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 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. Since the centralized solver approach assumes global knowledge and requires a complex Mixed Integer Programming solver to compute vehicle-to-platoon assignments, we consider the distributed greedy approach to have the best performance 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 very 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 to 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. 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 very 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 V.

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 destination, or if vehicles are in proximity. The system continuously evaluates the cost and benefit of forming a platoon with a vehicle at the same time. Similar to 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 subsystems 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 on [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 aforementioned strategies for platoon formation show that optimal groupings substantially improve the performance gain. However, 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). While some studies perform a comparison of their centralized optimal solution with a centralized heuristic, a proper comparison between centralized and distributed approaches is still missing from the literature. 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 heuristic for solving the assignment problem. Simulations already indicated that the choice of the approach, and the willingness to deviate from individual objectives, have a huge impact on the resulting assignments. 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 with realistic freeway traffic in PlaFoSim.
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. 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.

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 very 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, 46].

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 (pair-wise) by fuel efficiency. To avoid assigning vehicles to platoons with very 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 to much of the own requirements instead of system-level aspects such as traffic flow. Assigning vehicles to platoons based on their properties is similar to 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. Let a vehicle or platoon be represented by the set

\[ \{n, D, p, l\} \],

where \( n \) is the 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 (\( l = p \) if \( n \) is driving individually). We define \( C \) as the set of all individual vehicles that need to be assigned to a vehicle or platoon and \( P \) as the set of all already existing platoons. The union thereof \( T := C \cup P \) defines the set of all potential assignment targets. 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 \( a_{ij} \)

\[
a_{ij} = \begin{cases} 1, & \text{if } c_i \text{ is assigned to } t_j \\ 0, & \text{otherwise} \end{cases} \tag{2}
\]

We call an assignment \( a_{ij} = 1 \) with \( c_i = t_j \) a self-assignment of vehicle \( c_i \), indicating that the vehicle \( c_i \) is not assigned to another vehicle or platoon but will keep driving individually. 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 \( f(c, t) \)

\[
f(c, t) = \alpha \cdot d_s(c, t) + (1 - \alpha) \cdot d_p(c, t) \tag{3}
\]

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

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

\[
d_p(c, t) = \min(\| p_c - p_t \|, \| \| l_i - p_c \| \|), \quad r \in \mathbb{N} \tag{5}
\]

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 \( c \). Typical approaches for calculating user similarity include the Jaccard coefficient, the cosine similarity, and the multi-dimensional Euclidean distance [47]. 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. Thus, 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:

$$d_s(c, t) \leq 1.0\ ,$$
$$d_p(c, t) \leq 1.0\ ,$$
$$p_c \leq l_t\ ,$$

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. 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

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

subject to the following constraints

$$\forall i \in I: \sum_{j \in J} a_{ij} = 1\ ,$$
$$\forall j \in J: \sum_{i \in I} a_{ij} \leq 1\ ,$$
$$\forall i \in I: a_{ij} = 1 \land c_i = t_j\ ,$$

if $\exists k \in I: a_{ik} = 1 \land c_k = t_j \land c_k \neq t_j$.

To summarize, we try to find the best fitting platoon candidate (i.e., the candidate with the biggest similarity, or, in other 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 assignment 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 $c_k$ (≠ $t_j$) 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 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

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

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

$$f(37, 13) = 0.242$$
$$f(20, 5) = 0.31$$

with a total deviation of $0.242 + 0.31 = 0.552$. 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 using centralized solver approach

In this approach, the optimization problem is solved periodically for every vehicle in the scenario at the same time. We aim
to minimize the overall deviation from the desired metrics while assigning as many vehicles to platoons as possible. 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 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 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. In case of a self-assignment, no maneuver is triggered and the vehicle stays individual.

E. Solution using centralized greedy approach

The complexity of centralized optimization in general has been shown to be NP hard [22]. Thus, we propose the centralized greedy approach that follows 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. It, however, uses greedy heuristics to compute the assignments.

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

---

**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 $p_{ci} > l_{tj}$ 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 \{ $a_{ij}, f_{ij}$ \};

end if

end for

end if

end for

**Output:** all decision variables and deviations: list(\{ $a_{ij}, f_{ij}$ \})

---

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

**Input:** all decision variables and deviations: 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 $t_j \neq c_i$

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 $p_{ci} > l_{tj}$ 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;

end if

end for

end if

end for

**Output:** all possible assignments: list(\{ $c_i, t_j, f(c_i, t_j)$ \})

---

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

**Input:** all possible assignments: list(\{ $c_i, t_j, f(c_i, t_j)$ \})

for all (unique) vehicles $c_i$ in the list do

$s \leftarrow \{t_j | t_j \in \text{list}(\{ c_i, t_j, f(c_i, t_j) \}) \}; \; \triangleright \text{all possible } t_j$

if $|s| > 0$ then

$b \leftarrow \text{min } f(c_i, t_j), t_j \in s; \; \triangleright \text{select best candidate}$

let $c_i$ join $b$;
remove list entries with $c_i$ or $b; \; \triangleright \text{apply constraints}$

end if

**Output:** vehicles performing a join maneuver
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 \( t_j \). 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 central greedy heuristic also produces two platoons, but it does not necessarily compute the aforementioned optimal solution. Instead, it might select
\[
\begin{align*}
  f(13,5) &= 0.519 \\
  f(37,20) &= 0.33
\end{align*}
\]
with a total deviation of 0.519 + 0.33 = 0.849, given the current order of possible assignments. 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: \( \{5,13\} \) and \( \{20,37\} \).

F. Solution using distributed greedy approach

In contrast to the central greedy approach, our distributed greedy approach 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. 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. A heuristic very similar to 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 central 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 can trigger join attempts for the following selection:
\[
\begin{align*}
  f(13,5) &= 0.519 \\
  f(20,13) &= 0.188 \\
  f(37,20) &= 0.33
\end{align*}
\]
The outcome of this depends on the order and the timing of the join attempts and can vary between one and two resulting platoons.

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 [48] 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 PlaFoSim\(^1\) in our study as we are more interested in large-scale effects of platoon management rather than microscopic control of the involved vehicles [49]. 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.\(^2\) 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

\(^1\)https://www.plafosim.de

\(^2\)https://github.com/heinovski/plafosim
We model vehicle demand in a macroscopic way via a flow with constant insertion (i.e., departure rate) to roughly keep ACC. A variation of 10%, and limited to values from vehicles, and 1 formed platoon consisting of 3 members. All members of one artificial limits of our distribution.

The desired driving speed is 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. The desired driving speed is sampled from a normal distribution with a mean of 120 km/h, a variation of 10%, and limited to values from [80 km/h, 160 km/h]. Vehicles start their trips individually, using either the popular Krauss model [50] or a standard ACC [46].

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 using our distributed (greedy) approach (see Section III-F) for platoon formation
- **centralized greedy** – vehicle platooning using our centralized (greedy) approach (see Section III-E) for platoon formation
- **centralized solver** – vehicle platooning using our optimal approach (see Section III-D) for platoon formation

We model vehicle demand in a macroscopic way via a flow with constant insertion (i.e., 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 approaches and not in the maximum possible traffic flow for every approach. The corresponding departure rate for a given target density is calculated as

\[ t_{\text{expected}} = \frac{d_{\text{trip}}}{v_{\text{desired}}} \]  
\[ P_{\text{arrival}} = \frac{1}{t_{\text{expected}}} \]

where \( t_{\text{expected}} \) is the expected travel time of the vehicles, given their fixed trip length and their desired driving speed (in m/s), \( P_{\text{arrival}} \) is the probability of vehicle arrivals per second, \( R_{\text{departure}} \) is the corresponding departure rate, given the target density \( D_{\text{desired}} \), number of lanes \( L \), and the length of the road \( d_{\text{road}} \).

A summary of all simulation parameters for the road network and traffic can be found in Table I. 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 [49]. A summary of all simulation parameters for the road network and traffic can be found in Table I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freeway length</td>
<td>100 km</td>
</tr>
<tr>
<td>Number of lanes</td>
<td>3</td>
</tr>
<tr>
<td>Ramp interval</td>
<td>10 km</td>
</tr>
<tr>
<td>Depart positions</td>
<td>random on-ramp</td>
</tr>
<tr>
<td>Arrival positions</td>
<td>random off-ramp at trip end</td>
</tr>
<tr>
<td>Fixed trip length</td>
<td>50 km</td>
</tr>
<tr>
<td>Desired speed</td>
<td>( N (120 \text{ km/h}, 10%) )</td>
</tr>
<tr>
<td>Min. desired speed</td>
<td>80 km/h</td>
</tr>
<tr>
<td>Max. desired speed</td>
<td>160 km/h</td>
</tr>
<tr>
<td>Target density</td>
<td>5, 10, 15, 20 and 25 veh/km/lane</td>
</tr>
<tr>
<td>Target no. of vehicles</td>
<td>1500, 3000, 4500, 6000 and 7500 veh</td>
</tr>
<tr>
<td>Departure rate</td>
<td>3564, 7129, 10693, 14257 and 17822 veh/h</td>
</tr>
</tbody>
</table>

Cars Following model | Krauss, ACC, and CACC
Krauss desired headway | 1 s
ACC desired headway | 1 s
CACC desired gap | 5 m
Max. speed \( v_{\text{max}} \) | 200 km/h
Max. acceleration | 2.5 m/s\(^2\)
Max. deceleration | 10 m/s\(^2\)
Vehicle length | 5 m
Min. gap between veh. | 2.5 m

Achieved by solving a MIP using the Gurobi optilization library. 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

---

\(^4\)Only very few vehicles (0.4%) choose a desired driving speed equal to the artificial limits of our distribution.

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.

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.

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 I and II, 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 $\approx 8000$. 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 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...
Figure 5. The 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 0 and 800.

Figure 6. The 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.

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., $\leq 90$ km/h and $\geq 150$ 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. 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). Similarly to the found candidates metric, we count these vehicles within the filtered candidates metric for all vehicles and in all approaches. Vehicles that are not available because of the limited communication range with the distributed greedy approach are not counted as filtered.

The results are shown in Figure 5. Due to the global knowledge and decision making for all vehicles in the scenario, both centralized approaches filter many more candidates than the distributed greedy approach, which has only limited local knowledge. 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.

D. Formation Process

After analyzing the number of platooning opportunities, we now take a 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. Thus, it includes execution of the formation algorithm to compute a vehicle-to-platoon assignment as well as the corresponding join process itself. Similarly, this time is measured if a vehicle became a leader of a newly formed platoon.

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. 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. 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 take a 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 (15). 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 very 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 are able to to 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 observe the effect of high traffic very prominently. As expected, the driving speed is reduced for all approaches when the scenario becomes more crowded due to natural traffic effects. The effect is generally worse for non-platooning approaches due to larger safety gaps between vehicles, resulting in stop-and-go traffic [51]. For all platooning approaches, the speed is similar to 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 similar to 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
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 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 vehicles are included. The centralized solver approach performs the worst 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).
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 aforementioned 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 flow 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 10. The average relative deviation from the desired driving speed per vehicle for all simulated approaches. The triangle within a box depicts the mean value.

Figure 11. The 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.

Figure 12. The 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.
Figure 13. The 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 12. The general tendency follows our expectations: the ratio increases with vehicle density. This 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)\(^6\) 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 [52], 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 accelerations & decelerations 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 take a 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.

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

V. 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

\(^6\)https://www.hbefa.net/e/index.html
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 are able to 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. Since the centralized solver approach assumes global knowledge and requires a complex MIP solver to compute vehicle-to-platoon assignments, we consider the distributed greedy approach to have the best performance 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 focus more on the heterogeneity of vehicles by considering more of their properties such as their destination and capabilities within the similarity function.

VI. ACKNOWLEDGEMENTS

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

REFERENCES


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 serves as a reviewer for 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 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.