Fuel-Efficient En Route Formation of Truck Platoons

Sebastian van de Hoef, Karl Henrik Johansson, *Fellow, IEEE*, and Dimos V. Dimarogonas, *Member, IEEE*

Abstract—The problem of how to coordinate a large fleet of trucks with a given itinerary to enable fuel-efficient platooning is considered. Platooning is a promising technology that enables trucks to save significant amounts of fuel by driving close together and thus reducing air drag. A setting is considered in which each truck in a fleet is provided with a start location, a destination, a departure time, and an arrival deadline from a higher planning level. Fuel-efficient plans should be computed. The plans consist of routes and speed profiles that allow trucks to arrive by their arrival deadlines. Hereby, trucks can meet on common parts of their routes and form platoons, resulting in decreased fuel consumption. We formulate a combinatorial optimization problem that combines plans involving only two vehicles. We show that this problem is difficult to solve for large problem instances. Hence, a heuristic algorithm is proposed. The resulting plans are further optimized using convex optimization techniques. The method is evaluated with Monte Carlo simulations in a realistic setting. We demonstrate that the proposed algorithm can compute plans for thousands of trucks and that significant fuel savings can be achieved.

Index Terms—Energy-efficient road freight transport, vehicle platooning, automated vehicle operation, intelligent logistics, connected vehicles, cooperative systems.

I. INTRODUCTION

PLATOONING is foreseen to become a common element in intelligent transportation systems. The term refers to a group of vehicles forming a road train without any physical coupling between them. A short inter-vehicle distance is maintained by automatic control and vehicle-to-vehicle communication. Platooning has received a lot of attention due to its potential to increase road throughput by reducing the inter-vehicle gaps. It can also help facilitate the (semi-)automatic operation of vehicles [1]. This paper focuses on the potential of platooning in reducing fuel consumption. Similar to what racing cyclists exploit, the follower vehicles, and to a lesser degree the lead vehicle, experience a reduction in air drag, which translates into reduced fuel consumption [2]–[4]. Advances in wireless communication, satellite-based position, and advanced driver support systems have made the wide deployment of platooning systems feasible and have attracted the attention of major truck manufacturers. Increased fuel costs

The authors are with the ACCESS Linnaeus Centre, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden (e-mail: shvdh@kth.se; kallej@kth.se; dimos@kth.se).

and environmental awareness make the implementation of such systems likely in the near future.

Using platooning to reduce fuel consumption for a large number of trucks leads to a challenging coordination problem. Consider two trucks that travel between the same two regions but from different locations and at approximately the same time. Then the trucks can adjust their speeds slightly at the beginning of their journeys, form a platoon at the start of the common part of their route and thus save fuel during part of their trips. This approach might involve one of the trucks having to drive slightly faster before the two merge, which increases air-drag and consequently fuel consumption during the initial phase. One truck might instead slow down to let the other truck catch up but then travel at an increased speed later on to arrive at its destination on time. If many trucks are involved, it is not straightforward to compute an optimal plan for all trucks.

The main contribution of the paper is to derive an efficient and scalable method to coordinate platooning of a large number of trucks in a fuel-efficient way explicitly considering the effect of speed and platooning on the fuel consumption. The core novelty is that the computation of platoon plans is done in three computationally and mathematically tractable stages. The first stage involves the computation of platoon plans only taking into account two vehicles at a time. The second stage selects one such plan for each vehicle. Since the problem to be solved in the second stage is shown to be NP-hard, an iterative algorithm to compute heuristic solutions is proposed. In the third stage, the resulting plans are optimized further using convex optimization techniques. The potential of the method is demonstrated using Monte Carlo simulations. The simulations demonstrate that the method is able to significantly reduce the fuel consumption of a fleet of vehicles as well as to handle a realistic number of transport assignments for the example of Sweden.

The efficient operation of transport systems is a widely studied field due to its large economic and environmental impact. Planning conducted by transportation operators ranges all the way from strategic over tactical to operational planning [5]. The planning on the latter stage typically happens on the level of departure and arrival times which is the input to the problem considered in this paper. Research on eco-routing aims to reduce fuel consumption by appropriate choice of route and travel speeds for individual vehicles [6]. Furthermore, operators of road infrastructure use variable speed limits, ramp metering, variable route recommendations, etc. to improve safety and efficiency of the road transportation system [7].

Various aspects of platoon coordination have been considered in the literature. Larsson *et al.* [8] formulate a mixed

integer linear programming problem, without considering the speed dependency of fuel consumption, and prove that the problem is NP-hard. Liang *et al.* [9] consider a simple catch-up coordination scheme and evaluate it on real fleet data. In [10] local controllers for coordinating the formation of platoons are proposed. Meisen *et al.* [11] use data-mining to identify economic platoons based on various criteria. Unlike this paper the method presented in [11] allows that trucks wait for other trucks to form the platoon. Preliminary material used in this paper has been presented in [12] and [13].

One of the key elements to make the problem tractable for realistic numbers of trucks, is to select a subset of vehicles called coordination leaders to which other vehicles adapt. The way we select this subset of vehicles is inspired by a clustering algorithm called partitioning around medoids [14]. Clustering is present in a variety of different contexts. A large body of research focuses on clustering methods as an analysis tool to structure, understand, and classify large data sets [15], [16]. Examples include the clustering of graphs in the scope of community detection [17]–[19]. Closely related to our problem of choosing coordination leaders is leader election, where a group of agents has to jointly determine a leader [20], [21]. The approach developed in this paper can be seen as local leader election, where pairwise fuel savings are interpreted as preferences for trucks being coordination leaders.

The paper is organized as follows. We start in Section II by formulating the coordination problem and introduce the structure of the proposed solution in Section III. In Section IV the problem is broken down to a purely combinatorial problem of selecting coordination leaders and the problem is shown to be NP-hard. In Section V an iterative algorithm is developed to find heuristic solutions to the combinatorial problem. Section VI discusses how to jointly optimize the plans when the trucks are constrained to the platoons as proposed by the algorithm from Section V. In Section VII the method is demonstrated by Monte Carlo simulations in a realistic scenario with up to 5 000 trucks on the Swedish road network.

II. PROBLEM FORMULATION

In this section we formulate the problem and introduce the notation. We have an index set $\mathcal{N}_c$ of finitely many transport assignments, each tied to a specific truck. A transport assignment $\mathcal{A} = (\mathcal{P}^S, \mathcal{P}^D, t^S, t^D)$ consists of a start position $\mathcal{P}^S$, a destination $\mathcal{P}^D$, a start time t^S, and an arrival deadline t^D. We model the road network as a directed graph $\mathcal{G}_r = (\mathcal{N}_r, \mathcal{E}_r)$ with nodes $\mathcal{N}_r$ and edges $\mathcal{E}_r$. Nodes correspond to intersections or endpoints in the road network and edges correspond to road segments connecting these intersections. The function $L : \mathcal{E}_r \to \mathbb{R}^+$ maps each edge in $\mathcal{E}_r$ to the length of the corresponding road segment. A vehicle position is a pair $(e, x) \in \mathcal{E}_r \times [0, L(e)]$ where e indicates the current road segment and x how far the vehicle has traveled along that segment.

The goal is to compute fuel-efficient plans for the trucks that ensure arrival before each trucks' individual deadline. Each plan includes a route in the road network from start to destination and encodes a piecewise constant speed trajectory. The speed is constrained to a range of feasible speeds

$[v_{\min}, v_{\max}]$, which is supposed to be the same for all vehicles and road segments.[1] For the sake of this high-level planning, it is reasonable to assume that trucks change their speed instantaneously.

Definition 1 (Vehicle Plan): A vehicle plan $\mathcal{P} = (\mathbf{e}, \mathbf{v}, \hat{\mathbf{t}})$ consists of a route $\mathbf{e}$, a speed sequence $\mathbf{v}$, and a time sequence $\hat{\mathbf{t}}$. The route is a sequence of N_e edges in the road network: $\mathbf{e} = (\mathbf{e}[1], \dots, \mathbf{e}[N_e])$, $\mathbf{e}[i] \in \mathcal{E}_r$. The speed sequence is a sequence of N_v speeds $\mathbf{v} = (\mathbf{v}[1], \dots, \mathbf{v}[N_v])$, where speeds are within the feasible speed range $0 < v_{\min} \leq \mathbf{v}[i] \leq v_{\max}$. The time sequence $\hat{\mathbf{t}} = (\hat{\mathbf{t}}[1], \dots, \hat{\mathbf{t}}[N_v + 1])$ defines when the speed changes. Speed $\mathbf{v}[i]$ is selected from $\hat{\mathbf{t}}[i]$ until $\hat{\mathbf{t}}[i + 1]$.

Note that changes in speed can happen, in principle, everywhere and not only at the beginning of route segments. The symbols N_e and N_v are introduced for notational convenience and their value may be different for different vehicle plans.

We want to compute a vehicle plan for each truck. A valid vehicle plan brings the truck from its start position $\mathcal{P}^S = (e^S, x^S)$, where it is at time t^S, to its destination $\mathcal{P}^D = (e^D, x^D)$ before its deadline t^D.

Vehicle plans are constrained by two conditions. The first condition requires the trip to start at the start time $\hat{\mathbf{t}}[1] = t^S$ and ends before the deadline $\hat{\mathbf{t}}[N_v + 1] = t^A \leq t^D$. The second condition ensures that the truck arrives at its destination when the trip ends, i.e., the distance traveled is

$$D := \sum_{i=1}^{N_e - 1} L(\mathbf{e}[i]) + x^D - x^S = \sum_{i=1}^{N_v} \mathbf{v}[i](\hat{\mathbf{t}}[i + 1] - \hat{\mathbf{t}}[i]).$$

We introduce the notion of trajectories as functions of continuous time. A vehicle trajectory consists of an edge trajectory ϵ and a linear position trajectory ζ. The edge trajectory for $t \in [t^S, t^A)$ is given by $\epsilon(t) = \mathbf{e}[j]$ where j depends on t and is the largest integer that satisfies $\sum_{i=1}^{j-1} L(\mathbf{e}[i]) - x^S < \int_{t^S}^{t} \phi(\tau)d\tau$, and where the speed trajectory $\phi(t) = \mathbf{v}[i]$ for $t \in [\hat{\mathbf{t}}[i], \hat{\mathbf{t}}[i + 1])$, $i \in \{1, \dots, N_v\}$. The linear position, i.e., the second element of the position, at time t is given by $\zeta(t) = \int_{t^S}^{t} \phi(\tau)d\tau - \sum_{i=1}^{j-1} L(\mathbf{e}[i]) + x^S$.

When trucks platoon, their positions coincide in our model and hence we neglect the physical dimension of the trucks. Each platoon consists of a platoon leader and a number of platoon followers. We introduce the platoon trajectory $\pi_n : [t_n^S, t_n^A] \to \{0, 1\}$ for truck $n \in \mathcal{N}_c$. A platoon trajectory equals 1 when truck n is a platoon follower and 0 when it is a platoon leader or traveling alone. Thus $\pi_n(t) = 1$ implies that there is another truck $m \in \mathcal{N}_c$ with $m \neq n$ and $(\epsilon_n(t), \zeta_n(t)) = (\epsilon_m(t), \zeta_m(t))$.

We model the fuel consumption per distance traveled as a function of the speed and of whether the truck is a platoon follower or not. A platoon leader is assumed to have the same fuel consumption as a truck that travels alone while a platoon

---

[1]The approach developed in this paper can be generalized in order to relax this assumption.
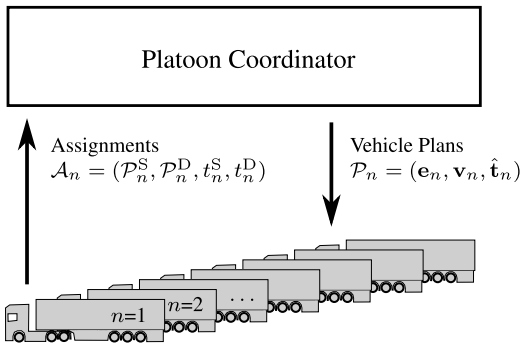
Fig. 1. Schematic of the platoon coordinator. Trucks provide their assignment data and the platoon coordinator computes fuel efficient vehicle plans.

follower has a reduced fuel consumption. Hereby, we neglect the relatively small reduction in fuel consumption when a truck is a platoon leader compared to traveling by itself. We denote the fuel consumption per distance traveled as $f : [v_{\min}, v_{\max}] \times \{0, 1\} \rightarrow \mathbb{R}^+$ where

$$f(v, p) = \begin{cases} f_0(v) & \text{if } p = 0 \\ f_\mathrm{p}(v) & \text{if } p = 1. \end{cases} \qquad (1)$$

The function f_0 models the fuel consumption when the truck is a platoon leader or when it travels solo, and f_p the fuel consumption when the truck is a platoon follower. These functions can either be derived from an analytical model or fitted to data [22]. We purposely omit that fuel consumption depends on road and vehicle parameters in order to keep the presentation concise. All the presented results can be augmented to handle those additional parameters.

The problem that we want to solve is to find a vehicle plan for each vehicle, and we want to minimize the combined fuel consumption of these plans. The total fuel consumption $F(\phi_n, \pi_n)$ associated with vehicle n's plan is given by integrating the fuel consumption according to (1) over the duration of the trip

$$F(\phi_n, \pi_n) = \int_{t_n^\mathrm{S}}^{t_n^\mathrm{A}} f(\phi_n(t), \pi_n(t))\phi_n(t)\mathrm{d}t, \qquad (2)$$

where ϕ_n is the speed trajectory, π_n the platoon trajectory, t_n^S the start time, and t_n^A the arrival time of truck n. The combined fuel consumption F_c is given by

$$F_\mathrm{c} = \sum_{n \in \mathcal{N}_\mathrm{c}} F(\phi_n, \pi_n). \qquad (3)$$

Our primary goal is to compute vehicle plans that minimize F_c.

III. PLATOON COORDINATOR

Consider the centralized platoon coordinator in Fig. 1. Trucks connect to the coordinator via vehicle-to-infrastructure communication and share their assignment data. The coordinator then computes fuel-efficient vehicle plans for the trucks. These plans are sent to the trucks and executed. This process

is repeated whenever there is updated information, such as deviations from the plans and new assignments. The current vehicle position is then the new start position of an assignment that is already being executed.

The computation of the vehicle plans happens in four stages:
1) Computation of the routes $\mathbf{e}_n$, $n \in \mathcal{N}_\mathrm{c}$: routes are calculated using an algorithm for route calculation in road networks.
2) Computation of pairwise vehicle plans: many plans involving two vehicles are computed. The fuel savings of these plans are recorded as the coordination graph $\mathcal{G}_\mathrm{c}$ introduced in the following.
3) Selection of pairwise plans: a consistent subset of the plans computed in the previous stage is combined by selecting a subset $\mathcal{N}_\mathrm{l} \subset \mathcal{N}_\mathrm{c}$, so-called called coordination leaders.
4) Joint speed-profile optimization: the selected pairwise plans are jointly optimized for lower fuel consumption while platoon partners as well as merge and split locations are kept as computed in step 3).

Stage 1) computes the routes $\mathbf{e}_n$, $n \in \mathcal{N}_\mathrm{c}$ and stages 2)-4) compute the speed sequences $\mathbf{v}_n$ and time sequences $\hat{\mathbf{t}}_n$ for $n \in \mathcal{N}_\mathrm{c}$ making use of the ability of the trucks to form platoons in order to achieve low fuel consumption. Algorithms for route calculation in road networks are well developed [6], [23] and not further discussed in this paper. We discuss stages 2)-4) in the following sections.

IV. SELECTING PAIRWISE VEHICLE PLANS

In this section, we formulate a combinatorial optimization problem corresponding to the second and third computation stage introduced in Section III. The problem is proven to be NP-hard which motivates the heuristic algorithm developed in Section V.

To begin with, we need to be able to compute what we call a default plan. This is a valid vehicle plan according to Definition 1 with either the lowest possible or most fuel optimal constant speed.

Definition 2 (Default Plan): The default plan is a vehicle plan $\mathcal{P} = (\mathbf{e}, \mathbf{v}, \hat{\mathbf{t}})$ with speed sequence $\mathbf{v} = (v_{\mathrm{cd}})$ and time sequence $\hat{\mathbf{t}} = (t^\mathrm{S}, D/v_{\mathrm{cd}})$. The most fuel optimal speed without platooning v_{cd} is computed as

$$v_{\mathrm{cd}} = \operatorname*{argmin}_{v \in (v_{\mathrm{cm}}, v_{\max}]} f_0(v),$$

where v_{cm} is the lowest constant speed to arrive before the deadline:

$$v_{\mathrm{cm}} = \max\left(v_{\min}, \frac{D}{t^\mathrm{D} - t^\mathrm{S}}\right).$$

An adapted plan, as introduced next, is such that the speed sequence $\mathbf{v}_n$ and time sequence $\hat{\mathbf{t}}_n$ of a follower truck n is adapted in a way that allows the follower to platoon during part of its journey with a leader m. The leader sticks to its default plan, which is important in order to be able to compose these plans. The plan is computed in a way that minimizes the fuel consumption of n.
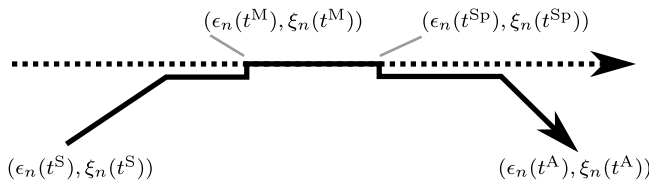
Fig. 2. Overview of the relevant time instances of the adapted plan. The solid line illustrates the route of the adapted plan n and the dashed line the one of the plan that it is adapted to and has index m. The parallel sections of the line indicate that the trucks share the route and the section where the lines are on top of each other indicates that the trucks platoon there.

Definition 3 (Adapted Plan): An adapted plan is a vehicle plan $\mathcal{P}_n = (\mathbf{e}_n, \mathbf{v}_n, \hat{\mathbf{t}}_n)$ adapted to vehicle plan $\mathcal{P}_m = (\mathbf{e}_m, \mathbf{v}_m, \hat{\mathbf{t}}_m)$, such that $(\epsilon_n(t), \xi_n(t)) = (\epsilon_m(t), \xi_m(t))$ for $t \in [\hat{\mathbf{t}}_n[2], \hat{\mathbf{t}}_n[N_{\mathrm{v},n}])$.

We denote the merge time as $t^{\mathrm{M}} = \hat{\mathbf{t}}_n[2]$ and the split time as $t^{\mathrm{Sp}} = \hat{\mathbf{t}}_n[N_{\mathrm{v},n}]$. Truck n becomes the platoon follower of truck m at time t^{M}, stays platoon follower until t^{Sp}, when the two trucks separate. This sequence of events occurs only once. Fig. 2 illustrates the adapted plan. We denote the speed trajectory ϕ corresponding to the speed sequence $\mathbf{v}$ and the time sequence $\hat{\mathbf{t}}$ of the adapted vehicle plan of truck n adapted to truck m as $\phi_{n,m}$.

The fuel consumption of truck n with its plan adapted to truck m is modeled as in (2). We denote the platoon trajectory of the adapted plan $\pi_{n,m}(t)$. We have that $\pi_{n,m}(t) = 1$ for $t \in [t^{\mathrm{M}}, t^{\mathrm{Sp}})$ and $\pi_{n,m}(t) = 0$ for $t \in [t^{\mathrm{S}}, t^{\mathrm{M}}) \cup [t^{\mathrm{Sp}}, t^{\mathrm{A}})$. The fuel consumption of m is not altered by the fact that n and m platoon, since m's speed trajectory does not change and since m takes the role of a platoon leader. The reduction in fuel consumption that results from n implementing the adapted plan and not its default plan is $\Delta F(n, m) = F(\phi_n, \pi_n) - F(\phi_{n,m}, \pi_{n,m})$ where $\pi_n \equiv 0$, which is positive if n adapting to m saves fuel. If no plan that is adapted to m exists for n, we define $\Delta F(n, m) = 0$. There might exist no adapted plan because the routes do not overlap or because the constraint on the maximum speed in conjunction with the arrival deadline makes it impossible for the trucks to form a platoon.

We now compute ΔF for all 2-permutations in $\mathcal{N}_{\mathrm{c}}$. We are only interested in adapted plans that save fuel, i.e., for which ΔF is positive. We can conveniently collect this information in a weighted graph that we call the coordination graph.

Definition 4 (Coordination Graph): The coordination graph is a weighted directed graph $\mathcal{G}_{\mathrm{c}} = (\mathcal{N}_{\mathrm{c}}, \mathcal{E}_{\mathrm{c}}, \Delta F)$. Recall that the elements of $\mathcal{N}_{\mathrm{c}}$ represent the trucks. $\mathcal{E}_{\mathrm{c}} \subseteq \mathcal{N}_{\mathrm{c}} \times \mathcal{N}_{\mathrm{c}}$ is a set of edges, and $\Delta F : \mathcal{E}_{\mathrm{c}} \to \mathbb{R}^+$ are edge weights, such that there is an edge $(n, m) \in \mathcal{E}_{\mathrm{c}}$, if the adapted plan of n to m saves fuel compared to i's default plan, i.e., $\mathcal{E}_{\mathrm{c}} = \{(i, j) \in \mathcal{N}_{\mathrm{c}} \times \mathcal{N}_{\mathrm{c}} : \Delta F(i, j) > 0, i \neq j\}$.

Furthermore, we introduce the set of in-neighbors of a node $n \in \mathcal{N}_{\mathrm{c}}$ as $\mathcal{N}_n^{\mathrm{i}} = \{i \in \mathcal{N}_{\mathrm{c}} : (i, n) \in \mathcal{E}_{\mathrm{c}}\}$ and the set of out-neighbors n as $\mathcal{N}_n^{\mathrm{o}} = \{i \in \mathcal{N}_{\mathrm{c}} : (n, i) \in \mathcal{E}_{\mathrm{c}}\}$. We define the the maximum over an empty set to be zero, i.e., $\max_{i \in \emptyset}(\cdot) = 0$.

With these definitions, we are ready to formulate the problem of finding a fuel optimal set of coordination leaders $\mathcal{N}_{\mathrm{l}}$.

Input: $\mathcal{G}_{\mathrm{c}}$
Output: $\mathcal{N}_{\mathrm{l}}$
  $\mathcal{N}_{\mathrm{l}} \leftarrow \emptyset$
  while $\{\bar{n} \in \mathcal{N}_{\mathrm{c}} : \Delta u(\bar{n}, \mathcal{N}_{\mathrm{l}}) > 0\} \neq \emptyset$ **do**
    Select $n \in \{\bar{n} \in \mathcal{N}_{\mathrm{c}} : \Delta u(\bar{n}, \mathcal{N}_{\mathrm{l}}) > 0\}$
    if $n \in \mathcal{N}_{\mathrm{l}}$ **then**
      $\mathcal{N}_{\mathrm{l}} \leftarrow \mathcal{N}_{\mathrm{l}} \setminus \{n\}$
    else
      $\mathcal{N}_{\mathrm{l}} \leftarrow \mathcal{N}_{\mathrm{l}} \cup \{n\}$
    end if
  end while

Fig. 3. The clustering algorithm, which is an iterative algorithm to compute the set of coordination leaders $\mathcal{N}_{\mathrm{l}}$.

Problem 1: Given as input a coordination graph $\mathcal{G}_{\mathrm{c}} = (\mathcal{N}_{\mathrm{c}}, \mathcal{E}_{\mathrm{c}}, \Delta F)$ find a subset $\mathcal{N}_{\mathrm{l}} \subset \mathcal{N}_{\mathrm{c}}$ of nodes that maximizes

$$f_{\mathrm{ce}}(\mathcal{N}_{\mathrm{l}}) = \sum_{i \in \mathcal{N}_{\mathrm{c}} \setminus \mathcal{N}_{\mathrm{l}}} \max_{j \in \mathcal{N}_i^{\mathrm{o}} \cap \mathcal{N}_{\mathrm{l}}} \Delta F(i, j). \tag{4}$$

The coordination leaders select their default plans. The remaining assignments, called coordination followers, select their plans adapted to the coordination leader that yields the largest fuel savings $\Delta F(n, m)$. Since the selection of adapted plans does not alter the speed trajectories of the coordination leaders, several coordination followers can select the same coordination leader without affecting the fuel savings that result from this adaptation, potentially resulting in platoons of more than two vehicles. The objective function $f_{\mathrm{ce}}(\mathcal{N}_{\mathrm{l}})$ equals the sum of all these fuel savings. If $(n, m) \in \mathcal{E}_{\mathrm{c}}$ with $n \in \mathcal{N}_{\mathrm{c}} \setminus \mathcal{N}_{\mathrm{l}}$ and $m = \arg \max_{m \in \mathcal{N}_n^{\mathrm{o}} \cap \mathcal{N}_{\mathrm{l}}} \Delta F(n, m)$, we say that n is the coordination follower of m and m is the coordination leader of n. If m has no out-neighbor in $\mathcal{N}_{\mathrm{l}}$, then $\max_{m \in (\mathcal{N}_n^{\mathrm{o}} \cap \mathcal{N}_{\mathrm{l}})} \Delta F(n, m) = \max_{m \in \emptyset} \Delta F(n, m) = 0$.

Problem 1 is a combinatorial optimization problem. We can compute an optimal solution in finite time by using exhaustive search or a branch and bound technique. However, the computational complexity of such an exact computation might be too high. In fact, we show in the following that this problem is an NP-hard problem. This a strong indicator that searching for an algorithm that computes solutions for every coordination graph and scales well in the size of the coordination graph is futile [24].

Proposition 1: Problem 1 is NP-hard, where the size of the input $\mathcal{G}_{\mathrm{c}}$ is measured as $|\mathcal{N}_{\mathrm{c}}| + |\mathcal{E}_{\mathrm{c}}|$.

The proof can be found in the appendix.

One disadvantage of the approach presented in this section is that each truck can only join one platoon. This can however be somewhat mitigated by frequent re-planning. For instance, at some later point in time, it might turn out more beneficial for a truck to leave its current platoon and join another one.

V. ITERATIVE SELECTION OF COORDINATION LEADERS

In this section we present an algorithm that computes heuristic solutions to Problem 1. Motivated by the result that Problem 1 is NP-hard, we apply an iterative strategy that converges to a local maximum.

Consider the algorithm in Fig. 3, which we call the clustering algorithm. The input is a coordination graph $\mathcal{G}_c$ and the output is a set of coordination leaders $\mathcal{N}_l$. Initially $\mathcal{N}_l$ is an empty set. In each iteration, a node $n \in \mathcal{N}_c$ is selected for which the objective function f_{ce} is increased if it is added to $\mathcal{N}_l$ or removed from $\mathcal{N}_l$, and $\mathcal{N}_l$ is updated accordingly. The difference in f_{ce} when adding or removing a node in $\mathcal{N}_c$ to or from the set of coordination leaders $\mathcal{N}_l$ is given by a function Δu. The algorithm iterates until no further increase of f_{ce} is possible.

The function Δu that measures how much is gained from switching whether n belongs to $\mathcal{N}_l$ is defined as follows:

$$\Delta u(n, \mathcal{N}_l) = \begin{cases} f_{ce}(\mathcal{N}_l \setminus \{n\}) - f_{ce}(\mathcal{N}_l) & \text{if } n \in \mathcal{N}_l \\ f_{ce}(\mathcal{N}_l \cup \{n\}) - f_{ce}(\mathcal{N}_l) & \text{otherwise.} \end{cases} \quad (5)$$

If $n \notin \mathcal{N}_l$, we get

$$f_{ce}(\mathcal{N}_l \cup \{n\}) - f_{ce}(\mathcal{N}_l)$$
$$= \sum_{i \in \mathcal{N}_n^i \setminus \mathcal{N}_l} \left(\max_{j \in \mathcal{N}_i^o \cap (\mathcal{N}_l \cup \{n\})} \Delta F(i, j) - \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_l} \Delta F(i, j) \right)$$
$$- \max_{i \in \mathcal{N}_n^o \cap \mathcal{N}_l} \Delta F(n, i).$$

The sum over i covers nodes that can select n as their new coordination leader. The last summand accounts for n possibly not being a coordination follower any longer.

If $n \in \mathcal{N}_l$, we get

$$f_{ce}(\mathcal{N}_l \setminus \{n\}) - f_{ce}(\mathcal{N}_l)$$
$$= \sum_{i \in \mathcal{N}_n^i \setminus \mathcal{N}_l} \left(\max_{j \in \mathcal{N}_i^o \cap (\mathcal{N}_l \setminus \{n\})} \Delta F(i, j) - \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_l} \Delta F(i, j) \right)$$
$$+ \max_{i \in \mathcal{N}_n^o \cap (\mathcal{N}_l \setminus \{n\})} \Delta F(n, i).$$

The sum over i covers nodes that can have n as their coordination leader before the change. The last summand accounts for n possibly becoming a coordination follower.

In this paper, we consider two methods to select n from the set $\{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_l) > 0\}$. The first method is to select n in a greedy manner according to $n = \arg \max_{\bar{n} \in \mathcal{N}_c} \Delta u(\bar{n}, \mathcal{N}_l)$. The second method is to choose n randomly with equal probability from the set $\{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_l) > 0\}$.

The clustering algorithm is guaranteed to converge in finite time. This is due to the number of possible subsets of $\mathcal{N}_c$ being finite and thus the number of possible assignments of $\mathcal{N}_l$ is finite. In every iteration $f_{ce}(\mathcal{N}_l)$ strictly increases which means that $\mathcal{N}_l$ changes in every iteration and the same assignment for $\mathcal{N}_l$ never reoccurs. So in the worst case the clustering algorithm iterates over all subsets of $\mathcal{N}_c$ before termination.

The clustering algorithm can be efficient. Note for instance that the function Δu can be computed based on the sub-graph induced by the one- and two-hop neighbors of n only. This means that the average complexity of computing Δu is a function of the average node degree and not of the number of nodes in the coordination graph. Furthermore, if a node is added to or removed from $\mathcal{N}_l$, then only the Δu for the two-hop neighbors needs to be recomputed.
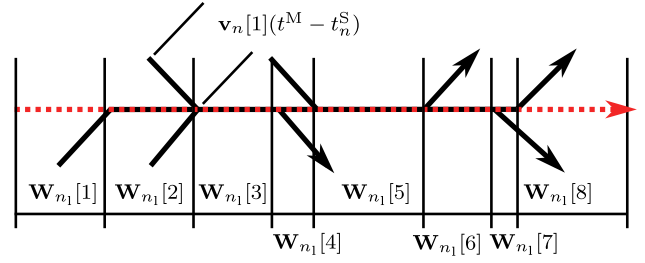
Fig. 4. Illustration of how the sequences $\mathbf{W}_n$ are defined. The red, dotted line represents the route of the coordination leader and the black, solid lines with arrows represent the routes of the coordination followers. The thin lines indicate the distances that the elements of $\mathbf{W}_{n_1}$ correspond to.

Simulations suggest that selecting n in a greedy or a random manner makes little difference for the quality of the computed solution. However, greedy node selection tends to lead to less iterations of the algorithm and is thus better suited for a serial implementation. Random node selection might be preferable for a parallel implementation due to the reduced need for synchronization.

Having computed the set of coordination leaders, there is immediately a vehicle plan for each truck. These plans are jointly optimized as discussed in the following section.

VI. JOINT VEHICLE PLAN OPTIMIZATION

In this section, we derive how to jointly optimize the vehicle plans that are selected by the clustering algorithm. We do this by formulating a convex optimization problem with linear constraints for a group consisting of a coordination leader and its coordination followers. Hereby, the timing when platoons are assembled and broken apart is adjusted while the locations where this happens is not changed. Trucks that are not matched to any coordination leader or are not coordination leaders themselves just follow their default plans and are not considered in this section.

Consider a coordination leader $n_1 \in \mathcal{N}_l$ and its followers $\mathcal{N}_{fl,n_1} = \{n \in \mathcal{N}_c \setminus \mathcal{N}_l : n_1 = \arg \max_{i \in \mathcal{N}_l \cap \mathcal{N}_n^o} \Delta F(n, i)\}$. This group of agents is denoted $\mathcal{N}_g = \{n_1\} \cup \mathcal{N}_{fl,n_1}$. We construct an ordered set of time instances $\mathbf{t} = (\mathbf{t}[1], \mathbf{t}[2], \dots)$. This set contains the start time and the arrival deadline of the coordination leader, and the merge times and the split times of its followers. We divide the distance traveled by the leader from start to destination according to these time instances and get the distances $\mathbf{W}_{n_1}[i] = (\mathbf{t}[i+1] - \mathbf{t}[i])/v_{cd}$ between these points, where v_{cd} is the speed of the leader according to its default plan. These are the distances between the points where coordination followers join or leave the platoon. Similarly, for a coordination follower $n \in \mathcal{N}_{fl,n_1}$, we have $\mathbf{W}_n = (v_n[1](t_n^M - t_n^S),$ $\mathbf{W}_{n_1}[i_n^M], \dots, \mathbf{W}_{n_1}[i_n^{Sp}], v_n[N_{v,n}](t_n^A - t_n^{Sp}))$. The variables $t_n^S, t_n^M, t_n^{Sp}, t_n^A$ denote the start time, merge time, split time, and arrival time of follower n according to its adapted plan. The first element of $\mathbf{W}_n$ is the distance along the route from start to the merge point. For the part of the route the follower platoons with the coordination leader, the entries are the same as for the coordination leader. The indices i_n^M, i_n^{Sp} are defined accordingly. The last element of $\mathbf{W}_n$ is the distance from the

split point to the destination of the follower. Fig. 4 illustrates the definition of $\mathbf{W}_n$.

We introduce sequences $\mathbf{p}_n = (\mathbf{p}_n[1], \ldots, \mathbf{p}_n[|\mathbf{W}_n|])$ that indicate on which segments of the journey the coordination follower is a platoon follower. If truck n is a platoon follower on the segment that corresponds to $\mathbf{W}_n[i]$ for some i, then $\mathbf{p}_n[i] = 1$. Otherwise we have $\mathbf{p}_n[i] = 0$. For the coordination leader n_1, we have $\mathbf{p}_{n_1} = (0, \ldots, 0)$ and for a coordination follower $n \in \mathcal{N}_{\mathrm{fl},n_1}$, we have that $\mathbf{p}_n = (0, 1, 1, \ldots, 1, 0)$.

We express the speed and time sequence of truck $n \in \{n_1\} \cup \mathcal{N}_{\mathrm{fl},n_1}$ as traversal times $\mathbf{T}_n = (\mathbf{T}_n[1], \ldots, \mathbf{T}_n[|\mathbf{W}_n|])$ of the segments $\mathbf{W}_n$. The speed on each such segment remains constant and can be computed as $\mathbf{v}_n[i] = \mathbf{W}_n[i]/\mathbf{T}_n[i]$. The traversal times of the segments in all trucks' routes are the optimization variables. Working with traversal times rather than the sequence of speeds $\mathbf{v}$ allows us to state the optimization problem with linear constraints. The times when the speed changes $\hat{\mathbf{t}}_n$, are computed as $\hat{\mathbf{t}}_n[i] = t_n^S + \sum_{j=1}^{i-1} \mathbf{T}_n[j]$ for $i = 1, \ldots, N_{\mathrm{v},n} + 1$.

With these definitions, we are ready to state the following optimization problem:

Problem 2:

$$\min_{\{\mathbf{T}_n : n \in \mathcal{N}_{\mathrm{g}}\}} \sum_{n \in \mathcal{N}_{\mathrm{g}}} \sum_{i=1}^{N_{\mathrm{v},n}} f\left(\frac{\mathbf{W}_n[i]}{\mathbf{T}_n[i]}, \mathbf{p}_n[i]\right) \mathbf{W}_n[i] \tag{6a}$$

s.t. for $n \in \mathcal{N}_{\mathrm{g}}$:

$$\frac{\mathbf{W}_n[i]}{v_{\max}} \leq \mathbf{T}_n[i], \quad i \in \{1, \ldots, N_{\mathrm{v},n}\} \tag{6b}$$

$$\frac{\mathbf{W}_n[i]}{v_{\min}} \geq \mathbf{T}_n[i], \quad i \in \{1, \ldots, N_{\mathrm{v},n}\} \tag{6c}$$

$$t_n^S + \sum_{i=1}^{N_{\mathrm{v},n}} \mathbf{T}_n[i] \leq t_n^D \tag{6d}$$

and for $n \in \mathcal{N}_{\mathrm{fl},n_1}$:

$$t_n^S + \mathbf{T}_n[1] = t_{n_1}^S + \sum_{i=1}^{i_n^M - 1} \mathbf{T}_{n_1}[i] \tag{6e}$$

$$\mathbf{T}_n[1+i] = \mathbf{T}_{n_1}[i_n^M + i - 1], \quad i \in \{1, \ldots, i_n^{Sp} - i_n^M + 1\}. \tag{6f}$$

The objective function (6a) equals the combined fuel consumption $\sum_{n \in \mathcal{N}_{\mathrm{g}}} F(\phi_n, \pi_n)$ for the assignments $\mathcal{N}_{\mathrm{g}}$ which is part of the sum that defines the combined fuel consumption of all assignments F_{c} defined in (3). It is composed of the fuel consumption of the coordination leader and the coordination followers. The coordination leader is considered to travel alone or take the role as the platoon leader throughout its journey. The coordination followers travel alone on the first and the last segment of their journey. They become platoon followers in-between these segments.

There are two sets of constraints. The first set applies to all trucks and ensures that the sequences $\mathbf{T}_n$ correspond to valid vehicle plans. In particular, the constraints (6b) and (6c) express that the trajectories stay within the allowed range of speed. The constraints (6d) express that all trucks arrive
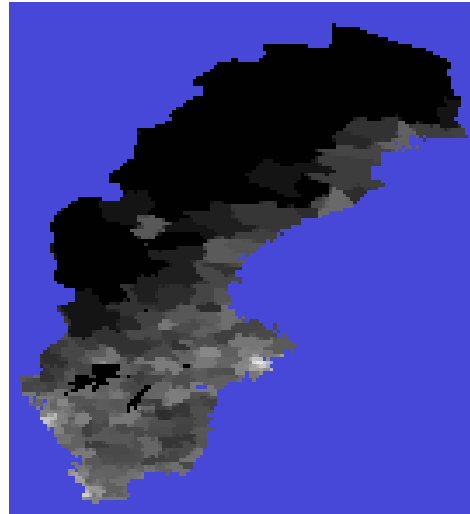

Fig. 5. Population density map from which the start and goal locations are sampled. The brighter the pixel, the larger the population density in that area. Areas not belonging to mainland Sweden are shown in blue.

before their deadline. The second set of constraints ensures that platooning happens as specified in the original pairwise plans. The constraints (6e) ensure that the coordination leader and each of its followers arrive at the same time at their respective merge point. The constraints (6f) ensure that the speed of the leader and the speed of the follower are the same when they are supposed to platoon.

When f_0, f_{p} are such that $f_0(T^{-1})$ and $f_{\mathrm{p}}(T^{-1})$ are convex in T for $T > 0$, then the objective (6a) is a sum of convex functions and hence convex. For instance, polynomials with arbitrary constant part and non-negative coefficients fulfill this requirement. Furthermore, all constraints are linear. Thus, Problem 2 is a convex optimization problem for which well developed numerical solvers are readily available [25], [26]. The optimization is initialized with the pairwise plans.

VII. SIMULATIONS

In this section, we evaluate the coordination method outlined in the previous sections with Monte Carlo simulations. We show that coordination of truck platooning can lead to significant reductions in fuel consumption compared to the current situation where trucks do not platoon, as well as compared to spontaneous platooning where trucks only form platoons if they happen to be in the vicinity of another.

We generate transport assignments randomly. The start and goal locations are sampled within mainland Sweden. The probability of an assignment starting or ending at a particular location is proportional to the population density [27], see Fig. 5. The resolution is 0.1 degrees in longitude and latitude and the road network node that is closest to the sampled coordinate is chosen. We calculate the routes with the Open Source Routing Machine [28]. Assignments for which no route can be found are disregarded. If the route is longer than 400 kilometers, a 400 kilometers long subsection of the route is randomly selected. This is to take into account that merge points too far from the current position should not be considered for coordination since the uncertainty becomes too

Fig. 6. The routes of a platoon coordinator with four coordination followers. The route of the coordination leader in shown in black, the routes of the coordination followers are dashed. The beginning of a route is marked with a star. The merge point of a follower is indicated with an upwards-facing triangle and the split point with a downwards-facing triangle.

large due to traffic, new assignments, and rest periods of the driver. Start locations along the route are considered since we believe that platoon coordination systems will frequently re-plan for assignments that are already en route and suspended for the driver to rest.

The fuel model is an affine approximation around 80 km/h of the analytical fuel model in [29]. We have for the fuel per distance traveled in kilograms diesel per meter

$$f_0(v) = 8.4159 \cdot 10^{-6} v + 4.8021 \cdot 10^{-5}$$
$$f_p(v) = 5.0495 \cdot 10^{-6} v + 8.5426 \cdot 10^{-5}.$$

According to this model, the relative reduction in fuel consumption of a platoon follower is 15.9 percent at a speed of 80 km/h.

We consider a default speed of 80 km/h and we assume that the speed can be freely chosen between $v_{min} = 70$ km/h and $v_{max} = 90$ km/h throughout the entire journey. We sample the start time of the assignments uniformly in an interval of 2 hours and compute the arrival deadlines according to the default speed.

The pairwise plans are such that trucks platoon as long as possible. Once a coordination follower splits up from the coordination leader, it drives fast enough to arrive in time at its destination and at least at default speed. The split points are such that arriving in time is feasible. Thus, trucks are guaranteed to meet their deadlines and the initial value for the joint vehicle plan optimization fulfills the constraints. Fig. 6 shows an example of the routes of a coordination leader and its coordination followers and where the coordination followers join and leave the platoon.

We compare our proposed platoon coordinator to fuel savings that arise from spontaneous platooning, i.e., that trucks happen to get into each others vicinity and then spontaneously form platoons. To this end, we collect all the link arrival times according to the default plans for each link in the scenario.

We sort these times and collect them in ascending order in groups of at most one minute difference in their edge arrival time. We assume that each of these groups forms a platoon driving at default speed and that the default trajectory is not altered by the platooning. This is a generous estimate since it neglects any kind of coordination effort which would be present for time gaps up to one minute.

In order to assess the the quality of the solution computed by the clustering algorithm, we establish an upper bound on the solution of Problem 1. This upper bound is based on the intuition to assign every truck its best coordination leader and ignore that coordination leaders do not contribute to the objective. We have that

$$f_{ce}(\mathcal{N}_1) = \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j)$$
$$\leq \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o} \Delta F(i, j)$$
$$\leq \sum_{i \in \mathcal{N}_c} \max_{j \in \mathcal{N}_i^o} \Delta F(i, j), \tag{7}$$

where the second inequality holds since $\Delta F(i, j) > 0$ for all $(i, j) \in \mathcal{E}$.

This bound can only be tight when there is an optimal solution where no coordination leader has an out-neighbor. Otherwise the coordination leaders cannot contribute to the sum. Nevertheless, the bound helps us assess how far a heuristic solution can be away from the optimum.

We implemented platoon coordination in Python and used CVXOPT [26] for convex optimization. The execution of the clustering algorithm takes less than a second for 2000 transport assignments. Even faster computation times could be achieved by optimizing the implementation.

Each simulation consists of the following steps:
1) Random generation of transport assignments
2) Computation of routes and default plans
3) Computation of the coordination graph
4) Computation of coordination leaders according to Section V
5) Joint vehicle plan optimization according to Section VI

We evaluate how different numbers of assignments affect the amount of platooning and the fuel savings relative to the default plans. For comparison we compute the fuel savings of spontaneous platooning. We run the clustering algorithm with greedy and random node selection and compute the upper bound of the objective function f_{ce}. The results are averaged over 150 simulation runs.

Fig. 7 visualizes an example coordination graph. In addition it shows which assignments are selected in step 4). We can see that only a small fraction of assignment pairs can save fuel by forming a platoon. As the number of assignments grows, more opportunities are available for each assignment which can translate into larger fuel savings [10].

Fig. 8 shows the effect on the fuel savings when the numbers of transport assignments that are coordinated is varied. It is possible to make a number of observations based on these data. First of all, the fuel savings increase rapidly with the number of transport assignments when the absolute number of
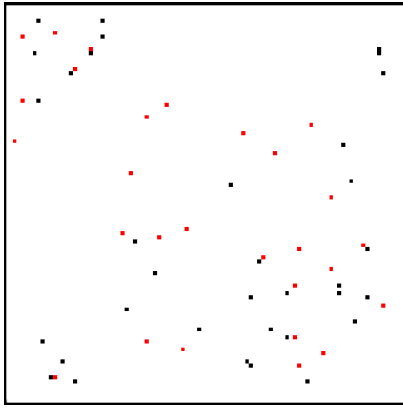
Fig. 7.   This plots visualizes the adjacency matrix of a coordination graphs with 100 assignments. Nonzero entries are indicated with a black or a red dot, each corresponding to an edge in the coordination graph. Edges whose corresponding plans are selected by the clustering algorithm correspond to the red dots.
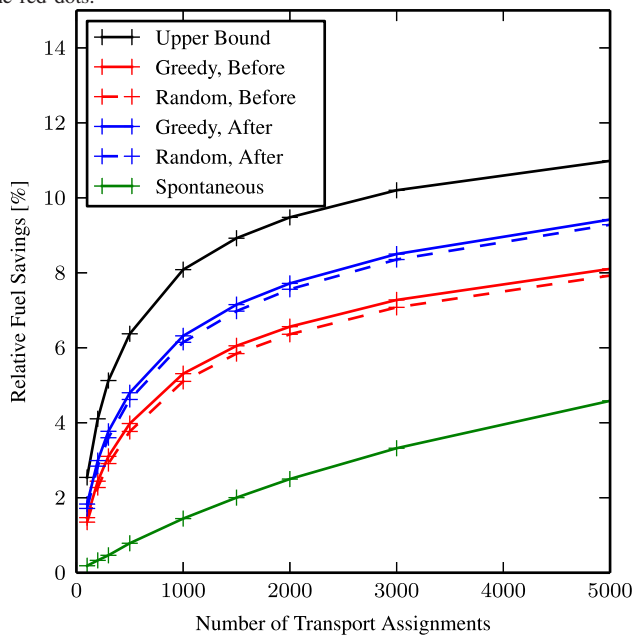

Fig. 8.   The relative fuel savings due to platooning compared to the default plans with varying numbers of assignments. "Greedy" indicates that greedy node selection was used in the clustering algorithm, whereas "Random" indicates random node selection. The keywords "Before"/"After" refer to the relative fuel savings before/after the joint optimization of the vehicle plans. "Spontaneous" are the relative fuel savings based on the estimate of fuel savings due to spontaneous platooning. "Upper Bound" refers to the upper bound on f_{ce} as stated in (7).
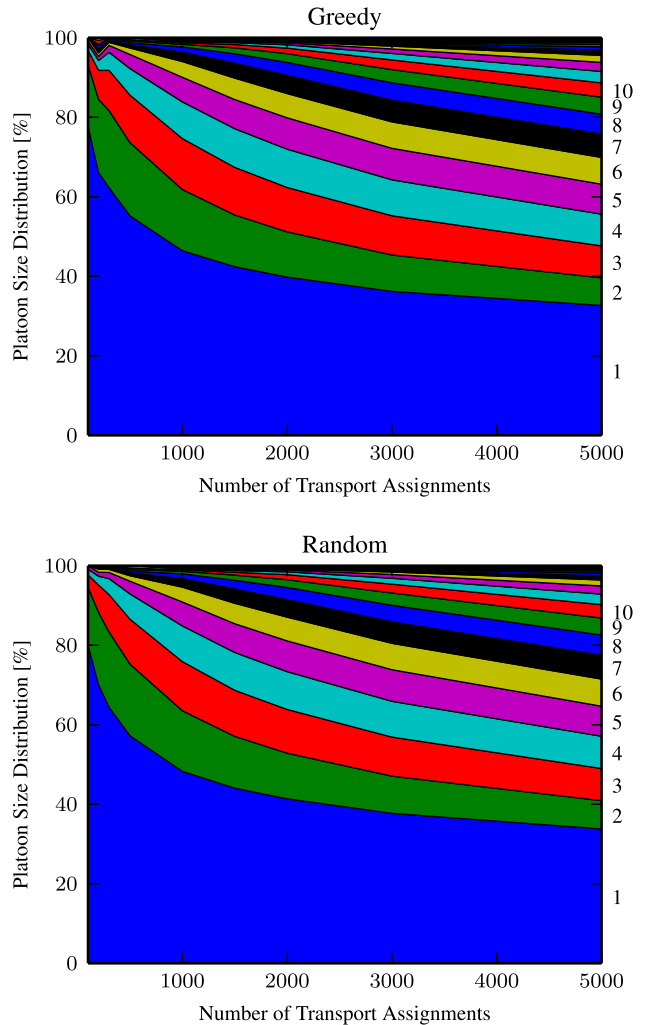

Fig. 9.   This figure shows the distribution of platoon sizes per distance traveled over the number of assignments in percent. The upper plot shows the results of greedy node selection whereas the lower plot shows those of random node selection in the clustering algorithm. To the right, the size of platoon is indicated for a platoon size up to ten. So, when the difference between the first and the second boundary from below is for instance at 20%, it means that 20% of the distance was traveled as member of a platoon of size 2.

assignments is small. As more and more assignments are added, this trend stagnates and the relative fuel savings increase only slowly. Ideally this should approach asymptotically the maximum fuel savings of 15.9% as the number of transport assignments goes to infinity, since then virtually every truck is a platoon follower for its entire journey. There is only a small difference between greedy and random node selection, however, with the greedy node selection outperforming the random node selection consistently. For a parallel or even a distributed implementation of the clustering algorithm, random node selection would be preferable due to the reduced need for synchronization whereas greedy node selection is faster in a centralized setting. Furthermore, the results after selecting the coordination leaders and before the joint convex

optimization are less than the upper bound but only about 30% worse. Since the upper bound is not tight, this indicates that the clustering algorithm performs well. We can see a clear improvement in the fuel savings by the joint optimization of the vehicle plans. Spontaneous platooning gives fuel savings that are less than half of what can be achieved by coordination. Also bear in mind that this is a generous estimate of fuel savings by spontaneous platooning so that the real difference would probably be even larger.

We conclude that coordinated platooning can yield significant fuel savings and that coordination is crucial leveraging these savings. For 2000 transport assignments starting over the course of two hours, we get 7.6% reduction in fuel consumption. A number of 2000 trucks starting in that time interval on an area like Sweden is a realistic number. The total distance traveled in the simulated scenario is in the same order of magnitude as the total distance traveled by domestic road freight transport in Sweden within two hours, assuming that traffic volume is equally spread over the year [30]. The
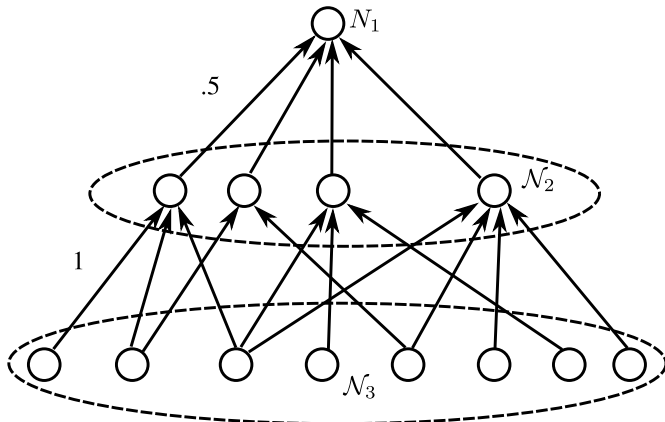
Fig. 10.   Illustration of the graph used to prove that Problem 1 is NP-hard.

density of the road freight traffic that was simulated is only a fraction of the total road freight traffic in countries with high population density. The small fuel savings for the platoon leaders, which have been neglected, would further increase the platooning benefit.

Fig. 9 shows how the distribution of platoon sizes changes with the number of transport assignments. We can see that the larger the number of transport assignments, the more distance is traveled in large platoons. For 2000 assignments over half the distance traveled is in a platoon. Most of the distance is traveled in platoons with ten or less vehicles. This is promising since large platoons might be difficult to control and thus the platoon coordinator would have to prevent planning for larger platoons. Since these large platoons only account for a small fraction of the distance traveled, this would not have too large an impact on the total fuel savings. The largest platoon formed has 28 vehicles. A noticeable effect occurs at a number of 200 transport assignments when more distance is traveled in relatively large platoons compared to the distribution with a number of 300 transport assignments. It seems that some kind of phase transition occurs at these points, where enough assignments are in the system to go from one coordination leader with many followers to having several coordination leaders that are better suited for their followers. To understand this phenomenon is subject of future work.

The simulations show that computing plans for a large number of vehicles to form platoons is feasible with the methods outlined in this paper. It motivates that real-time platoon coordination enables significant reductions in fuel consumption and might be the key to leveraging the full potential of truck platooning.

VIII. Conclusion

A centralized truck platoon coordinator was proposed. The system provides trucks with vehicle plans that lead to reduced fuel consumption by making use of platooning. As time evolves, plans are updated to account for deviations and new assignments. In order to handle the complexity of such coordination, the problem was formulated in a way that for each truck a number of plans adapted to the default plans of other vehicles are computed. Each adapted plan involves platooning for some distance as platoon follower and thus saving fuel. It was

derived how these plans should be systematically combined in order to maximize the total fuel savings. The NP-hardness of this problem motivates the proposed heuristic solution method. Furthermore, we derived how to jointly optimize the vehicle plans resulting from the combination of default plans and adapted plans. The effectiveness of the method was demonstrated in a realistic simulation study. The simulations motivate that such systems should be deployed once trucks with the ability to platoon are commercially available.

There are various directions of future work. One direction is to understand how the transport assignments and the road network relate to fuel savings achieved by this method. Furthermore, we want to study a receding horizon implementation of the platoon coordinator under the presence of disturbances. Another direction is to study the system in a setting where more practical details such as speed limits, traffic, driver rest times, different vehicle types etc. are taken into account. Finally, similar coordination strategies might be relevant for other types of multi-agent systems. This work was carried out in the scope of the COMPANION EU project [31] and the proposed platoon coordinator is being implemented in a demonstrator featuring simulated and real trucks.

Appendix
Proof of Proposition 1

We show the result by reduction of the optimization version of the set covering problem to Problem 1. The optimization version of the set covering problem is well known to be NP-hard. Reduction to a known hard problem is a common proof technique for this kind of result [24]. We do this by constructing a coordination graph $\mathcal{G}_c$ for which there is a one-to-one correspondence between coordination leaders and selected sets for the cover. Then we show that the minimum number of leaders that corresponds to a set cover gives the maximum value for f_{ce}.

Consider the following set covering problem. We have a finite set $\mathcal{U}$. Furthermore, let $\mathcal{S}_u$ be a family of subsets of $\mathcal{U}$ with $\bigcup_{\mathcal{S} \in \mathcal{S}_u} \mathcal{S} = \mathcal{U}$. The problem is to find the smallest number of subsets in $\mathcal{S}_u$ whose union is $\mathcal{U}$.

We construct the coordination graph as the one shown in Fig. 10. We introduce a node for each element in $\mathcal{U}$. We denote the set of these nodes with $\mathcal{N}_3$ and let $\mu_3 : \mathcal{U} \to \mathcal{N}_3$ be a bijective mapping from the elements in $\mathcal{U}$ to the nodes in $\mathcal{N}_3$. We introduce a node for each element in $\mathcal{S}_u$. We denote the set of these nodes with $\mathcal{N}_2$ and let $\mu_2 : \mathcal{S}_u \to \mathcal{N}_2$ be a bijective mapping from elements in $\mathcal{S}_u$ to nodes in $\mathcal{N}_2$. Consider a node $n_2 \in \mathcal{N}_2$ that corresponds to the element $\mathcal{S} \in \mathcal{S}_u$. The in-neighbors of n_2 are $\mathcal{N}_{n_2}^{i} = \{\mu_3(\mathcal{S}) : \mathcal{S} \in \mu_2^{-1}(n_2)\}$. The weight of the corresponding edges is 1. We introduce an additional node N_1. There is an edge from each node in $\mathcal{N}_2$ to N_1 with weight 0.5. Clearly, this reduction is linear in the size of the input $\mathcal{U}, \mathcal{S}_u$.

Since N_1 has no out-neighbors, its membership in $\mathcal{N}_1$ can only increase $f_{ce}(\mathcal{N}_1)$. Since all nodes in $\mathcal{N}_3$ have no in-neighbors, adding a node in $\mathcal{N}_3$ to $\mathcal{N}_1$ can only decrease $f_{ce}(\mathcal{N}_1)$. Thus, the problem of finding the optimal $\mathcal{N}_1$ reduces to finding which nodes in $\mathcal{N}_2$ belong to $\mathcal{N}_1$. In the optimal solution, each node in $\mathcal{N}_3$ has at least one out-neighbor in

$\mathcal{N}_1$. Otherwise we could add any out-neighbor of that node to $\mathcal{N}_1$ and increase $f_{ce}(\mathcal{N}_1)$ by at least 0.5. Therefore, $\{\mu_2^{-1}(n) : n \in \mathcal{N}_1 \cap \mathcal{N}_2\}$ is a set cover of $\mathcal{U}$. Otherwise there would be $u \in \mathcal{U}$ such that there is no $\mathcal{S} \in \{\mu_2^{-1}(n) : n \in \mathcal{N}_1 \cap \mathcal{N}_2\}$ with $u \in \mathcal{S}$. If such a u existed, $\mu_3(u)$ would be a node with no out-neighbor in $\mathcal{N}_1 \cap \mathcal{N}_2$. Furthermore let $\bar{\mathcal{S}}_u \subseteq \mathcal{S}_u$ be a set cover of $\mathcal{U}$. Then $\{\mu_2(\mathcal{S}) : \mathcal{S} \in \bar{\mathcal{S}}_u\}$ has the property that

$$\bigcup_{n \in \{\mu_2(\mathcal{S}) : \mathcal{S} \in \bar{\mathcal{S}}_u\}} \mathcal{N}_n^i = \mathcal{N}_3,$$ so any set cover has the property

that all nodes in $\mathcal{N}_3$ have at least one out-neighbor in $\mathcal{N}_1$. Each node in $\mathcal{N}_2$ contributes with 0.5 to the objective if it is not in $\mathcal{N}_1$. Therefore, the optimal $\mathcal{N}_1$ contains a minimum number of nodes from $\mathcal{N}_2$ such that every node in $\mathcal{N}_3$ has at least one out-neighbor in $\mathcal{N}_1 \cap \mathcal{N}_2$. Since any $\mathcal{N}_1 \cap \mathcal{N}_2$ that fulfills this property maps to a set cover $\bar{\mathcal{S}}_u$ and vice versa, and since $|\mathcal{N}_1 \cap \mathcal{N}_2| = |\bar{\mathcal{S}}_u|$, we have that $\bar{\mathcal{S}}_u$ is the solution to the set covering problem. Thus the NP-hard set-covering problem can be reduced to Problem 1, which shows that Problem 1 is NP-hard.

References

[1] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *Proc. IEEE*, vol. 88, no. 7, pp. 913–925, Jul. 2000.

[2] C. Bonnet and H. Fritz, "Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing," SAE, Warrendale, PA, USA, Tech. Rep. 2000-01-3056, 2000.

[3] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of Class 8 vehicles over a range of speeds, following distances, and mass," *SAE Int. J. Commercial Vehicles*, vol. 7, no. 2, pp. 626–639, Sep. 2014.

[4] S. Tsugawa, "An overview on an automated truck platoon within the energy ITS project," *IFAC Proc. Vols*, vol. 46, no. 21, pp. 41–46, 2013.

[5] T. G. Crainic and G. Laporte, "Planning models for freight transportation," *Eur. J. Oper. Res.*, vol. 97, no. 3, pp. 409–438, 1997.

[6] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu, "Eco-routing navigation system based on multisource historical and real-time traffic information," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1694–1704, Dec. 2012.

[7] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proc. IEEE*, vol. 91, no. 12, pp. 2043–2067, Dec. 2003.

[8] E. Larsson, G. Sennton, and J. Larson, "The vehicle platooning problem: Computational complexity and heuristics," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 258–277, Nov. 2015.

[9] 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 *Proc. 7th IFAC Symp. Adv. Autom. Control*, 2013, pp. 738–743.

[10] J. Larson, K.-Y. Liang, and K. H. Johansson, "A distributed framework for coordinated heavy-duty vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 419–429, Feb. 2015.

[11] P. Meisen *et al.*, "A data-mining technique for the planning and organization of truck platoons," in *Proc. Int. Conf. Heavy Veh., Heavy Veh. Transp. Technol.*, 2008, pp. 389–402.

[12] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, "Fuel-optimal centralized coordination of truck platooning based on shortest paths," in *Proc. Amer. Control Conf.*, 2015, pp. 3740–3745.

[13] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, "Coordinating truck platooning by clustering pairwise fuel-optimal plans," in *Proc. 18th IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 408–415.

[14] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 2008.

[15] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.

[16] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[17] V. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech., Theory Experim.*, vol. 2008, p. P10008, Oct. 2008.

[18] S. Harenberg *et al.*, "Community detection in large-scale networks: A survey and empirical evaluation," *Comput. Statist.*, vol. 6, no. 6, pp. 426–439, 2014.

[19] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[20] H. Garcia-Molina, "Elections in a distributed computing system," *IEEE Trans. Comput.*, vol. C-31, no. 1, pp. 48–59, Jan. 1982.

[21] S. Singh and J. Kurose, "Electing 'good' leaders," *J. Parallel Distrib. Comput.*, vol. 21, no. 2, pp. 184–201, 1994.

[22] E. Demir, T. Bektaş, and G. Laporte, "A review of recent research on green road freight transportation," *Eur. J. Oper. Res.*, vol. 237, no. 3, pp. 775–793, 2014.

[23] P. Sanders and D. Schultes, "Engineering fast route planning algorithms," in *Proc. 6th Int. Workshop Experim. Algorithms*, Rome, Italy, Jun. 2007, pp. 23–36.

[24] T. H. Cormen *et al.*, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[26] M. S. Andersen *et al.* (2013). *CVXOPT: A Python Package for Convex Optimization*. [Online]. Available: https://cvxopt.org

[27] *Population Density Grid, V3, 2000*, NASA, NASA Socioecon. Data and Appl. Center (SEDAC), Palisades, NY, USA, 2015. [Online]. Available: http://sedac.ciesin.columbia.edu/citations

[28] D. Luxen and C. Vetter, "Real-time routing with OpenStreetMap data," in *Proc. 19th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2011, pp. 513–516.

[29] B. Besselink *et al.*, "Cyber–physical control of road freight transport," *Proc. IEEE*, vol. 104, no. 5, pp. 1128–1141, May 2016.

[30] B. Saxon, Ed., "Swedish national and international road goods transport 2014," Transp. Anal. Stockholm, Sweden, Tech. Rep. 2015:21, 2015.

[31] S. Eilers *et al.*, "COMPANION—Towards co-operative platoon management of heavy-duty vehicles," in *Proc. 18th IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 1267–1273.

Sebastian van de Hoef received the B.Sc. degree in mechatronics from Technische Universität Darmstadt in 2011 and the M.Sc. degree in systems, control, and robotics from KTH Royal Institute of Technology, Stockholm, Sweden, in 2013, where he is currently working toward the Ph.D. degree with the ACCESS Linnaeus Centre, Department of Automatic Control. His research interests include dynamic coordination of distributed cyber-physical systems with a focus on coordination of truck platooning.

Karl Henrik Johansson (S'92–M'98–SM'08–F'13) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University. He has held visiting positions with UC Berkeley, Caltech, NTU, the HKUST Institute of Advanced Studies, and NTNU. He is currently the Director of the Stockholm Strategic Research Area ICT The Next Generation and a Professor with the School of Electrical Engineering, KTH Royal Institute of Technology. His research interests are in networked control systems, cyber-physical systems, and applications in transportation, energy, and automation. He is a member of the IEEE Control Systems Society Board of Governors and the European Control Association Council. He has received several best paper awards and other distinctions, including a ten-year Wallenberg Scholar Grant, a Senior Researcher Position with the Swedish Research Council, the Future Research Leader Award from the Swedish Foundation for Strategic Research, and the triennial Young Author Prize from IFAC. He is the IEEE Distinguished Lecturer.

Dimos V. Dimarogonas (M'10) was born in Athens, Greece, in 1978. He received the Diploma degree in electrical and computer engineering and the Ph.D. degree in mechanical engineering from National Technical University of Athens, Athens, in 2001 and 2007, respectively. Between 2007 and 2009, he was a Post-Doctoral Researcher with the Automatic Control Laboratory, School of Electrical Engineering, ACCESS Linnaeus Centre, KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently an Associate Professor. Between 2009 and 2010, he was a Post-Doctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Boston. He was a Docent in automatic control with KTH Royal Institute of Technology in 2012. His research interests include multiagent systems, hybrid systems and control, robot navigation, and networked control. He is a member of the Technical Chamber of Greece. He serves in the Editorial Board of *Automatica*, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and *IET Control Theory and Applications*.