|}{m \cdot D_c}, \quad m \in [0, 1], \quad (4)$$

$$d_p(c, t) = \frac{\min(\|p_c - p_t\|, \|l_t - p_c\|)}{r}, \quad r \in \mathbb{N}, \quad (5)$$

where m is the maximum allowed deviation from the desired driving speed of vehicle c and r is the maximum allowed deviation from the position on the freeway of vehicle c (i.e., the search range) for vehicles being considered as potential candidates. Equations (4) and (5) both calculate a deviation relative to the respective maximum value (i.e., m and r), thereby defining a window of allowed deviation in $[0, 1]$. While Equation (4) simply uses the desired driving speed of c and t ,

Equation (5) considers the location of vehicle or platoon t in relation to vehicle c . We assign the largest possible deviation value (i.e., 1.0) for modeling the similarity of a vehicle to itself ($c_i = t_j$). Thereby, we allow a self-assignment, but clearly favor assignments to other vehicles or platoons.

Typical approaches for calculating user similarity include the Jaccard coefficient, the cosine similarity, and the multi-dimensional Euclidean distance [48]. These, however, cannot be applied to our problem as they either work only on sets (Jaccard) or do not support defining a maximum deviation as well as importance per property (cosine and Euclidean). Thus, our model uses a one-dimensional Euclidean distance that is normalized based on a maximum allowed deviation (a deviation window) and weighted using a coefficient per property. Equation (3) can flexibly be extended with arbitrary other properties while keeping a total value in $[0, 1]$.

The previous equations are subject to the following constraints:

$$d_s(c, t) \leq 1.0, \quad (6)$$

$$d_p(c, t) \leq 1.0, \quad (7)$$

$$p_c \leq l_t, \quad (8)$$

where p_c is the position of vehicle c and l_t is the position of the last vehicle in the platoon t ($l_t = p_c$ if t is driving individually). It is important to mention that Equation (8) requires that a vehicle or a platoon t is in front of the searching vehicle c in order to be considered as a potential candidate. Thereby, without loss of generality, we currently only allow joining at the back of a vehicle or platoon, which requires less coordination [43, p. 98] and does not influence other vehicles as well as already existing platoon members as much (assuming only the joining vehicle adjusts its driving speed in order to approach the platoon). "If an existing platoon [had] to slow down to allow a following single vehicle to join (faster), there would be significant wasted energy as multiple vehicles [had] to undergo braking and subsequent acceleration." [31, p. 7574]. Considering also vehicles in the front can increase the probability of finding a candidate and will potentially influence the role the vehicles will have in the resulting platoon (leader vs. follower). For conciseness, we do not consider those effects in this work. It is important to note that, like A , also F in general is not symmetric.

Using the previous definitions, we can describe the objective of our optimization problem as

$$\text{minimize } \sum_{i \in I} \sum_{j \in J} a_{ij} f_{ij}, \quad (9)$$

subject to the following constraints

$$\forall i \in I : \sum_{j \in J} a_{ij} = 1, \quad (10)$$

$$\forall j \in J : \sum_{i \in I} a_{ij} \leq 1, \quad (11)$$

$$\begin{aligned} & \forall i \in I : a_{ij} = 1 \wedge c_i = t_j, \\ & \text{if } \exists k \in I : a_{kl} = 1 \wedge c_i = t_l \wedge c_k \neq t_j. \end{aligned} \quad (12)$$

To summarize, we try to find the best fitting platoon candidate (i.e., the candidate with the biggest similarity, or, in other

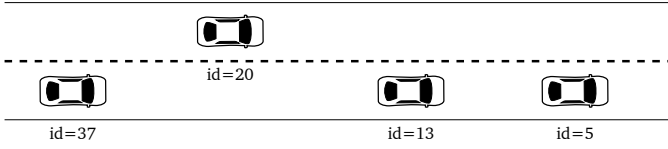


Figure 1. Example scenario: Four vehicles are driving individually on an arbitrary road with two lanes (e.g., a freeway) and try to find a platoon.

words, with the smallest deviation) t_j for each vehicle c_i to form a platoon with. Equation (10) assures that vehicle c_i is assigned to exactly one target vehicle or platoon t_j . Here, a self-assignment indicates that the vehicle continues to drive individually. Equation (11) assures that at most one vehicle is assigned to every target vehicle or platoon t_j . This is related to the join maneuver that needs to be executed in order to successfully implement an assignment. Equation (12) assures that vehicle c_i is not assigned to any vehicle or platoon t_j (i.e., a self-assignment) if another vehicle c_k ($\neq t_j$) is already assigned to it ($t_l = c_i$).

After assignment ($a_{ij} = 1$), vehicle c_i performs a join maneuver towards its designated target platoon or vehicle t_j . Assuming a successful join maneuver, vehicle c_i became part of a platoon with vehicle t_j (or the platoon t_j was already part of). Once vehicles become platoon members, they stay in the platoon until they reach their destination.

C. Example Scenario

As an example for the optimization problem, consider the scenario depicted in Figure 1, where four vehicles are driving individually on an arbitrary road with two lanes (e.g., a freeway) and now try to find a platoon. The vehicles in the example are defined by their set of properties, $\{5, 121 \text{ km/h}, 430 \text{ m}, 430 \text{ m}\}$, $\{13, 89 \text{ km/h}, 270 \text{ m}, 270 \text{ m}\}$, $\{20, 107 \text{ km/h}, 250 \text{ m}, 250 \text{ m}\}$, $\{37, 93 \text{ km/h}, 70 \text{ m}, 70 \text{ m}\}$, and the algorithm in this case uses the following set of parameters:

$$\alpha = 0.6, m = 0.4, r = 400 \text{ m} .$$

By using these properties and parameters, the list of possible platoon candidates and their corresponding deviation $f(c, t)$ can be calculated as

$$\begin{aligned} f(13, 5) &= 0.6 \cdot 0.599 + 0.4 \cdot 0.4 = 0.519 \\ f(20, 5) &= 0.6 \cdot 0.218 + 0.4 \cdot 0.45 = 0.31 \\ f(20, 13) &= 0.6 \cdot 0.28 + 0.4 \cdot 0.05 = 0.188 \\ f(37, 5) &= 0.6 \cdot 0.501 + 0.4 \cdot 0.9 = 0.66 \\ f(37, 13) &= 0.6 \cdot 0.071 + 0.4 \cdot 0.5 = 0.242 \\ f(37, 20) &= 0.6 \cdot 0.25 + 0.4 \cdot 0.45 = 0.33 . \end{aligned} \quad (13)$$

It is important to note that we skipped all entries for self-assignments in this list. By also considering self-assignments, all vehicles have at least one platoon candidate to *join* with a deviation value of ≤ 1.0 . In particular 5 can only be assigned to itself.

Algorithm 1 Creation of decision variables for the solver in the *centralized solver* approach

Input: list of all vehicles in the scenario
for all vehicles c_i in the scenario; **do**
 if c_i in platoon **or** in maneuver **then**
 next; $\triangleright c_i$ is (currently) not available
 add constraint: only one assignment of c_i ;
 for all vehicles / platoons t_j in the scenario **do**
 if (t_j in platoon **and** **not** platoon leader) **or** t_j in maneuver **then**
 next; $\triangleright t_j$ is (currently) not available
 if $p_{c_i} > l_{t_j}$ **then**
 next; \triangleright do not consider vehicles / platoons behind
 if $d_s(c_i, t_j) > 1.0$ **or** $d_p(c_i, t_j) > 1.0$ **then**
 next; \triangleright too large deviation in speed or position
 if $c_i = t_j$ **then**
 $f_{ij} = 1.0$; \triangleright use large deviation for self-assignment
 else
 calculate f_{ij} based on d_s and d_p ;
 add decision variable and deviation $\{a_{ij}, f_{ij}\}$;
Output: all decision variables and deviations: list($\{a_{ij}, f_{ij}\}$)

From the list of possible candidates and their corresponding deviations, the (optimal) solution minimizing the overall deviation is

$$\begin{aligned} f(5, 5) &= 1.0 \\ f(37, 13) &= 0.242 \\ f(20, 5) &= 0.31 \end{aligned}$$

with a total deviation of $1.0 + 0.242 + 0.31 = 1.552$. Here, selecting a candidate pair blocks both involved vehicles, making them unavailable for further selection. Since a vehicle can only be in one maneuver at a time, at most two assignments can be executed in parallel. After these maneuvers are finished, the vehicles in the scenario are grouped into two platoons: $\{13, 37\}$ and $\{5, 20\}$.

D. Solution approach: “centralized solver”

We aim to minimize the overall deviation from the desired metrics while assigning as many vehicles to platoons as possible. Thus, in this approach, the optimization problem is solved periodically for every vehicle in the scenario at the same time. A central server has global knowledge about all vehicles and their corresponding properties, which is collected by means of an infrastructure-based network such as 5G-based C-V2X. Using the vehicles’ information, a mathematical solver can compute an optimal solution for all vehicles. The complexity of centralized optimization in general has been shown to be NP-hard [22], which makes a real-world deployment difficult, but it can provide an upper bound for possible solutions.

For all possible vehicle-to-platoon assignments between $c_i \in C$ and $t_j \in T$, the decision variable a_{ij} (see Equation (2)) needs to be created. This also includes vehicles that cannot be assigned (at the time being). We show the process of creating the decision variables in Algorithm 1. Note that not all combinations of c_i and t_j are technically possible (see

Algorithm 2 Addition of constraints for the solver in the *centralized solver* approach

Input: all decision variables and deviations: $\text{list}(\{a_{ij}, f_{ij}\})$
for all (unique) vehicles / platoons t_j in all entries in list
do
 for all entries in list with target t_j and vehicle $c_i \neq t_j$
 do
 add constraint: at most one assignment to t_j ;
 add constraint: at most one assignment of c_i ;
 for all entries in list with vehicle t_j and target $c_i \neq t_j$
 do
 add constraint: at most one assignment of t_j ;

Algorithm 3 Collection of possible candidates in the *centralized greedy* approach

Input: all vehicles in the scenario
for all vehicles c_i in the scenario; **do**
 if c_i in platoon **or** c_i in maneuver **then**
 next; $\triangleright c_i$ is (currently) not available
 for all vehicles / platoons t_j in the scenario with $t_j \neq c_i$
 do
 if (t_j in platoon **and** t_j **not** platoon leader) **or** t_j in maneuver **then**
 next; $\triangleright t_j$ is (currently) not available
 if $p_{c_i} > l_{t_j}$ **then**
 next; \triangleright do not consider vehicles / platoons behind
 if $d_s(c_i, t_j) > 1.0$ **or** $d_p(c_i, t_j) > 1.0$ **then**
 next; \triangleright too large deviation in speed or position
 add $\{c_i, t_j, f(c_i, t_j)\}$ to list of possible assignments;
Output: all possible assignments: $\text{list}(\{c_i, t_j, f(c_i, t_j)\})$

Equations (6) to (8)). In such cases, no decision variable a_{ij} is created. Besides the list of decision variables and corresponding deviations, the solver needs some further constraints in order to compute a useful solution (see Equations (10) to (12)). We depict the process of adding the constraints to the model for the solver in Algorithm 2. After the solver has computed a solution, it has to be applied to the vehicles. The solution is stored within the decision variables of the model, which are used to trigger corresponding join maneuvers among involved vehicles. Due to the centralized coordination, all assignments are synchronized, and maneuver conflicts are thereby avoided. In case of a self-assignment, no maneuver is triggered, and the vehicle stays individual.

E. Solution approach: “centralized greedy”

In order to make a real-world deployment of a centralized solution for the optimization problem more feasible, we aim to reduce its computational complexity. Thus, we propose the *centralized greedy* approach, which consists of greedy heuristics following the idea of the *centralized solver* approach: it computes vehicle-to-platoon assignments for every vehicle in the scenario at the same time using full knowledge about all vehicles.

In this approach, we first calculate the deviation $f(c_i, t_j)$ for all vehicles in the neighborhood which do not violate the

Algorithm 4 Selection of best candidate for assignment in the *centralized greedy* approach

Input: all possible assignments: $\text{list}(\{c_i, t_j, f(c_i, t_j)\})$
for all (unique) vehicles c_i in the list **do**
 $s \leftarrow \{t_j | t_j \in \text{list}(\{c_i, t_j, f_i(x)\})\}$; \triangleright all possible t_j
 if $\|s\| > 0$ **then**
 $b \leftarrow \min f(c_i, t_j), t_j \in s$; \triangleright select best candidate
 let c_i join b ;
 remove list entries with c_i or b ; \triangleright apply constraints
Output: vehicles performing a join maneuver

constraints given in Equations (6) to (8) and add an entry for them to a list of possible assignments, using Algorithm 3. An entry $\{c_i, t_j, f(c_i, t_j)\}$ in this list contains vehicle $c_i \in C$ and vehicle or platoon $t_j \in T$ as well as the total deviation of vehicle or platoon t_j from vehicle c_i (see Equation (3)). Note that the deviation of t_j from c_i is not symmetric as the opposite direction produces a different value.

Once all possible assignments and the corresponding deviations are computed, we use Algorithm 4 to select the best target t_j for every searching vehicle c_i from this list and trigger a corresponding join maneuver. In particular, we select the t_j with the smallest deviation $f(c_i, t_j)$ and remove all entries which contain vehicles c_i or t_j because they became unavailable for further selection. This heuristic is greedy as it makes decisions solely from the perspective of a searching vehicle c_i without considering consequences for other searching vehicles, which could join the same vehicle or platoon t_j . Thus, it does not consider all possible assignments for all vehicles during the decision making. Due to its nested for-loop, the computational complexity of this approach is $O(n^2)$.

Looking again at the example scenario from Section III-B, we observe that the *centralized greedy* heuristic also produces two platoons, but it does not necessarily compute the aforementioned optimal solution. Instead, without loss of generality, it will select

$$\begin{aligned} f(5, 5) &= 1.0 \\ f(13, 5) &= 0.519 \\ f(37, 20) &= 0.33 \end{aligned}$$

with a total deviation of $1.0+0.519+0.33 = 1.849$ by following the given order of possible assignments from Equation (13) (i.e., increasing by id). Here, selecting a candidate pair blocks both involved vehicles, making them unavailable for further selection. Since a vehicle can only be in one maneuver at a time, at most two assignments can be executed in parallel. Due to the centralized coordination, all assignments are synchronized, and maneuver conflicts are thereby avoided. After these maneuvers are finished, the vehicles in the scenario are grouped into two platoons: $\{5, 13\}$ and $\{20, 37\}$.

F. Solution approach: “distributed greedy”

Solving the optimization problem at a centralized entity for all vehicles using global knowledge about all vehicles has advantages. In contrast, however, it has limited scalability, introduces a single point of failure, and does not necessarily

Algorithm 5 Collection of possible candidates in the *distributed greedy* approach

Input: list of (available) vehicles / platoons t_j in communication range of vehicle c_i (i.e., Ω_i)

for all vehicles / platoons t_j in the neighborhood Ω_i **do**

if $p_{c_i} > l_{t_j}$ **then**

 next; ▷ *do not consider vehicles / platoons behind*

if $d_s(c_i, t_j) > 1.0$ **or** $d_p(c_i, t_j) > 1.0$ **then**

 next; ▷ *too large deviation in speed or position*

 add $\{c_i, t_j, f(c_i, t_j)\}$ to list of possible assignments;

Output: list of possible assignments: $\text{list}(\{c_i, t_j, f(c_i, t_j)\})$

present the best design considering the management of dynamic and fast-paced entities like vehicles [49]. Thus, in order to remove the dependency on a central entity, we propose the *distributed greedy* approach, which consists of greedy heuristics as well but works fully distributed: every vehicle $c_i \in C$ tries to compute a vehicle-to-platoon assignment to a vehicle or platoon in its neighborhood $\Omega_i \subset T$ individually. In contrast to both centralized approaches, vehicles in the *distributed greedy* case only have local knowledge of the scenario: they know other vehicles within a certain (communication) range, assuming some general neighbor management, e.g., from regular DSRC beacons [17], and can directly access other vehicles' information (i.e., speed, position, platoon state, and maneuver state). In all approaches, we assume that the information is always up to date, thus, vehicles which are not applicable anymore cannot be used as platooning opportunity.

Using the information about nearby vehicles, the heuristic given in Algorithm 5 is executed to prepare the list of possible assignments. Note that the deviation of t_j from c_i is not symmetric as the opposite direction produces a different value. A heuristic like Algorithm 4 is used to select a candidate vehicle or platoon t_j with the smallest deviation to join. Conceptually, the same assignments as in the *centralized greedy* approach are selected. However, the selection of possible candidates is limited to the restricted nature of the local knowledge and, therefore, depends on the time the heuristic is evaluated. Also, the vehicles are not synchronized and thus perform the assignment process asynchronously, leading to assignments and maneuvers not being triggered at the same time. The computational complexity of this approach is $O(n)$.

Looking again at the example scenario from Section III-B, we observe that the *distributed greedy* heuristic will trigger join attempts for the following subset of Equation (13):

$$\begin{aligned} f(13, 5) &= 0.519 \\ f(20, 13) &= 0.188 \\ f(37, 20) &= 0.33 \end{aligned}$$

The outcome of these join attempts depends on the order of execution defined by the trigger time at the involved vehicles (i.e., 13, 20 and 37). Since a vehicle can only be in one maneuver at a time, at most two assignments can be executed in parallel. If both 13 and 37 trigger their attempts roughly at the same time but before 20, two platoons will be formed: $\{5, 13\}$ and $\{20, 37\}$. Otherwise, only the first triggered attempt



Figure 2. Screenshot from the SUMO-based live GUI of PlaFoSim, showing a small region of our simulation scenario: 3 lane freeway, 5 individually driving vehicles, and 1 formed platoon consisting of 3 members. All members of one platoon are shown in the same color, which makes it easy to recognize them.

(e.g., from 20) will lead to a successful platoon and the others will be aborted due to unavailability of the involved vehicle(s) (e.g., 20).

IV. EVALUATION

Within this work, we evaluate the proposed platoon formation approaches in an extensive simulation study. We do so by comparing them to each other and to two baseline approaches without platooning, using typical metrics for traffic modeling as well as platooning. In the evaluation, we first report details of our three formation algorithms to show the impact of knowledge (local vs. global). We then discuss results from a macroscopic, system-level perspective of the scenario to show the impact on the general traffic system. Finally, we report results for typical platooning metrics from a microscopic, vehicle-level perspective of the individual vehicles.

A. Methodology

To observe effects of platooning and platoon formation algorithms as well as their impact, large-scale simulation studies with large scenarios and many vehicles are required. Many studies use Plexe [50] for this purpose. While being perfectly suited for fine-grained simulation of platoon controllers including the necessary wireless communication link, the approach suffers from limited scalability. Therefore, we use PlaFoSim² in our study as we are more interested in large-scale effects of platoon management rather than microscopic control of the involved vehicles [51]. Besides validating its underlying mobility models in a comparison to SUMO and Plexe, we ran the experiments from our earlier work [17] to validate the algorithm behavior using PlaFoSim, producing qualitatively equal results. We were able to increase the scenario in size and simulation time as well as in the number of parameter configurations. We use PlaFoSim version 0.15.4 with some additional changes, which we will also integrate into its upstream version.³ A screenshot from the SUMO-based live GUI of PlaFoSim is shown in Figure 2.

In our simulation study, we consider a 3-lane freeway (see Figure 2) of 100 km length with periodic on-/off-ramps every 10 km, which allow vehicles to enter and leave the freeway. Vehicles perform trips of 50 km between a pair of randomly selected on-/off ramps. We assume a road network without any disturbances to the road infrastructure (e.g., by road construction) or by traffic accidents.

The desired driving speed is sampled from a normal distribution with a mean of 120 km/h (33 m/s), a variation

²<https://www.plafosim.de>

³<https://github.com/heinovski/plafosim>

Table II
SIMULATION PARAMETERS FOR ROAD AND TRAFFIC

Parameter	Value
Freeway length	100 km
Number of lanes	3
Ramp interval	10 km
Depart positions	random on-ramp
Arrival positions	random off-ramp at trip end
Fixed trip length	50 km
Desired speed	\mathcal{N} (120 km/h, 10%)
Min. desired speed	80 km/h
Max. desired speed	160 km/h
Target density	5, 10, 15, 20 and 25 veh/km/lane
Target no. of vehicles	1500, 3000, 4500, 6000 and 7500 veh
Departure rate	3564, 7129, 10693, 14257 and 17822 veh/h
Car Following model	Krauss, ACC, and CACC
Krauss desired headway	1 s
ACC desired headway	1 s
CACC desired gap	5 m
Max. speed v_{\max}	200 km/h
Max. acceleration	2.5 m/s ²
Max. deceleration	10 m/s ² ⁴
Vehicle length	5 m
Min. gap between veh.	2.5 m

of 10%, and limited to values from [80 km/h, 160 km/h].⁵ Vehicles start their trips driving individually, using either the popular Krauss model [52] or a standard ACC (cf. [53, Eq. 6.18]). They keep a constant time-based safety gap order to avoid collisions. Platoon members use a standard CACC with constant spacing (cf. [47, Eq. 6]) after platoon formation is completed.

In our study, we compare our three proposed platoon formation algorithms with two baseline approaches without platooning:

- *human* – human driving (following the Krauss model)
- *ACC* – vehicles controlled by a standard ACC
- *distributed greedy* – vehicle platooning (CACC) using our distributed (greedy) approach (see Section III-F) for platoon formation
- *centralized greedy* – vehicle platooning (CACC) using our centralized (greedy) approach (see Section III-E) for platoon formation
- *centralized solver* – vehicle platooning (CACC) using our optimal approach (see Section III-D) for platoon formation

Thus, we can determine the improvement of vehicle platooning (CACC) in comparison to already existing but non-platooning driver assistance systems.

We model vehicle demand in a macroscopic way via a flow with constant insertion (i.e., new vehicles are continuously being inserted based on a departure rate) to roughly keep a given target density of vehicles on the road in order to achieve a stable traffic situation. We do so because we are only interested in a relative comparison between non-platooning and platooning

⁵The parameters roughly correspond to a typical German Autobahn with a recommended driving speed of 130 km/h and allow us to account for the desired driving speed (and limit) of trucks (80 km/h) as well as faster driving vehicles (160 km/h). We observed only few vehicles (0.4%) that were assigned a desired driving speed equal to the artificial limits of our distribution.

Table III
SIMULATION PARAMETERS FOR FORMATION LOGIC

Parameter	Value
Penetration rate	100%
Execution interval	60 s
Communication range for <i>distributed greedy</i>	500 m
Max. deviation from desired speed m	0.1–0.3, step 0.1
Max. deviation in position r	1000 m
Weight of speed deviation α	0.5
Time limit for MIP solver	600 s

approaches and not in the maximum possible traffic flow for every approach. The corresponding departure rate for a given target density is calculated as

$$t_{\text{expected}} = \frac{d_{\text{trip}}}{v_{\text{desired}}} \quad (14)$$

$$P_{\text{arrival}} = \frac{1}{t_{\text{expected}}} \quad (15)$$

$$R_{\text{departure}} = [P_{\text{arrival}} \cdot D_{\text{desired}} \cdot L \cdot d_{\text{road}} \cdot 3600] \quad (16)$$

where t_{expected} is the expected travel time of the vehicles, given their fixed trip length and their desired driving speed (in m/s), P_{arrival} is the probability of vehicle arrivals per second, $R_{\text{departure}}$ is the corresponding departure rate, given the target density D_{desired} , number of lanes L , and the length of the road d_{road} . A summary of all simulation parameters for the road network and traffic can be found in Table II. We chose 25 veh/km/lane as highest target vehicle density as initial simulations showed that at higher densities the scenario is too crowded to insert further vehicles in all simulated approaches.

Platoon formation is performed in regular intervals every 60 s. Forming platoons always consist of the following steps: (1) data collection of available vehicles and platoons, (2) computation of vehicle-to-platoon assignments depending on the selected approach, (3) execution of join maneuvers to implement the computed assignment(s). As the focus of this work is on the assignment process, platooning mobility, wireless communication, and platooning maneuvers are implemented in an abstracted way [51]. A summary of all simulation parameters for the platoon formation can be found in Table III. In order to consider deviation in desired speed and position equally, we set $\alpha = 0.5$ within this study. Further, we set the maximal allowed distance between vehicles $r = 1000$ m to investigate the benefit of the centralized approaches, which are not limited by the communication range (500 m). At the same time, it avoids assignments to far away platoon candidates that cannot be reached within a reasonable time. In general, however, vehicles can be assigned to (and join) any other vehicle or platoon which fulfills the constraints given in Section III-B.

The *centralized solver* approach is implemented by using a MIP solver from Google's OR-Tools library.⁶ The constraints for the assignment defined in Section III-D are modeled by row constraints and a integer decision variable with a coefficient corresponding to the respective deviation is added for every possible vehicle-to-platoon assignment. Since self-assignments

⁶OR-Tools version 9.0, <https://developers.google.com/optimization/>

are inherently allowed but not desired, a deviation value of 1.0 (the maximum) is used for these cases. To achieve a finite but useful execution time, we limit the solver’s execution time to 600 s. While the solver is running, the simulation time does not advance, and its state does not change. Thus, the solver’s execution time does not impact the vehicles’ travel time and corresponding statistics.

During join maneuvers, vehicles are moved to their designated platoon position, considering the approximate time for approaching under perfect conditions. For real-world deployments, solutions for coping with other vehicles interfering the maneuvers have been proposed [11, 54, 55]. We emulate a safe but lengthy join procedure without actual simulation of the required steps, thereby reducing simulation complexity and time, while also avoiding conflicts among vehicles. It is important to mention that we currently only allow joining at the rear of a vehicle or platoon and thus only vehicles in front are considered as potential platooning candidates (cf. Section III-B). After successful competition of the join maneuver, vehicles stay within the platoon until they reach their destination ramp, at which a leave maneuver is performed. If the platoon leader leaves the platoon, the next remaining vehicle within the formation becomes the new leader, keeping all properties of the platoon. If all other platoon members leave the platoon, the vehicle continues to drive individually and starts searching for new platooning opportunities.

We simulate 7200 s (2 h) of traffic in multiple repetitions for every configuration and approach. We first pre-fill the road network with the desired number of vehicles before the simulation starts. To cut off the initial transient period, we use the first 1800 s (0.5 h) of the simulation time as a warm-up period and ignore all results in this interval, only considering vehicles that departed after this period. Using all approaches and parameter combinations defined in Tables II and III, we simulate a total of 165 individual runs. Using an AMD Ryzen 9 5900X 12-Core processor @3.7GHz, we observe the shortest simulation time (50 min wall-clock time) with the smallest configuration (speed window 0.1, density 5 veh/km/lane) for the ACC approach. In contrast, we observe the longest simulation time (4 h wall-clock time) with the largest configuration (speed window 0.3, density 25 veh/km/lane) using the *centralized solver* approach. In the following, we omit showing confidence intervals for any reported average value, but we made sure that they are reasonably small.

B. Validation

We performed an extensive validation of the implemented solutions. As an example, we show the total number of vehicles in platoons for an exemplary simulation run over its entire simulation time in Figure 3. Starting with 0 vehicles at 0 s, the total number of vehicles grows over time until it reaches a almost constant value of ≈ 8000 . Slowly, platoons are formed and we can see that platoons sized of up to 14 vehicles are created. After the warm-up period of 1800 s (dashed line), the total number of vehicles in the simulation as well the distribution of platoon sizes stays almost constant. Other approaches and configurations show similar simulation

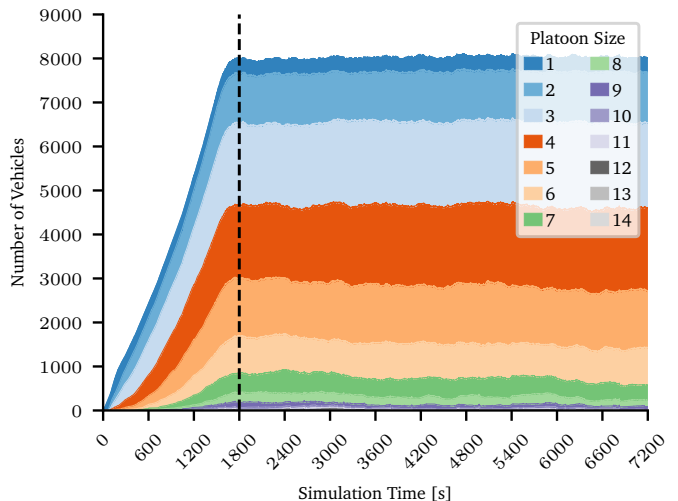


Figure 3. Total number of vehicles in platoons over the entire simulation time for the *distributed greedy* approach using the largest configuration (speed window 0.3, density 25 veh/km/lane). The platoon size is indicated by color and the warm-up period by the dashed line.

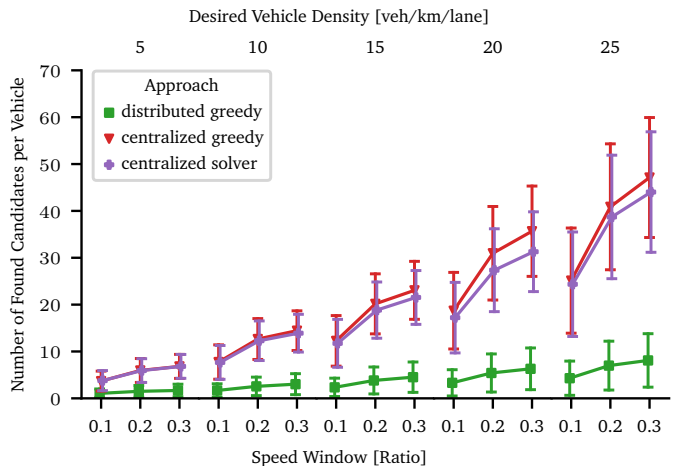


Figure 4. Average number of potential platooning candidates known to the platoon formation algorithm in every iteration (with standard deviation) per vehicle and for all platooning approaches. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario.

behavior, but the total number of vehicles in the simulation as well as the distribution of platoon sizes are slightly different as we will report in the following.

C. Available Candidates

The first step in the process of forming platoons is to find available platooning opportunities (candidates). These can be either individual vehicles or already existing platoons.

We show the results of the found candidates metric (i.e., the average number of potential platooning opportunities known to the platoon formation algorithm) in Figure 4. The found candidates metric counts the number of possible assignments for platoon formation as identified for a single vehicle.

A fundamental difference between the approaches is the available knowledge about other vehicles. While vehicles using

the *distributed greedy* approach only have local knowledge of other vehicles in their vicinity limited by a certain communication range, both centralized approaches have global knowledge of all vehicles within the entire scenario. Obviously, the more vehicles are known to the respective approach, the more vehicles can be used within the formation algorithm, increasing the probability to find a possible candidate. Of course, non-platooning approaches such as *human* and *ACC* do not have any knowledge of other vehicles since no V2X communication is used. Thus, these approaches are not shown here.

As expected, both centralized approaches similarly find a lot more potential candidates to use for the platoon formation than the *distributed greedy* approach due to their overall knowledge of the scenario. In fact, the *centralized greedy* approach finds a few more candidates than the *centralized solver* approach, starting at medium densities, because the *centralized solver* is able to fit more vehicles to platoons, thus, less are available for the next assignment rounds.

The number of found candidates for all approaches is also impacted by the speed window (i.e., the allowed speed deviation) as well as the desired vehicle density in the scenario. Larger speed deviation thresholds lead to a less strict definition of similarity (in desired driving speed), which leads to more potential platooning candidates being found. Similarly, higher desired vehicle densities lead to more vehicles in the overall scenario and thus more platooning opportunities being available in general. This effect is more prominent for both centralized approaches as their global knowledge of the scenario allows for a significant increase in candidates with a bigger density.

The variance in values (we show the standard deviation) for all approaches is related to the desired driving speed (distribution). Vehicles with a lower desired driving speed also have a smaller absolute (in m/s) speed window and thus less candidates. Further, vehicles with a desired driving speed from the outer area of the normal distribution (e.g., ≤ 90 km/h and ≥ 150 km/h) that we use to assign the speed, will have less candidates due to the limiting of the possible values, which are also less frequent. In contrast, vehicles with a desired driving speed of a frequent value (e.g., 120 km/h) have a lot of candidates. As expected, a larger speed window and a higher desired vehicle density increase the variance as well. This effect is pronounced by the global knowledge of both centralized approaches.

Even though the general pattern of the found candidates metric is in line with our expectations, the actual numbers are smaller than expected. Especially both centralized approaches should find tremendously more candidates due to their global knowledge about all vehicles (cf. vehicle densities). The reason for this is that some vehicles are not available (anymore) to be considered for vehicle-to-platoon assignments. They are either already in a platoon (i.e., a follower) or currently in the process of forming a platoon (i.e., in a maneuver). We count these vehicles within the filtered candidates metric for all vehicles and in all approaches, similarly to the found candidates metric. Vehicles that are not available because of their position, a too large deviation, or the limited communication range with the *distributed greedy* approach are not counted as filtered.

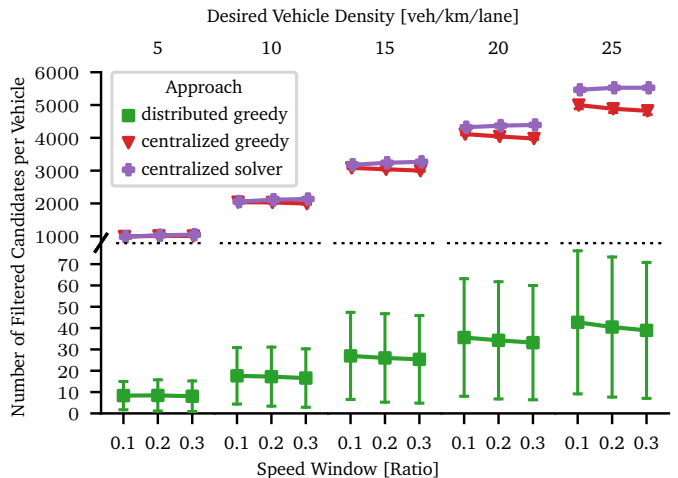


Figure 5. Average number of filtered platooning candidates in every iteration (with standard deviation) per vehicle and for all platooning approaches. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario. The y-axis is split and does not show values between 80 and 800.

The results are shown in Figure 5. Due to the global knowledge and decision making for all vehicles in the scenario, both centralized approaches need to filter almost all vehicles due to their platooning status. The *distributed greedy* approach needs to filter tremendously less candidates (i.e., two orders of magnitude) due to its limited local knowledge of other vehicles. In the *centralized solver* approach, more vehicles are already in a platoon and thus filtered in comparison to the *centralized greedy* approach due to the synchronized and balancing (leader vs. follower) decision making for all vehicles in the scenario. For all approaches, the sum of the found and filtered candidates metrics defines a lower bound for the number of vehicles initially known to the formation algorithm. Naturally, filtering all known vehicles that are not available (anymore) or have a too large deviation decreases the amount of effectively available candidates (the found candidates metric).

D. Formation Process

After analyzing the number of platooning opportunities, we now look at the actual formation process.

1) *Time to Platoon*: Figure 6 shows the average time required to become a platoon member per vehicle for all platooning approaches. This time is measured from the departure of a vehicle until a join maneuver is completed successfully. It includes waiting for (successful) execution(s) of the formation algorithm to compute a vehicle-to-platoon assignment as well as the corresponding join process itself, which in general was marginal. Similarly, this time is measured if a vehicle became a leader of a newly formed platoon. The actual runtime of the formation algorithm itself is not included as we did not model it for both greedy heuristics (i.e., *distributed greedy* and *centralized greedy*) and report the time for solving the optimization problem in the *centralized solver* approach separately in Section IV-G.

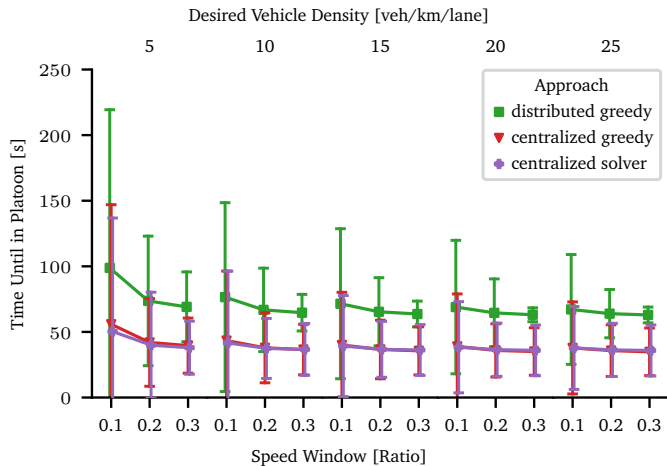


Figure 6. Average time (with standard deviation) required to become a platoon member per vehicle for all platooning approaches. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario.

We immediately observe that the *distributed greedy* approach experiences the longest time for all configurations. This is expected, as this approach has only limited local knowledge about available platooning opportunities. Thus, it requires more algorithm executions until a (possible) platooning opportunity can be found (and successfully joined). This effect becomes less prominent for bigger speed windows and larger vehicle densities. Both centralized approaches require less time in general and differ only marginally. This is expected, as both are using global knowledge about the vehicles and can thus compute successful assignments already at the first execution of the algorithm. The standard deviations due to the variance in the number of available candidates (see Figure 4) as well as the duration between departure and execution of the assignment process.

2) *Platoon Size*: Another typical metric in the context of platooning is the platoon size and its distribution. Unfortunately, the interpretation of the platoon size is difficult and requires specific context of the underlying design goals of an algorithm. In general, however, we aim at a low number of vehicles that are not in a platoon.

We report the distribution of the platoon size over various desired traffic densities in Figure 7. First, we observe that all platooning approaches produce platoons (platoon size > 1) and only a few vehicles do not find a platoon. While, the *centralized greedy* approach builds slightly smaller platoons than *centralized solver*, the *distributed greedy* approach builds slightly larger ones (max. sizes: 10, 11 and 16). For all approaches, most of the vehicles are in platoons with a size of 2–5.

E. System-level Metrics (Macroscopic)

After reporting details of the formation process itself, we now look at traffic-related metrics from a broader, macroscopic perspective.

1) *Traffic Flow*: Since platooning is a cooperative driving application, the performance of our algorithms and the platooning itself is influenced by the general traffic in the scenario. All approaches will have an impact on the resulting traffic, and so does the general configuration of our simulation scenario (e.g., the departure rate). In order to get an overview of the general traffic behavior in the scenario, Figure 8 shows the actual departure rate (Figure 8a) as well as the resulting vehicle density (Figure 8b) observed from the simulation for all of our simulated approaches and target vehicle densities. Since both metrics are directly related to each other, the observed effects correlate as well.

The observed values for departure rate and vehicle density increase with higher desired densities. This is consistent as we construct our departure rate from the desired vehicle density using Equation (16). With human and ACC driven vehicles, it is possible to meet the desired departure rates and vehicle densities at low and medium desired vehicle densities. Starting at a desired density of 20 veh/km/lane, however, the target cannot be met anymore. Here, the crowded traffic makes it difficult to insert new vehicles as there is only little remaining capacity (free space) on the road. Although the platooning approaches also suffer from the same effect, they can fulfill the desired departure rates and vehicle densities for all traffic configurations and produce a smaller deviation from the desired vehicle density.

2) *Driving Speed*: When looking at the average driving speed of all vehicles in the scenario (see Figure 9), we can prominently observe the effect of high traffic. As expected, the driving speed is reduced for all approaches when the scenario becomes more crowded due to natural traffic effects. Here, vehicles need to reduce their driving speed in order to be able to fulfill their time-based safety gap. The effect is generally worse for non-platooning approaches due to larger safety gaps between vehicles, resulting in stop-and-go traffic [56]. For all platooning approaches, the speed is like the smallest speed window as options are limited. But, a larger speed window allows more deviation and leads to an overall higher speed in both greedy approaches. In comparison, the *centralized solver* approach minimizes the deviation in speed and position for all vehicles simultaneously and, thus, leads to a slightly reduced driving speed. Since larger windows allow more vehicles to become platoon leaders (instead of platoon followers), and a platoon is always driving at the speed of its leader, the deviation (in driving speed) stays small for more vehicles. The *centralized solver* approach also keeps a slightly decreasing trend over windows, being more like the original objective of vehicles.

The trend for the platooning approaches remains similar for a desired vehicle density of up to 15 veh/km/lane. After that, the overall driving speed in the scenario decreases more noticeably for both greedy approaches than for the optimal approach; At the largest density, despite global knowledge, the *centralized greedy* approach performs the worst due to synchronization and greedy selection effects.

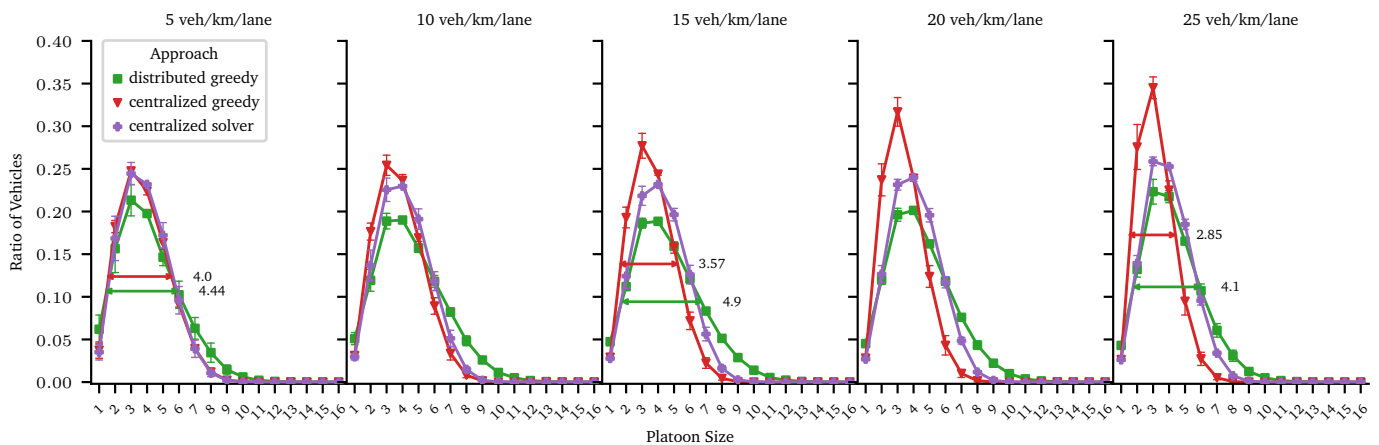


Figure 7. Distribution of platoon sizes depicted by the number of vehicles in platoons with their respective size for all platooning approaches. The facets show various values for the desired vehicle density in the scenario. The horizontal arrows and corresponding numbers indicate selected full width at half maximum values of the distribution with the same color.

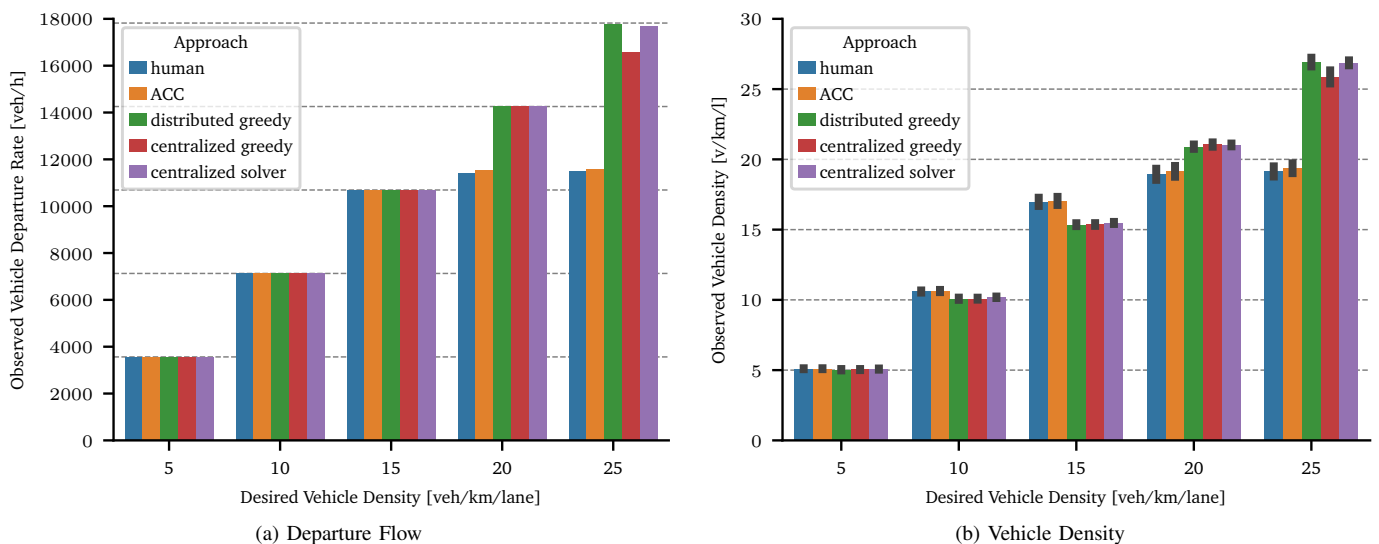


Figure 8. General traffic behavior observed from the simulation (with standard deviations) over all simulated approaches. The x-axis shows various values for the desired vehicle density in the scenario.

F. Vehicle-level Metrics (Microscopic)

We now analyze results for platooning-related effects from a microscopic (i.e., the individual vehicles') perspective.

1) *Deviation From Desired Driving Speed:* Vehicles are inherently required to deviate from their desired driving speed in order to form platoons with other vehicles, which likely drive at different speeds. The main goal of our formation algorithms is to minimize the vehicles' deviation from their own desired driving speed. Figure 10 shows the average relative deviation from the desired driving speed per vehicle for all simulated approaches. The platooning approaches have both a negative and a positive deviation, which is due to the compromise that vehicles need to make when forming platoons (i.e., they are adjusting their driving speed to the platoon leader). All platooning approaches perform more or less similarly and their positive deviation (driving faster than desired) mostly stays within the corresponding defined speed window (see Figure 10a).

Naturally, the value of the speed window impacts the actual deviation from the desired driving speed. As an example, we show results for a selected desired vehicle density of 5 veh/km/lane in Figure 10a. As expected, both greedy heuristics lead to a larger spread in deviation values with the *distributed greedy* approach having the largest. This is due to the smaller number of platooning opportunities in this approach, which requires vehicles to join platoons with large deviations in order to fulfill the goal of platooning. Increasing the speed window allows for even more deviation, thus also increasing the spread of values. While the values for the *centralized solver* approach are mostly symmetrically located around 0, both greedy heuristics tend to deviate more positively.

Independent of platooning, the amount of vehicles in the scenario has an impact on the deviation from the desired driving speed (see Figure 10b). When comparing the smallest and the largest simulated densities (5 and 25 veh/km/lane), the negative deviation increases from nearly 0% to -13% on

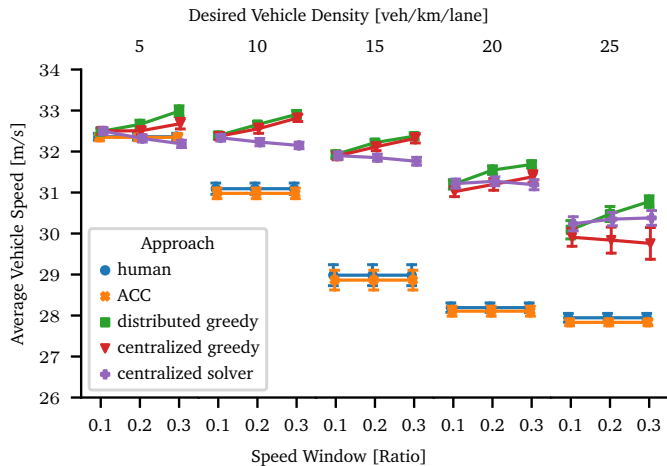


Figure 9. Average driving speed (with standard deviation) of all vehicles in the simulation for all simulated approaches. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario.

average for the non-platooning approaches. This is expected as the freeway becomes more crowded with larger densities and the vehicles need to adjust their driving speed based on the decreasing gaps to other vehicles in order to avoid crashes. The platooning approaches cope better with a crowded scenario. Their negative deviation reaches only -5 , -6 and -9% on average at the largest desired vehicle density for the *distributed greedy*, the *centralized solver*, and the *centralized greedy* approach, respectively. In fact, more vehicles even lead to a slightly lower positive deviation as more platooning opportunities become available, and vehicles have to make less compromises.

In order to study the effects of the platooning approaches in more detail, we show the ratio of vehicles deviating from the allowed speed window in Figure 11. For low densities, i.e., a small amount of available platooning opportunities, only few vehicles deviate from their desired driving speed for all approaches. When increasing the vehicle density, the deviation increases and a difference between approaches becomes visible. As expected, the *centralized solver* approach has the lowest deviation as it tries to minimize the deviation for all vehicles in an optimal way. With the largest speed window (0.3), all approaches perform much better because enough platooning opportunities are available to them.

2) *Travel Time*: Naturally, the previous effects on the driving speed can be seen when looking at vehicles' trip duration as their trip lengths are equal. When starting the trip, every vehicle estimates the time it is going to travel to its destination, assuming a constant speed at the desired value. Upon arrival, vehicles also record their real travel time and calculate the travel time ratio (i.e., the ratio of expected to real travel time). We use this metric to show the impact of the traffic and the use of platooning in general on vehicles' trip duration, which is a frequently used metric when assessing platooning systems. We show the travel time ratio for all vehicles in Figure 12. The general tendency follows our expectations: the ratio increases with vehicle density due to slower driving (cf. Figure 9). A

ratio > 1.0 indicates that vehicles generally need longer for their trips than expected. Both non-platooning approaches (i.e., *human* and *ACC*) share 1.0 as minimum value as their trips can only get influenced negatively by delays due to traffic. This effect is increased with higher vehicle densities as more vehicles are more impacted by the traffic. Similarly, the traffic also impacts all three platooning approaches and vehicles' trips are delayed (the ratio is larger than 1.0 on average). When platooning is enabled, deviations from the initially expected travel time in both directions can be observed. Since vehicles will adjust their driving speed to the target platoon (based on the configured speed window of the formation algorithm), their speed and the resulting travel time can be faster than expected.

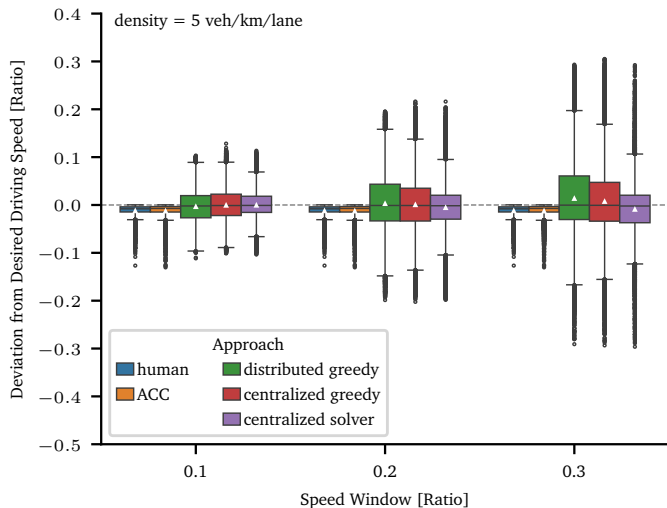
3) *Fuel Consumption*: A major argument for platooning is the reduced air drag between vehicles and the resulting reduced pollutants emission and fuel consumption. Similar results hold for electrified vehicles, we fall back to gas consumption simply because of the availability of validated and widely accepted consumption models. For modeling general vehicle emissions, we use the Handbook Emission Factors for Road Transport (HBEFA)⁷ version 3.1, following the approach implemented in SUMO. In particular, we currently use the *PC_G_EU4* emission class, which represents a gasoline driven passenger car with an engine corresponding to the European norm version 4. For vehicles that are driving in a platoon, we calculate a reduction in emissions and fuel consumption based on their position within the platoon [57], applying the approach from our earlier work [17].

We show the fuel consumption in Figure 13. The values plotted represent the fuel consumption of all vehicles in liters per 100 km, which is computed by using the total fuel consumption and the trip distance of the vehicles. Following the effects shown in Figure 9, a higher target vehicle density in general reduces the fuel consumption due to the slower driving speed with more traffic in the scenario. For human driving, starting at 20 veh/km/lane, the traffic transforms into stop-and-go, which leads to a lot of acceleration & deceleration maneuvers increasing the average value as well as the spread of the fuel consumption. Synchronized driving using *ACC* already helps reducing fuel consumption, in fact resulting in the lowest values, but also suffers from slower driving speed in general (cf. Figure 9). All platooning approaches result in similar values, but with increasing densities, platooning seemingly performs slightly worse than *ACC*. This is due to the higher average speed that is achieved with platooning.

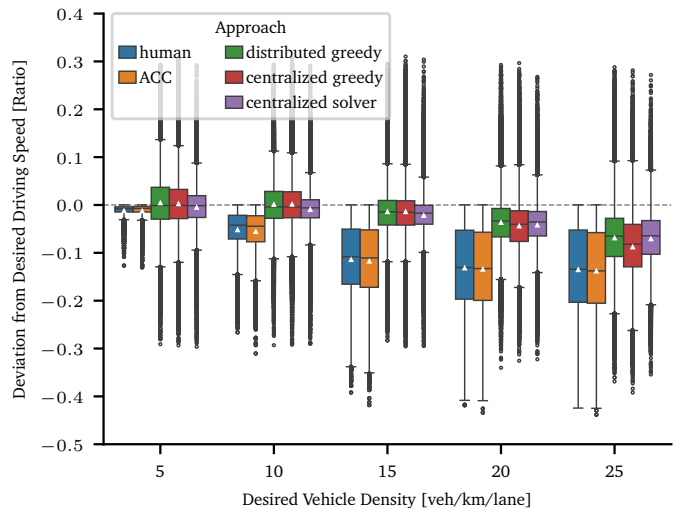
G. Performance of the MIP Solver

We finally look at the performance of the MIP solver for the *centralized solver* approach. Figure 14 shows the average runtime (with standard deviation) of all executions the solver. This parameter measures the time that the solver uses to solve the optimization problem after all decision variables and constraints are defined (see Algorithms 1 and 2). As expected, the runtime increases with a larger speed window and a higher vehicle density as the number of vehicles that the solver needs to consider as source and destination increases.

⁷<https://www.hbefa.net/e/index.html>



(a) Results for various speed windows (parameter m) for a selected desired vehicle density of 5 veh/km/lane. The results for *human* and *ACC* are independent of the speed window.



(b) Results for various desired vehicle densities for all simulated speed windows (parameter m).

Figure 10. Average relative deviation (ratio) from the desired driving speed per vehicle for all simulated approaches. The triangle within a box depicts the mean value.

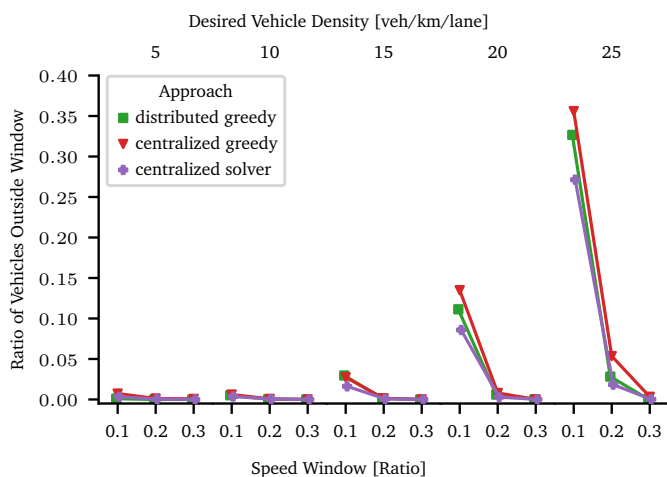


Figure 11. Ratio of vehicles deviating from the allowed speed window for all platooning approaches. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario.

It should be noted that in all configurations, the solution quality reported by the MIP solver was well above 99.99% in all iterations.

V. DISCUSSION

Obviously, both centralized approaches (*centralized solver* and *centralized greedy*) can rely on more knowledge about the scenario compared to the *distributed greedy* approach. They are aware of many more vehicles and, thus, more platooning opportunities, even though many vehicles are filtered due their current platooning and maneuver status. In the *centralized greedy* approach, many initially available candidates are naturally removed due to the greedy selection in the assignment process, leading to only few platooning opportunities in reality.

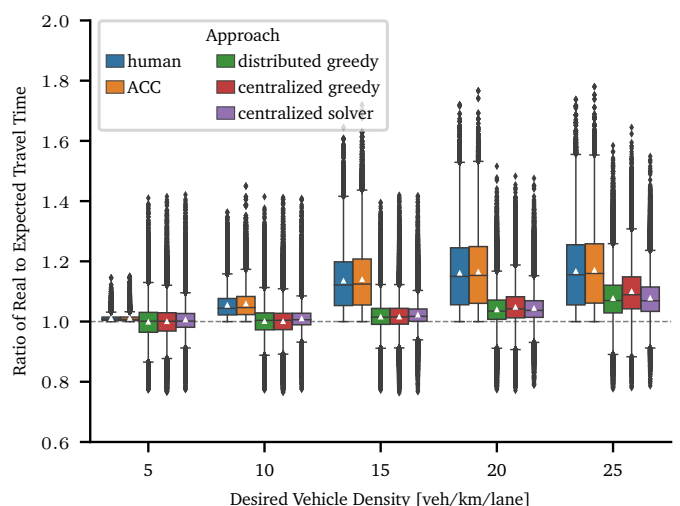


Figure 12. Ratio of expected to real travel time for all simulated approaches. The x-axis shows various values for the desired vehicle density in the scenario. The triangle within a box depicts the mean value. A value smaller than 1.0 indicates that the vehicle reached its destination faster than expected, a value larger than 1.0 in contrast indicates that the vehicle was slower than expected.

More vehicles in the scenario and a more relaxed speed window lead to more available candidates (i.e., platooning opportunities) in all approaches. More platooning opportunities lead to more assignments to close-by vehicles and platoons since the similarity function in Equation (3) also considers distance between vehicles.

The more vehicles the MIP solver of the *centralized solver* approach has to consider, the more complex the decision making becomes. Thus, the time required for computing a solution increases while the quality of the solution decreases, given a fixed time limit for the computation. Still, the quality of the computed solution reported by the MIP solver was well above

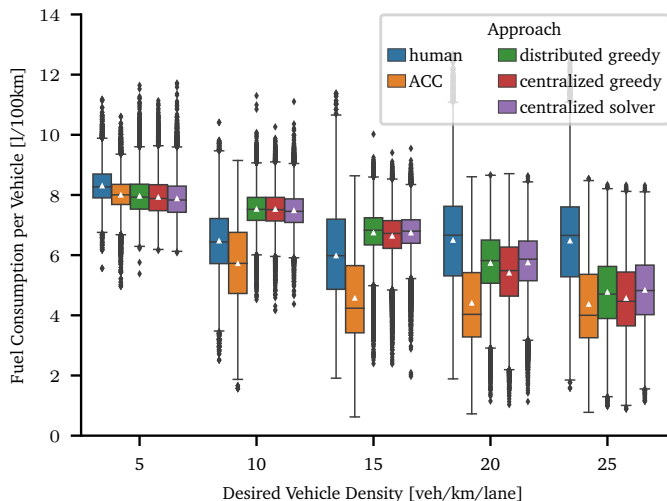


Figure 13. Fuel consumption per vehicle for all simulated approaches. The x-axis shows various values for the desired vehicle density in the scenario. The triangle within a box depicts the mean value.

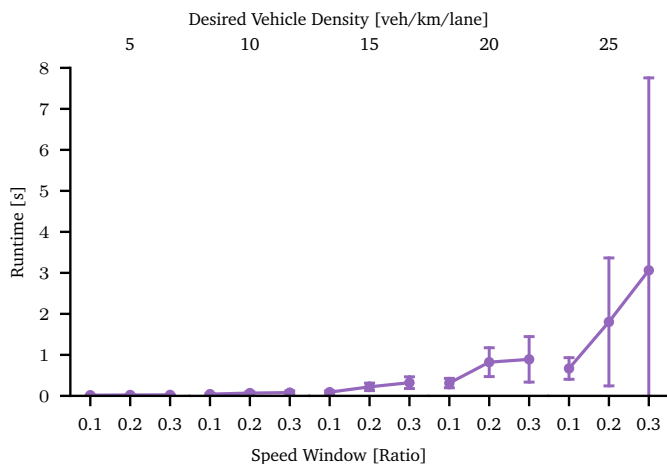


Figure 14. Average runtime (with standard deviation) of all executions of the MIP solver in the *centralized solver* approach. The x-axis shows various values of the speed window (parameter m) for our platoon formation algorithms. The facets show various values for the desired vehicle density in the scenario.

99.9% in all cases, even with the largest number of vehicles and, thus, possible vehicle-to-platoon assignments that the solver has to consider. The *centralized solver* approach, however, assumes global knowledge and requires a complex MIP solver to compute vehicle-to-platoon assignments. In comparison, both greedy approaches have a lower computational complexity (only $O(n)$ for the *distributed greedy* approach), which improves scalability for real-world deployments. Additionally, the *distributed greedy* approach requires only local knowledge, which eliminates the communication bottleneck towards a single (edge) server.

In all approaches, some time is needed to find a suitable platoon. This is particularly true for the *distributed greedy* approach because limited local knowledge allows for less platooning opportunities. The difference between the approaches becomes smaller with higher vehicle densities and larger allowed speed deviation. Having only one central entity to

perform the formation tasks in both centralized approaches works well for small scenarios or freeway sections, but does not scale to handle larger freeway networks (e.g., country-wide).

Obviously, the traffic density influences the driving speed and can lead to stop-and-go traffic in crowded scenarios [56]. All of our platooning approaches help improving the traffic flow by synchronizing driving. Due to more vehicles in platoons, the *centralized solver* and *distributed greedy* approaches perform slightly better than the *centralized greedy* approach.

Considering our main optimization goal, reducing the deviation from the individual desired driving speed, we observed a correlation to the traffic density: The higher the traffic, the more the deviation from the desired driving speed for all approaches. Platooning generally helps to reduce the deviation. Here, *centralized greedy* performs slightly worse. Adding a speed window for limiting the maximum allowed deviation, leads to reduced actual deviation due to platooning but can lead to more impact of the traffic on the driving: 20% allowed speed deviation already helps for more than 90% of the vehicles to stay within their window.

We also looked at the fuel consumption, which is directly impacted by the traffic as well as the resulting driving speed. A crowded scenario initially leads to less fuel due to the slower driving speed, but stop-and-go traffic increases the consumption. Platooning helps by synchronizing the traffic and reducing the air drag of vehicles, thus saving fuel. We see that *centralized solver* and *distributed greedy* lead to slightly higher fuel consumption due to higher driving speed.

The deviation from the desired driving speed directly impacts the travel time. While more traffic leads to longer trip duration, platooning can help to reduce this effect. Again, the *centralized greedy* approach performs slightly worse.

From all experiments we conclude that an allowed speed deviation of 20% is the best compromise between platooning effects and individual properties for all approaches. All approaches benefit from denser traffic producing more platooning opportunities. The limited local knowledge of the *distributed greedy* approach does not depict an issue anymore with medium densities.

VI. CONCLUSION

In this paper, we introduced and explored three approaches for vehicle-to-platoon assignments: *centralized solver*, *centralized greedy*, and *distributed greedy*, using a MIP solver and greedy heuristics, respectively. Conceptually, the approaches differ in both knowledge about other vehicles as well as the used methodology. We first define a similarity goal and, vice versa, the deviation among vehicles, thereby considering their individual requirements. The aim is to increase vehicles' similarity, thus, minimizing their deviation in desired driving speed and position. Our results show that all presented platooning approaches help in comparison to *human* and *ACC* driving. However, the selection of the formation algorithm is important, as it significantly influences the platoon assignment. Our simulations show that the *centralized solver* approach performs best in terms of individual platooning benefits but is conservative in terms of deviating from individual objectives. In contrast, both

presented greedy heuristics are less conservative and lead to a larger deviation from individual objectives. We also see that the willingness to compromise can pay off as more vehicles can benefit from platooning. While the *distributed greedy* approach has some disadvantages due to the limited local knowledge, it performs as good as the *centralized solver* approach in most metrics. Both outperform the *centralized greedy* approach, which suffers from synchronization and greedy selection effects. The *centralized solver* approach however assumes global knowledge and requires a complex MIP solver to compute vehicle-to-platoon assignments. Overall, the *distributed greedy* approach achieves close to optimal results but requires the least assumptions and complexity. Therefore, we consider the *distributed greedy* approach the best approach among all presented approaches.

In future work, we want to investigate how a decentralized approach can increase vehicle knowledge while reducing synchronization effects at the same time. Also, we plan to consider the heterogeneity of vehicles by using more of their properties such as their destination and vehicle capabilities within the similarity function. Lastly, we want to consider more sophisticated maneuvers that allow merging of platoons and algorithms that allow to react to changes in properties by re-assigning vehicles.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Max Schettler, Dominik S. Buse, Agon Memedi, and Doğanalp Ergenç for their support and feedback during the creation of this work.

REFERENCES

- [1] F. Dressler, F. Klingler, M. Segata, and R. Lo Cigno, "Cooperative Driving and the Tactile Internet," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 436–446, Feb. 2019.
- [2] R. Lo Cigno and M. Segata, "Cooperative driving: A comprehensive perspective, the role of communications, and its potential development," *Elsevier Computer Communications*, vol. 193, pp. 82–93, Sep. 2022.
- [3] S. E. Shladover et al., "Automated Vehicle Control Developments in the PATH Program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, Feb. 1991.
- [4] C. Bergenheim et al., "Overview of Platooning Systems," in *19th World Congress on Intelligent Transport Systems (ITS 2012)*, Vienna, Austria: TRB, Oct. 2012.
- [5] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A Survey on Platoon-Based Vehicular Cyber-Physical Systems," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 263–284, 2016.
- [6] B. van Arem, C. J. G. van Driel, and R. Visser, "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, Dec. 2006.
- [7] R. Horowitz and P. Varaiya, "Control Design of an Automated Highway System," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 913–925, Jul. 2000.
- [8] T. Robinson, E. Chan, and E. Coelingh, "Operating Platoons On Public Motorways: An Introduction To The SARTRE Platooning Programme," in *17th World Congress on Intelligent Transport Systems (ITS 2010)*, Busan, South Korea, Oct. 2010.
- [9] P. S. Jootel, "SAfe Road TRains for the Environment," SARTRE Project, Final Project Report, Oct. 2012.
- [10] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*, Wollongong, Australia: IEEE, Jul. 2013, pp. 1152–1157.
- [11] M. Segata, B. Bloessl, S. Joerer, F. Dressler, and R. Lo Cigno, "Supporting Platooning Maneuvers through IVC: An Initial Protocol Analysis for the Join Maneuver," in *11th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2014)*, Obergurgl, Austria: IEEE, Apr. 2014, pp. 130–137.
- [12] S. Maiti, S. Winter, L. Kulik, and S. Sarkar, "Analysis of an Ad-hoc Platoon Formation and Dissolution Strategy on a Multi-lane Highway," in *15th International Conference on GeoComputation (GeoComputation 2019)*, Queenstown, New Zealand: University of Auckland, Sep. 2019.
- [13] T. Sturm, C. Krupitzer, M. Segata, and C. Becker, "A Taxonomy of Optimization Factors for Platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, 2021.
- [14] A. K. Bhoopalam, N. Agatz, and R. Zuidwijk, "Planning of truck platoons: A literature review and directions for future research," *Elsevier Transportation Research Part B: Methodological*, vol. 107, pp. 212–228, Jan. 2018.
- [15] R. Hall and C. Chin, "Vehicle sorting for platoon formation: Impacts on highway entry and throughput," *Elsevier Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 405–420, Dec. 2005.
- [16] V. Lesch et al., "An Overview on Approaches for Coordination of Platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 049–10 065, Aug. 2022.
- [17] J. Heinovski and F. Dressler, "Platoon Formation: Optimized Car to Platoon Assignment Strategies and Protocols," in *10th IEEE Vehicular Networking Conference (VNC 2018)*, Taipei, Taiwan: IEEE, Dec. 2018.
- [18] J. Hou et al., "Large-Scale Vehicle Platooning: Advances and Challenges in Scheduling and Planning Techniques," *Elsevier Engineering*, vol. 28, pp. 26–48, Sep. 2023.
- [19] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "When is it Fuel Efficient for a Heavy Duty Vehicle to Catch Up With a Platoon?" In *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, Japan: Elsevier, Sep. 2013, pp. 738–743.
- [20] M. Saeednia and M. Menendez, "Analysis of Strategies for Truck Platooning: Hybrid Strategy," *TRB Transportation Research Record*, vol. 2547, no. 1, pp. 41–48, Jan. 2016.
- [21] S. Woo and A. Skabardonis, "Flow-aware platoon formation of Connected Automated Vehicles in a mixed traffic with human-driven vehicles," *Elsevier Transportation Research Part C: Emerging Technologies*, vol. 133, Dec. 2021.
- [22] E. Larsson, G. Sennton, and J. Larson, "The vehicle platooning problem: Computational complexity and heuristics," *Elsevier Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 258–277, Nov. 2015.
- [23] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, "Fuel-Optimal Centralized Coordination of Truck Platooning Based on Shortest Paths," in *American Nuclear Conference (ACC 2015)*, Chicago, IL: IEEE, Jul. 2015, pp. 3740–3745.
- [24] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, "Coordinating Truck Platooning by Clustering Pairwise Fuel-Optimal Plans," in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain: IEEE, Sep. 2015, pp. 408–415.
- [25] A. Nourmohammadzadeh and S. Hartmann, "Fuel Efficient Truck Platooning with Time Restrictions and Multiple Speeds Solved by a Particle Swarm Optimisation," in *Theory and Practice of Natural Computing*, ser. Lecture Notes in Computer Science, 1st ed., vol. 11324, Springer, 2018, pp. 188–200.
- [26] V. Sokolov, J. Larson, T. Munson, J. Auld, and D. Karbowski, "Maximization of Platoon Formation Through Centralized Routing and Departure Time Coordination," *TRB Transportation Research Record*, vol. 2667, no. 1, pp. 10–16, Jan. 2017.
- [27] Y. Zeng, M. Wang, and R. T. Rajan, "Decentralized coordination for truck platooning," *Wiley Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 15, pp. 1997–2015, Aug. 2022.
- [28] K. Y. Liang, J. Mårtensson, and K. H. Johansson, "Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1051–1061, Apr. 2016.
- [29] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, "Fuel-Efficient En Route Formation of Truck Platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 102–112, Jan. 2018.
- [30] C. Krupitzer et al., "RoCoSys: A Framework for Coordination of Mobile IoT Devices," in *1st International Workshop on Mobile and Pervasive Internet of Things (PerIoT 2017)*, Big Island, HI: IEEE, Mar. 2017, pp. 485–490.
- [31] D. Liu, B. Eksioğlu, M. J. Schmid, N. Huynh, and G. Comert, "Optimizing Energy Savings for a Fleet of Commercial Autonomous

- Trucks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7570–7586, Jul. 2022.
- [32] L. D. Baskar, B. D. Schutter, and H. Hellendoorn, “Dynamic Speed Limits and On-Ramp Metering for IVHS using Model Predictive Control,” in *11th IEEE International Conference on Intelligent Transportation Systems (ITSC 2008)*, Beijing, China: IEEE, Oct. 2008, pp. 821–826.
- [33] R. Larsen, J. Rich, and T. K. Rasmussen, “Hub-based truck platooning: Potentials and profitability,” *Elsevier Transportation Research Part E: Logistics and Transportation Review*, vol. 127, pp. 249–264, Jul. 2019.
- [34] A. Johansson, V. Turri, E. Nekouei, K. H. Johansson, and J. Mårtensson, “Truck Platoon Formation at Hubs: An Optimal Release Time Rule,” *Elsevier IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 312–15 318, 2020.
- [35] J. Larson, C. Kammer, K.-Y. Liang, and K. H. Johansson, “Coordinated route optimization for heavy-duty vehicle platoons,” in *16th IEEE International Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands: IEEE, Oct. 2013.
- [36] J. Larson, K. Y. Liang, and K. H. Johansson, “A Distributed Framework for Coordinated Heavy-Duty Vehicle Platooning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 419–429, Feb. 2015.
- [37] C. Krupitzer et al., “Towards Infrastructure-Aided Self-Organized Hybrid Platooning,” in *IEEE Global Conference on Internet of Things (GCIoT 2018)*, Alexandria, Egypt: IEEE, Dec. 2018.
- [38] M. Burov, N. Mehr, S. Smith, A. Kurzhanskiy, and M. Arcak, “Platoon Formation Algorithm for Minimizing Travel Time,” in *23rd IEEE International Conference on Intelligent Transportation Systems (ITSC 2020)*, Ródos, Greece: IEEE, Sep. 2020.
- [39] S. Zhu, D. Li, and M. Liu, “Hindrance-Aware Platoon Formation for Connected Vehicles in Mixed Traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 876–24 890, Dec. 2022.
- [40] M. A. Khan and L. Bölöni, “Convoy driving through ad-hoc coalition formation,” in *11th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS 2005)*, San Francisco, CA: IEEE, Mar. 2005, pp. 98–105.
- [41] T.-S. Dao, “A Decentralized Approach to Dynamic Collaborative Driving Coordination,” PhD Thesis, University of Waterloo, Waterloo, Canada, Aug. 2008.
- [42] T.-S. Dao, J. P. Huissoon, and C. M. Clark, “A strategy for optimisation of cooperative platoon formation,” *Inderscience International Journal of Vehicle Information and Communication Systems*, vol. 3, no. 1, pp. 28–43, 2013.
- [43] L. Hobert, “A study on platoon formations and reliable communication in vehicle platoons,” Master’s Thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Enschede, Netherlands, Jan. 2012.
- [44] O. Dokur, G. Olenski, and S. Katkooi, “An Edge Computing Approach for Autonomous Vehicle Platooning,” in *Internet of Things. IoT through a Multi-disciplinary Perspective*, ser. IFIP Advances in Information and Communication Technology, L. M. Camarinha-Matos, L. Ribeiro, and L. Strous, Eds., 1st ed., vol. 665, Springer, 2022, pp. 332–349.
- [45] D. Su and S. Ahn, “Autonomous platoon formation for VANET-enabled vehicles,” in *International Conference on Information and Communication Technology Convergence (ICTC 2016)*, Jeju City, South Korea: IEEE, Oct. 2016, pp. 247–250.
- [46] A. Sharma, D. Agrawal, N. Roy, S. Bhichar, and R. B. Battula, “POTENT - Decentralized Platoon Management with Heapify for Future Vehicular Networks,” in *Advanced Information Networking and Applications*, ser. Lecture Notes in Networks and Systems, 1st ed., vol. 449, Springer, 2022, pp. 387–398.
- [47] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, “Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, Jul. 2000.
- [48] D. Wu, J. Wu, and R. Wang, “An Energy-efficient and Trust-based Formation Algorithm for Cooperative Vehicle Platooning,” in *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, Honolulu, HI: IEEE, Feb. 2019, pp. 702–707.
- [49] D. Ergenç, D. Sorejevic, and M. Fischer, “Distributed Bio-inspired Configuration of Virtualized Mission-critical Networks,” in *IEEE Global Communications Conference (GLOBECOM 2022)*, Rio de Janeiro, Brazil: IEEE, Dec. 2022, pp. 988–993.
- [50] M. Segata et al., “Multi-Technology Cooperative Driving: An Analysis Based on PLEXE,” *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4792–4806, Aug. 2023.
- [51] J. Heinovski, D. S. Buse, and F. Dressler, “Scalable Simulation of Platoon Formation Maneuvers with PlaFoSim,” in *13th IEEE Vehicular Networking Conference (VNC 2021), Poster Session*, Virtual Conference: IEEE, Nov. 2021, pp. 137–138.
- [52] S. Krauß, “Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics,” PhD Thesis, Mathematical Institute, Köln, Germany, Apr. 1998.
- [53] R. Rajamani, *Vehicle Dynamics and Control* (Mechanical Engineering Series), 2nd ed. Boston, MA: Springer, 2012, p. 498.
- [54] M. Strunz, J. Heinovski, and F. Dressler, “CoOP: V2V-based Cooperative Overtaking for Platoons on Freeways,” in *24th IEEE International Conference on Intelligent Transportation Systems (ITSC 2021)*, Indianapolis, IN: IEEE, Sep. 2021, pp. 1090–1097.
- [55] A. Paranjothi, M. Atiquzzaman, and M. S. Khan, “PMCD: Platoon - Merging approach for cooperative driving,” *Wiley Internet Technology Letters*, vol. 3, no. 1, Jan. 2020.
- [56] A. Hayat et al., “A Holistic Approach to the Energy-Efficient Smoothing of Traffic via Autonomous Vehicles,” in *Intelligent Control and Smart Energy Management*, ser. Springer Optimization and Its Applications, M. J. Blondin, o. P. F. Trovão, and P. M. Pardalos Hicham Chaoui, Eds., 1st ed., vol. 181, Springer, 2012, pp. 285–316.
- [57] C.-H. Bruneau, K. Khadra, and I. Mortazavi, “Flow analysis of square-back simplified vehicles in platoon,” *Elsevier International Journal of Heat and Fluid Flow*, vol. 66, pp. 43–59, Aug. 2017.



Julian Heinovski (heinovski@ccs-labs.org) is a PhD candidate and researcher at the Telecommunications Networks (TKN) group at the School of Electrical Engineering and Computer Science, TU Berlin, Germany. He received his B.Sc. and M.Sc. degrees from the Dept. of Computer Science, Paderborn University, Germany, in 2016 and 2018, respectively. Julian is a Student Member of IEEE and ACM as well as a Member of IEEE Intelligent Transportation Systems Society (ITSS) and IEEE Vehicular Technology Society (VTS). He served as a reviewer for various

manuscripts in the field of vehicular networks, cooperative driving, and intelligent transportation systems. His research interest is in cooperative driving and intelligent transportation systems with a focus on vehicular platooning.



Falko Dressler (dressler@ccs-labs.org) is full professor and Chair for Telecommunication Networks at the School of Electrical Engineering and Computer Science, TU Berlin. He received his M.Sc. and Ph.D. degrees from the Dept. of Computer Science, University of Erlangen in 1998 and 2003, respectively. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, IEEE Trans. on Network Science and Engineering,

Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow as well as an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (sub-6GHz, mmWave, visible light, molecular communication) and wireless-based sensing with applications in ad hoc and sensor networks, the Internet of Things, and Cyber-Physical Systems.