Spatial and Temporal Synchronization of Truck Platoons

Anirudh Kishore Bhoopalam*[1], Niels Agatz[1], and Rob Zuidwijk[1]

[1]Rotterdam School of Management, Erasmus University

Abstract

Truck platooning technology allows trucks to drive at short headways to save fuel and associated emissions. However, fuel savings from platooning are relatively small so forming platoons should be convenient and with minimum detour and delays. In this paper, we focus on developing optimization technology to match trucks into platoons. We formulate a mathematical program for the platoon routing problem with time windows (PRP-TW) based on a time-space network. We provide polynomial time algorithms to solve special cases of the PRP-TW with 2-truck platoons. Based on these special cases, we build several fast heuristics for the PRP-TW and show that these heuristics perform well. Moreover, we show that simple 2-truck platoons already capture most of the potential savings of platooning.

1 Introduction

Truck platooning technology allows trucks to virtually connect and automatically drive in close formation. The short headways decrease air drag and thereby fuel consumption. By saving fuel, platooning can contribute both to costs savings and lower vehicle emissions (Scora and Barth 2006). Other benefits of platooning include improved road capacity utilization and traffic throughput (Schladover et al. 2015, Van Arem et al. 2006, Lioris et al. 2017). Moreover, as the following trucks break and steer automatically with the leader truck in front, there is a lower risk of rear-end collisions. Additional cost savings may be possible in the future when the following trucks in a platoon can operate without drivers (Kilcarr 2016) or the drivers in the following trucks can take breaks to comply with legal work-time restrictions.

Truck manufacturers across the world have conducted real-world tests with truck platooning (Bhoopalam et al. 2018). So far, most of these tests focus primarily on technological challenges related to making the trucks drive in tandem (Bergenhem et al. 2012). Equally important are the challenges related to the planning and execution of platoons. Careful planning is especially

*Corresponding author: kishorebhoopalam@rsm.nl

important when the truck density is low or when the density of trucks with compatible platooning technology is low. In such cases, it is unlikely that we can form platoons spontaneously and achieve comparable benefits as when these platoons would be planned in advance (Sokolov et al. 2017, Liang et al. 2014). Another reason that careful planning is important is that the potential fuel savings of platooning are relatively small, i.e., between 4 percent and 13 percent (McAuliffe et al. 2018).

The planning of platoons involves determining the paths and time schedules of a set of trucks, each with a single origin and destination, with the objective to minimize the system-wide travel costs. We refer to the associated planning problem as the Platoon Routing Problem with Time Windows or the PRP-TW. We define a *platoon* as a set of two or more virtually connected trucks that drive together at short headways. Legal restrictions will limit the maximum number of trucks in a platoon. For example, the maximum platoon size was limited to two for tests in Kentucky and Missouri (Goble 2018). The reason for this is that larger platoons may disrupt traffic flows at highway on- and off-ramps (Calvert et al. 2019, van Maarseveen 2017).

One truck can potentially join different platoons along its journey. This creates inter-dependencies in time and space between different platoons. These inter-dependencies make the planning and coordinating of platoons more challenging. In fact, Larsson et al. (2015) show that these inter-dependencies make the Platoon Routing Problem NP-hard, even without deadlines and when all trucks start at the same origin.

In this paper, we model and solve the platoon routing problem with time windows (PRP-TW) but will specifically focus on finding fast solution methods for problem settings in which two trucks platoon together and all trucks are in at most one platoon per trip. We refer to the planning problem that corresponds to this special case as the *Platoon Pair Routing problem with time windows (PPRP-TW)*. Here, we also look at the case without time windows or the PPRP to gain further insights. As the PPRP-TW setting only requires coordination between two trucks, the issue of inter-dependencies is smaller and we therefore believe that these platoons are easier to implement in practice and more robust in the face of travel time variability. We evaluate the effectiveness of these smaller platoons.

The main contributions of this paper can be summarized as follows. First, we present polynomial time approaches to determine the routes and time schedules of two-truck platoons by studying both the PPRP-TW and the PPRP. Second, we introduce a novel mathematical programming formulation for the PRP-TW that involves larger platoons and multiple platoons per trip. Third, we present several fast heuristics based on the polynomial time approaches for the PPRP that provide high-quality solutions. Finally, we conduct numerical experiments to evaluate the advantages and disadvantages of restricting the number of platoons trucks can join on their journeys and the number of trucks per platoon.

The remainder of the paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we define the PRP-TW and provide some useful properties. Section 4 presents a Mixed Integer Programming formulation for the PRP-TW. In Section 5, we focus on the PPRP-TW. Section 6 presents two heuristics and an improvement procedure for the PRP-TW based on the insights from the PPRP-TW and the PPRP. Section 7 presents the results of our numerical experiments. Finally, Section 8 summarizes our conclusions and outlines directions for future research.

2 Related literature

There has been growing interest in studying the planning and operations pertaining to the application of automated driving technology not just on roads (Mahmassani 2016), but also on water (Colling and Hekkenberg 2020), air (Richert and Cortés 2013), and even a mix of roads and air, for example, with drones (Poikonen and Golden 2020, Poikonen et al. 2019, Otto et al. 2018). In this section, we restrict ourselves to platooning and discuss the most relevant and recent literature on truck platoon planning. Bhoopalam et al. (2018) provide a more extensive overview.

A number of papers focus on platoon planning in which trucks travel on a fixed path, thus ignoring any possible routing flexibility. These papers typically focus on a single highway lane or designated hub (Adler et al. 2016, Larsen et al. 2019). A number of these specifically focus on adjusting speeds to form platoons (Liang et al. 2013, Sun and Yin 2019). Others have focused more on different platooning cost functions (Boysen et al. 2018) or travel time uncertainty (Zhang et al. 2017) or dynamic planning (Van de Hoef 2018).

In line with our paper, others do incorporate the route planning of the trucks. Larson et al. (2013) set a threshold for the detour a truck can make. They consider a traffic junction and evaluate whether it would be beneficial for approaching trucks to adjust their speed(s) to form a platoon together. They propose a pairwise merging heuristic in which they continuously merge the most beneficial platoon pairs to generate multi-truck platoons and show that the platooning savings increase when trucks can make a detour.

Larsson et al. (2015) also use such a pairwise heuristic for a centralized setting on a network. Here, they ignore deadlines or time constraints. In addition to this pairwise heuristic, they also present a hub heuristic which groups trucks based on their likelihood of traversing a certain edge. They formulate a MIP to evaluate their heuristics. In their subsequent work, Larson et al. (2016) focus on reducing the problem size by defining additional constraints on aspects such as the detour, which they use to set certain decision variables to zero. As a next step, Luo et al. (2018) extend the model from Larson et al. (2016) by allowing trucks to have different speeds on different edges. Nourmohammadzadeh and Hartmann (2016) also formulate a MIP which they use to evaluate a genetic algorithm based meta-heuristic, and then an ant-colony optimization meta-heuristic (Nourmohammadzadeh and Hartmann 2019).

Albinski et al. (2020) look at the effect of introducing regulations on driver rest times. They formulate a MIP and present a pre-processing procedure to reduce the problem size. They consider different levels of driver involvement and notice that the platoon savings vary greatly.

None of these studies explicitly focus on simple 2-truck platoons with flexible routes and how they compare with larger platoons. As mentioned earlier, these 2-truck platoons have multiple benefits and we focus our attention toward evaluating them in this paper.

3 Problem definition and properties

We now define the PRP-TW. Following this, we describe a couple of theoretical properties relevant to the PRP-TW.

3.1 Problem definition

We model the PRP-TW on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with non-negative arc lengths in which the costs of traversing arc $a \in \mathcal{A}$ connecting nodes u and v is c_a or c_{uv}. Consider a set of trucks $\mathcal{K}$ in which truck $k \in \mathcal{K}$ has origin $o_k \in \mathcal{V}$ and destination $d_k \in \mathcal{V}$. A truck can save (fuel) costs by driving together in a platoon with one or more other trucks. Let $\rho \in (0.5, 1)$ be the average fuel consumption factor for a truck in a platoon. The fuel costs for a single truck driving in a platoon on arc a are ρc_a, where $1 - \rho$ represents the fuel savings for a truck in a platoon as compared to driving alone. Note that ρ is greater than 0.5 as the total combined fuel consumption of a 2-truck (or larger) platoon cannot be less than the fuel consumption of a truck driving alone. As is common in the literature (see for example Nourmohammadzadeh and Hartmann (2016), Larsson et al. (2015)), we consider the costs savings to be independent of its position in the platoon. Furthermore, we assume that trucks in a platoon share the savings equally. Therefore, the parameter ρ captures the average savings of each truck in the platoon and abstracts away from different savings for leaders and followers. The total combined cost of a n-truck platoon traversing a is $c_a^n = n \cdot \rho \cdot c_a$.

Each truck specifies an earliest departure time a_k from the origin and a latest arrival time l_k at the destination. Each truck travels at the same constant speed. This reflects the fact that trucks often drive at approximately the maximum allowable speed on a road segment and can not safely or legally drive faster or slower. Let t_a be the travel time needed to traverse arc a. Let the latest departure time $\bar{a}_k$ denote the latest time that the truck can leave the origin to reach the destination in time. The time flexibility f_k of truck $k \in \mathcal{K}$ is the difference between the earliest and latest departure time or $f_k = l_k - a_k - t_{o_k d_k}$. This flexibility specifies the maximum possible waiting and detour time within a trip.

We assume that trucks can not stop and wait along an edge, that is, trucks can only wait at nodes. This, along with our assumption that trucks travel at the same constant speed means that platoons are only formed (or dissolved) at nodes.

Given a set of trucks, the truck platooning problem involves finding the path and time schedule of all trucks so as minimizing costs to arrive at their destinations before their deadlines. Note that we only consider which trucks form a platoon and not focus on which truck will act as the leader and follower in the platoon.

We can restrict the PRP-TW to allow at most $N \in \mathbb{N}_{\geq 2}$ trucks per platoon and at most $M \in \mathbb{N}$ platoons per truck trip. We use (N, M) to denote a setting with these constraints.

3.2 Basic properties

In this section, we outline a few useful properties of feasible and optimal truck platoons. First, we can eliminate meeting points and split points for which the detour would be too long to offset the potential savings.

Property 1 *Given fuel consumption factor ρ, a set of trucks $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| > 1$ can feasibly form a platoon together at meeting point m only if $\rho < \dfrac{\sum_{k \in \mathcal{K}'} c_{o_k d_k} - \sum_{k \in \mathcal{K}'} c_{o_k m}}{\sum_{k \in \mathcal{K}'} c_{m d_k}}$.*

Proof. Consider the case where the trucks all meet at a single m. In this case, their platooning costs will at least be $\sum\limits_k c_{o_k m} + \rho \cdot \sum\limits_k c_{m d_k}$. For this platooning solution to be beneficial, we need

$$\sum_k c_{o_k m} + \rho \cdot \sum_k c_{m d_k} < \sum_k c_{o_k d_k}$$

$$\rho \sum_k c_{m d_k} < \sum_k c_{o_k d_k} - \sum_k c_{o_k m}$$

$$\rho < \frac{\sum_k c_{o_k d_k} - \sum_k c_{o_k m}}{\sum_k c_{m d_k}} \Box$$

Note that a similar property holds for split point s: $\rho < \dfrac{\sum_k c_{o_k d_k} - \sum_k c_{s d_k}}{\sum_k c_{o_k s}}$.

Second, we can eliminate some platoons due to incompatibility of the earliest departure and latest arrival time of the different trucks.

Property 2 *A set of trucks $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| > 1$ can never be in a platoon together if the* latest *earliest departure time of the trucks is later than the* earliest *latest arrival time of the trucks, i.e., $\max\limits_{k \in \mathcal{K}'} a_k > \min\limits_{k \in \mathcal{K}'} l_k$.*

Proof. Consider a set of trucks $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| > 1$ that travel together in a platoon at time t. By definition $\max\limits_{k \in \mathcal{K}'} a_k \leq t \leq \min\limits_{k \in \mathcal{K}'} l_k$. This provides a contradiction if $\max\limits_{k \in \mathcal{K}'} a_k > \min\limits_{k \in \mathcal{K}'} l_k$. $\Box$

Finally, when the maximum platoon size is greater than two, it is always possible to make platoons for all trucks that jointly traverse an arc. Moreover, for certain cost structures, the total system costs will remain the same no matter how the trucks are allocated to platoons.

Property 3 *If the cost savings per truck do not depend on the size of the platoon and the maximum platoon size $N \geq 3$, all trucks that jointly traverse an arc will be part of a platoon in the optimal solution*

Proof. Consider k trucks that jointly traversing an arc a. If k is an even number, trucks can be matched into $\frac{k}{2}$ 2-truck platoons. If k is an odd number, the trucks may be matched into $\frac{k-3}{2}$ 2-truck platoons and one 3-truck platoon. Therefore, trucks jointly traversing arcs can always be organized into platoons, which brings down total costs as compared to some of them driving alone. Therefore, in an optimal solution all these trucks will be in a platoon. □

4 Mixed Integer Programming formulation

We model the PRP-TW on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ by decomposing the problem into two phases. In the first phase, we focus on our primary goal of minimizing system costs by finding the minimum-cost paths of all trucks in the system. In the second phase, we find the specific platoons associated with these paths.

Phase 1 - Finding paths of trucks and platoons

The PRP-TW involves the synchronization of truck routes in both space and time. To capture the spatial dimension, we create multi-graph $\mathcal{G}_{\mathcal{S}} = (\mathcal{V}, \mathcal{A}_{\mathcal{S}})$ in which any pair of connected nodes $(u, v) \in \mathcal{G}$ is now connected by a set of parallel arcs of two types: (i) *solo* arcs and (ii) n-platoon arcs ($n > 1$). This captures the fact that trucks can travel between two nodes $(u, v) \in \mathcal{G}$ either alone or as part of a platoon of a specific size, each with a different cost.

To model time, we further modify $\mathcal{G}_{\mathcal{S}}$ into a time-expanded network $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}} \cup \mathcal{H}_{\mathcal{T}})$ by discretizing the problem into time periods of equal length $\triangle$, meaning we have a set of time points $\mathcal{T} = \{0, \triangle, 2\triangle, 3\triangle, ..., H\triangle\}$. Note that the total number of time periods H is dictated by the latest l_k for truck $k \in \mathcal{K}$ as $H = \lceil \frac{max_{k\in\mathcal{K}} l_k}{\triangle} \rceil$. The time-expanded network $\mathcal{G}_{\mathcal{T}}$ therefore has a copy u_i of each node $u \in \mathcal{V}$ for each of the time points $i \in \mathcal{T}$ and so $|\mathcal{V}_{\mathcal{T}}| = (H + 1) \cdot |\mathcal{V}|$.

Like Boland et al. (2017), we split the arc set into two sets $\mathcal{A}_{\mathcal{T}}$ and $\mathcal{H}_{\mathcal{T}}$ for ease of explanation. An arc $a \in \mathcal{A}_{\mathcal{T}}$ connects two nodes $u_i, v_j \in \mathcal{V}_{\mathcal{T}}$ such that $j = i + \lceil t_{uv}/\triangle \rceil$. Travel along such an arc represents movement in both space and time. An arc $a \in \mathcal{H}_{\mathcal{T}}$ connects two nodes $u_{i\triangle}, u_{(i+1)\triangle} \in \mathcal{V}_{\mathcal{T}}$ which represents movement only in time, i.e., the truck waits.

Trucks only incur costs when they travel spatially, that is, there are no waiting costs. Hence, the cost of traveling on a *solo* arc a is c_a. Let binary variable x_{ak}^{1t} be 1 if truck k begins traversing solo arc a at t. The corresponding binary variable x_{ak}^{nt} takes a value of 1 if truck k begins traversing n-platoon arc a at t. There are no variable costs associated with traveling on an *n-platoon* arc. There is only a fixed cost c_a^n for each additional *n-platoon*. Let y_a^{nt} denote the maximum number of n-platoons that start traversing arc a at t. The number of trucks is therefore $n \cdot y_a^{nt}$, i.e., it can only increase in steps of n. The maximum value this variable can take is defined by $n \cdot y_a^{nt} \leq |\mathcal{K}|$ where $|\mathcal{K}|$ is the number of trucks in the system. If we fix the value of y_a^{nt}, the model can be characterized as a continuous time service network design problem (Boland et al. 2017)

Each arc a has the same travel time t_a for each truck, irrespective of whether the truck travels solo or in a platoon. We can then formulate a mixed integer program for phase 1 of the PRP-TW. Before doing so, we introduce some additional notation: $\delta^+(v)$ denotes the set of arcs originating from node v and $\delta^-(v)$ denotes the set of arcs ending at node v.

$$\min \quad \sum_{a \in \mathcal{A}_\mathcal{T}} \sum_{t \in \mathcal{T}} (\sum_{k} c_a x_{ak}^{1t} + \sum_{n>1} c_a^n y_a^{nt}) \tag{1}$$

$$\sum_{a \in \delta^+(v)} x_{ak}^{nt} - \sum_{a \in \delta^-(v)} x_{ak}^{n(t+t_a)} = \begin{cases} 1 & (v,t) = (o_k, a_k) \\ -1 & (v,t) = (d_k, l_k) \\ 0 & otherwise \end{cases} \qquad \forall v \in \mathcal{V}, t \in \mathcal{T}, 1 \leq n \leq N, k \in \mathcal{K} \tag{2}$$

$$\sum_{k} x_{ak}^{nt} \leq n \cdot y_a^{nt} \qquad \forall a \in \mathcal{A}_\mathcal{T}, 1 < n \leq N \tag{3}$$

$$x_{ak}^{nt} \in \{0,1\} \qquad \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, k \in H, t \in \mathcal{T}, n \leq N \tag{4}$$

$$y_a^{nt} \in \mathbb{Z}^+ \qquad \forall a \in \mathcal{A}_\mathcal{T}, t \in \mathcal{T}, n \leq N \tag{5}$$

The objective function (1) minimizes the total costs of the system - the first term denoting the variable costs when trucks travel on solo arcs and the second term the fixed costs of taking a platoon arc. Constraints (2) are a set of flow constraints that ensure the timely departure and arrival of trucks at their origins and destinations respectively. Constraints (3) allow sufficient capacity on platoon arcs for trucks to make use of should it lead to the minimization of costs. Constraints (4) and (5) specify the domains of the decision variables.

We can simplify our general formulation to accommodate our specific symmetric platooning cost structure. Recall from Section 3 that we use a cost structure of the form $c_a^n = n \cdot \rho \cdot c_a$ for an n truck platoon. This means that the costs savings of driving in a platoon are not dependent on the position of the truck in the platoon. More generally, any number of trucks greater than two can be matched into platoons of at least size two in multiple ways without changing the total system costs (Property 3). This means we do not need different arcs for different platoon sizes and can merge the different n-platoon arcs into one general platooning arc. Therefore, we use c_a^p to denote the costs for a truck to travel arc a in a platoon *of any size*. Let w_{ak}^t be 1 if truck k begins traveling on solo-arc a by itself at t. Similarly let x_{ak}^t be the corresponding variable if truck k travels on a platoon-arc (of any size). We can then formulate the problem as -

$$\min \quad \sum_{a \in \mathcal{A}_\mathcal{T}} \sum_{t \in \mathcal{T}} (\sum_k c_a w_{ak}^t + \sum_k c_a^p x_{ak}^t) \tag{6}$$

$$\sum_{a \in \delta^+(v)} (w_{ak}^t + x_{ak}^t) - \sum_{a \in \delta^-(v)} (w_{ak}^{t+t_a} + x_{ak}^{t+t_a}) = \begin{cases} 1 & (v,t) = (o_k, a_k) \\ -1 & (v,t) = (d_k, l_k) \\ 0 & otherwise \end{cases} \quad \forall v \in \mathcal{V}, t \in \mathcal{T}, n \leq N, k \in \mathcal{K} \tag{7}$$

$$x_{ai}^t \leq \sum_{k \neq i} x_{ak}^t \qquad \forall a \in \mathcal{A}_\mathcal{T}, i \in \mathcal{K}, t \in \mathcal{T} \tag{8}$$

$$w_{ak}^t \in \{0,1\} \qquad \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, k \in \mathcal{K}, t \in \mathcal{T} \tag{9}$$

$$x_{ak}^t \in \{0,1\} \qquad \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, k \in \mathcal{K}, t \in \mathcal{T} \tag{10}$$

The objective function (6) again seeks to minimize system-wide travel costs. Constraints (7) are again a set of flow constraints that ensure trucks leave their origin and arrive at their destination as per schedule. Constraints (8) ensure that a truck can travel on a platooning arc only if there is at least one other truck travelling on the same arc at the same time. Constraints (9) and (10) specify the domains of the variables.

When the maximum number of trucks allowed in a platoon is two, we need to include an additional constraint to ensure that an individual truck doesn't get routed on a platoon arc when there is, in fact, no room for it, that is, if all trucks on a platooning arc are already in platoons of size 2. We do this by introducing an additional variable to ensure that the number of trucks on a platooning arc stays even as $\sum_k x_{ak}^t = 2 \cdot z_a^t \forall a \in \mathcal{A}_\mathcal{T}, t \in \mathcal{T}$ where z_a^t is an integer variable.

Phase 2 - Finding specific trucks and platoons per path

Phase 1 determines the high-level flow of trucks throughout the network. As part of its output, it specifies the route and time schedules of each truck and whether the truck drives individually or as part of a platoon on each time-space arc along its route. However, it does not provide us with information on the platoon partners of each truck. In phase 2 of our approach, we formulate another MIP to generate specific platoons based on the output of phase 1.

There are many different possible specific platoons that can be created from the solution of stage 1. We aim to find the solution that minimizes the number of platoons in the system. The rationale behind this is that forming platoons is inconvenient for the associated trucks and the surrounding traffic.

Let $\mathcal{K}_a^t = \{i \in \mathcal{K} : x_{ai}^t = 1\}$ be the set of trucks on platoon arc a at time t. The set of feasible platoons on arc a and time t is $\mathcal{S}_a^t := \{\mathcal{S}' \subseteq \mathcal{K}_a^t : 2 \leq |\mathcal{S}'| \leq N\}$. This allows us to define $\mathcal{S}_{a(i)}^t := \{\tilde{\mathcal{S}} \subseteq \mathcal{S}_a^t : i \in \tilde{\mathcal{S}}\}$, which is the set of possible platoons that involve truck i. Let $\mathcal{S} = \cup_{a \in \mathcal{A}, t \in \mathcal{T}} \mathcal{S}_a^t$. Let y_{as}^t be a binary decision variable equal to 1 if platoon formation $s \in \mathcal{S}$ is used on arc a at time t. Similarly, let y_s be a binary decision variable equal to 1 if platoon formation $s \in S$ is used at least once. We can then formulate a mixed integer program as follows.

$$
\begin{aligned}
\min \sum_{s \in \mathcal{S}} \quad y_s & & & \qquad (11)\\
\sum_{a \in \mathcal{A}_\mathcal{T}} \sum_{t \in \mathcal{T}} y_{as}^t \leq M y_s & & \forall s \in \mathcal{S} & \qquad (12)\\
y_{as}^t \leq x_{ai}^t & & \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, t \in \mathcal{T}, s \in \mathcal{S}_{a(i)}^t, i \in \mathcal{K} & \qquad (13)\\
\sum_{s \in \mathcal{S}_{a(i)}^t} y_{as}^t \geq x_{ai}^t & & \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, t \in \mathcal{T} & \qquad (14)\\
y_{as}^t \in \{0, 1\} & & \forall a \in \mathcal{A}_\mathcal{T} \cup \mathcal{H}_\mathcal{T}, t \in \mathcal{T}, s \in \mathcal{S} & \qquad (15)\\
y_s \in \{0, 1\} & & \forall s \in \mathcal{S} & \qquad (16)\\
& & & \qquad (17)
\end{aligned}
$$

The objective function (11) minimizes the total number of platoons. Constraints (12) make sure that y_s is forced to one if trucks in s form a platoon together at any segment of their routes. Constraints (13) ensure that trucks in any s can be together in a platoon on a certain arc only if they all traverse that arc at the same time as determined by the first MIP. Constraints (14) ensure that we satisfy the solution of phase 1. Constraints (15) and (16) specify the domains of the variables.

Note that the stage 2 approach will find the solution with the guaranteed lowest possible inconvenience when stage 1 has a unique solution. This is the case when there are no multiple shortest paths with exactly the same length in the underlying network. In practice, it is unlikely that there will be such equal-length paths. In theory, one can assume without loss of generality that all shortest paths in a graph are unique since ties may be broken by the addition of tiny random variables to arc lengths (Bodwin 2019).

5 Special Cases: Platoon Pair Routing Problems

In this section, we look at special cases of the PRP-TW in which a truck can be part of at most one platoon per trip (M = 1) and platoons have a size of two (N = 2). We call this the *platoon pair routing problem with time windows (PPRP-TW).* Given all possible truck platoon pairs and their respective costs, we can model this problem as a general matching problem. These special cases provide us with insights and building blocks to develop heuristics for the PRP-TW.

Let $\mathcal{G}_\mathcal{P}$ be the *platoon pair graph* in which each node represents a truck in $\mathcal{K}$. In case two trucks $i, j \in \mathcal{K}$ can beneficially form a platoon, then the weight of the edge (i, j) represents the corresponding platoon costs L_{ij}. If a platoon with $i, j \in \mathcal{K}$ is not feasible or beneficial, then L_{ij} is the sum of the two shortest paths between the respective origins and destinations of the two trucks. Therefore, platoon pair graph $\mathcal{G}_\mathcal{P}$ is a complete graph. The problem to minimize the total system costs uses the objective $\min \sum_{i,j} L_{ij}x_{ij}$ with binary variables x_{ij}. To ensure that every truck is matched with a partner or travels alone, we include the constraints $\sum_i x_{ij} \geq 1 \, \forall j \in \mathcal{K}$. Therefore, to create platoon pair graph $\mathcal{G}_\mathcal{P}$, we need to solve a sub-problem to find costs L_{ij} for all $i, j \in \mathcal{K}$. Here, we consider all travel costs proportional to the travel distance. For readability, we ignore the multiplier between distances and costs in the remainder of this section.

Theorem 1 *An optimal solution to the PPRP-TW can be obtained in* $\mathcal{O}\big(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}|+|\mathcal{A}|log|\mathcal{V}|)+|\mathcal{K}|^3\big)$

Proof. Consider a single pair $i, j \in \mathcal{K}$. Consider an arbitrary pair of nodes $u, v \in \mathcal{V}$ to be the meeting point and split point. We can easily determine the costs of this platoon as

$$L_{ij}(u, v) = \sum_{k=1,2} d(o_k, u) + 2 \cdot \rho \cdot d(u, v) + \sum_{k=1,2} d(v, d_k)$$

In the worst case, finding the least-cost time-feasible platoon entails finding all shortest paths between the origins and meeting point u, the meeting point u and the split point v, and the split point v and the destinations. Each shortest path can be found in $\mathcal{O}(|\mathcal{V}|+|\mathcal{A}|log|\mathcal{V}|)$ (Dantzig 1960). Doing this for all $|\mathcal{V}|^2$ pairs and compute the minimum costs L_{ij} therefore requires $\mathcal{O}\big(|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|)\big)$. The next step is to compute the L_{ij} for all pairs $i, j \in \mathcal{K}$. This can be achieved in $\mathcal{O}\big(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|)\big)$. With this, we now have platoon pair graph $\mathcal{G}_\mathcal{P}$. The final step to determining the total costs is finding the minimum weight matching on platoon pair graph $\mathcal{G}_\mathcal{P}$, which is a complete graph with $|\mathcal{K}|$ nodes.This can be accomplished in $\mathcal{O}(|\mathcal{K}|^3)$ (Gabow 1990). The overall complexity is therefore $\mathcal{O}\big(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|) + |\mathcal{K}|^3\big)$. □

Next, we focus on a relaxed version of the PPRP-TW in which the time windows are not binding. We call this the *platoon pair routing problem (PPRP)*. The solution to this problem without time windows is a lower bound to the problem with time windows. In practice, we expect that the optimal solutions of this relaxed problem to usually also be time feasible as they are often associated with small detours.

Therefore, the insights we gain from analyzing the PPRP allow us to reduce complexity for cases in which the time windows are not very restrictive thereby speeding up computations and reducing run times for practical cases with time windows.

5.1 PPRP

Recall from Section 3 that the costs of traveling an edge a between nodes (u, v) alone is c_{uv} and the costs of traveling in a platoon is ρc_{uv} for each truck. We can transform this problem into a shortest path problem that aims to find a single least-costs path between a source and the sink in a transformed graph $\mathcal{G}_{\mathcal{E}} = (\mathcal{V}_{\mathcal{E}}, \mathcal{A}_{\mathcal{E}})$. We build the transformed graph by explicitly modeling the following three parts of a platoon routing solution.

1. the first legs of the trucks from their origins to the meeting point,
2. the joint platooning part
3. the last legs of the trucks to their destinations from the split point.

To model (1), we add a source node S to the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. For each node $v \in \mathcal{V}$ that is a feasible meeting point based on Property 1, we add an edge connecting S and v. The weight of this edge is the sum of the costs for the two trucks to travel from their origins to meeting point v, i.e., $d_{\mathcal{E}}(S, v) = \sum\limits_{k=1,2} d(o_k, v)$. To model (2), we adjust the weights of the edges $(u, v) \in \mathcal{A}$ of the original graph $\mathcal{G}_{\mathcal{E}}$ to account for the fuel consumption factor, i.e., $d_{\mathcal{E}}(u, v) = 2 \cdot \rho \cdot d(u, v)$. To model (3), we add a sink node T to the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. For each node $v \in \mathcal{V}$ that is a feasible split point based on Property 1, we add an edge connecting v and T. The weight of this edge is the sum of the costs for the two trucks to travel from split point v to their destinations, i.e., $d_{\mathcal{E}}(v, T) = \sum\limits_{k=1,2} d(v, d_k)$.

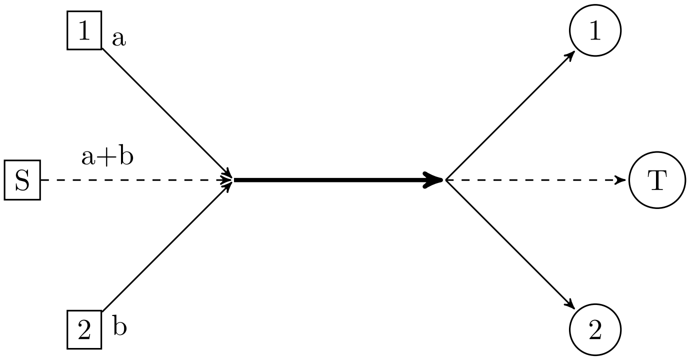


Figure 1: Transforming two paths into a single path

Proposition 1 *PPRP subproblem is equivalent to the shortest path problem in the following sense: any feasible solution to one problem can be transformed to a feasible solution of the other problem while attaining the same objective value.*

Proof. The proposition follows from the fact that

$$L(m, s) = \sum_k d(o_k, m) + 2 \cdot \rho \cdot d(m, s) + \sum_k d(s, d_k) = d_{\mathcal{E}}(S, T).$$

By definition, the distance $d_{\mathcal{E}}(S,T)$ is established by a shortest path in $\mathcal{G}_{\mathcal{E}}$. The first edge in that path (S,m) establishes the meeting point $m \in \mathcal{V}$ and the last edge (s,T) in that path establishes the split point $s \in \mathcal{V}$. □

In the following theorem, we establish the complexity of the PPRP.

Theorem 2 *An optimal solution to the PPRP can be obtained in* $\mathcal{O}\big(|\mathcal{V}||\mathcal{K}|^2(|\mathcal{V}|+|\mathcal{A}|log|\mathcal{V}|)+|\mathcal{K}|^3)\big)$

Proof. Consider a single pair $i,j \in \mathcal{K}$. Adding source S and sink T can be done in constant time. Modeling (1) requires us to add an edge while computing its weight, which we do by finding the shortest paths in $\mathcal{O}(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|)$. In the worst case, this needs to be done for every $v \in \mathcal{V}$ and this takes $\mathcal{O}\big(|\mathcal{V}|(|\mathcal{V}| + \mathcal{A}log|\mathcal{V}|)\big)$. Modeling (2) entails going over each arc $a \in \mathcal{A}$ of the graph G to modify its weight which takes $\mathcal{O}(|\mathcal{A}|)$. Finally, modeling (3) has the same complexity of modeling (1). Therefore the total time complexity to build transformed graph $\mathcal{G}_{\mathcal{E}}$ for a pair $i,j \in \mathcal{K}$ is $\mathcal{O}\big(|\mathcal{V}|(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|)\big)$. Then, we need to find a single shortest path between S and T on transformed graph $\mathcal{G}_{\mathcal{E}}$, which has $(|\mathcal{V}| + 2)$ nodes and $(|\mathcal{A}| + 2|\mathcal{V}|)$ edges. This can therefore be done in $\mathcal{O}(|\mathcal{V}| + (|\mathcal{A}| + |\mathcal{V}|)log|\mathcal{V}|)$. Hence the total time complexity to find the costs L_{ij} for $i,j \in \mathcal{K}$ and therefore build platoon pair graph $\mathcal{G}_{\mathcal{P}}$ is $\mathcal{O}\big(|\mathcal{V}|(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|)\big)$. As before, we can find the minimum weight matching in $\mathcal{O}(|\mathcal{K}|^3)$ making the overall complexity $\mathcal{O}\big(|\mathcal{V}||\mathcal{K}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|) + |\mathcal{K}|^3)\big)$. □

There is a factor $|\mathcal{V}|$ difference in the complexity of the subproblems of the PPRP and the PPRP-TW. This is because we only iterate over every node $u \in \mathcal{V}$ for the PPRP, as opposed to iterating over every node *pair* $u, v \in \mathcal{V}$ for the PPRP-TW.

We now outline how we can use the PPRP subproblem solution method to solve the PPRP-TW. Since the PPRP subproblem solution is a lower bound to the PPRP-TW subproblem, we do not need to solve the PPRP-TW subproblem for truck pairs $i,j \in \mathcal{K}$ that do not have a beneficial PPRP subproblem solution. For those that do, we include them in the platoon pair graph $\mathcal{G}_{\mathcal{P}}$ if they are time-feasible. Only for those pairs that are not time-feasible, we solve the the PPRP-TW subproblem using the MIP from Section 4 (note that we only need phase 1). We can then find the minimum weight matching as earlier. Note that in practice, we can pre-compute or use parallel computing for the different steps to build the platoon pair graph $\mathcal{G}_{\mathcal{P}}$.

5.2 PPRP on the Euclidean Plane

In the Euclidean plane, finding the costs of both trucks taking their respective shortest paths is an easy and straightforward exercise. Finding the costs of platooning, on the other hand, requires us to determine the meeting point m and split point s that minimize the platoon costs. Once we

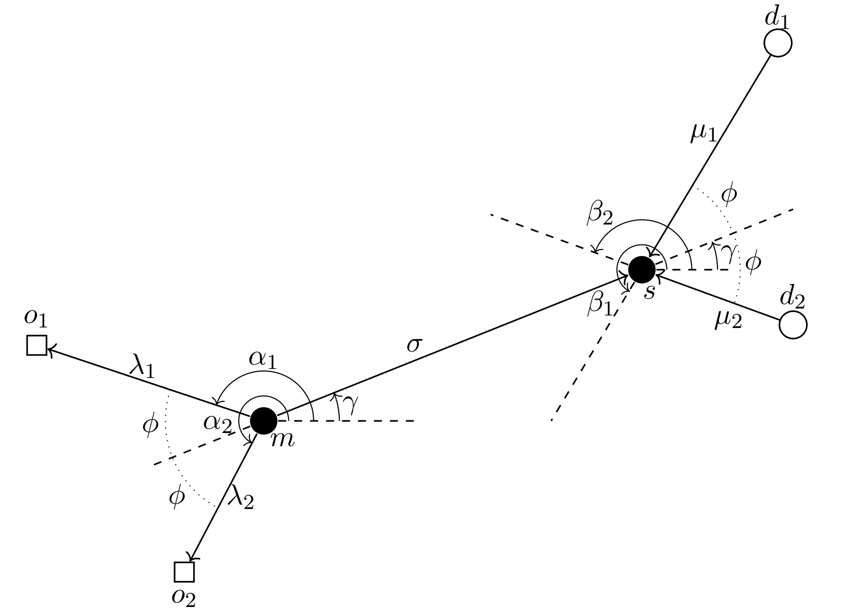


Figure 2: Platooning distance optimization in $\mathbb{R}^2$

have these points, the travel costs may be determined as follows:

$$L(m,s) = \sum_k \|o_k - m\|_2 + 2\rho\|m-s\|_2 + \sum_k \|s - d_k\|_2 \tag{18}$$

where we assume that $s \neq m$. Observe that in the case when $s = m$, we arrive at

$$L(m,m) = \sum_k \Big\{ \|o_k - m\|_2 + \|m - d_k\|_2 \Big\} \geq \sum_k \|o_k - d_k\|_2,$$

which implies that platooning is not beneficial. Also when $\rho = 1$, platooning will never result in savings, so we do not consider this case. Another case in which platooning is not beneficial is when $\|o_1 - o_2\|_2 + \|d_1 - d_2\|_2 \geq \sum_k \|o_k - d_k\|_2$, as the total distance with platooning will then always be larger than the direct truck distances.

We now describe how we find m and s. Figure 2 shows the notation that we use. In addition to these, we introduce $\delta_k \in \{-1, 1\}$ which specify the orientation of the origins and destinations with respect to each other. For instance, in Figure 2, we have $\delta_1 = +1$ and $\delta_2 = -1$. A sign change of δ_1 corresponds with a swap of the two origins and a sign change of δ_2 represents a swap in the destinations. Also, define unit vector $u_\delta = (\cos(\gamma + \pi + \delta\phi), \sin(\gamma + \pi + \delta\phi))^T$ for $\delta \in \{-1, +1\}$, and $v = (\cos\gamma, \sin\gamma)^T$. The proof of Theorem 3 is provided in Appendix A.

Theorem 3 *For* $0 < \rho < 1$*, the total cost* $L(m,s)$ *attains a unique minimum at meeting point* m *and separation point* s*, which can be determined as follows. We write* $\lambda_k = \|o_k - m\|_2$*,* $\mu_k = \|s - d_k\|_2$*, and* $\sigma = \|s - m\|_2$*. This implies*

$$L(m,s) = \sum_k \lambda_k + 2\rho\sigma + \sum_k \mu_k. \tag{19}$$

Moreover, we have that $m = o_1 - \lambda_1 u_{-\delta_1} = o_2 - \lambda_2 u_{+\delta_1}$, $s = d_1 + \mu_1 u_{-\delta_2} = d_2 + \mu_2 u_{+\delta_2}$, *and* $s - m = \sigma v$. *In addition,*

$$\lambda_1 = -r_o \frac{\sin(\varphi_o - \gamma - \delta_1 \phi)}{\sin 2\delta_1 \phi}, \tag{20}$$

$$\lambda_2 = -r_o \frac{\sin(\varphi_o - \gamma + \delta_1 \phi)}{\sin 2\delta_1 \phi}, \tag{21}$$

$$\mu_1 = r_d \frac{\sin(\varphi_d - \gamma - \delta_2 \phi)}{\sin 2\delta_2 \phi}, \tag{22}$$

$$\mu_2 = r_d \frac{\sin(\varphi_d - \gamma + \delta_2 \phi)}{\sin 2\delta_2 \phi}, \tag{23}$$

and

$$\sigma = r_1 \cos(\varphi_1 - \gamma) - (\lambda_1 + \mu_1)\cos\phi. \tag{24}$$

These parameters are defined by the initial data $o_2 - o_1 = (r_o \cos\varphi_o, r_o \sin\varphi_o)^T$, $d_2 - d_1 = (r_d \cos\varphi_d, r_d \sin\varphi_d)^T$, *and* $d_1 - o_1 = (r_1 \cos\varphi_1, r_1 \sin\varphi_1)^T$, *and the angle* γ. *The latter is given by*

$$\gamma = \arctan\frac{A}{B} + \delta_3 \pi, \tag{25}$$

where $\delta_3 \in \{0, 1\}$ *and* A *and* B *are defined by the initial data as follows:*

$$A = 2r_1 \cos\phi \sin\varphi_1 + r_d \sin(\varphi_d - \delta_2\phi) - r_o \sin(\varphi_o - \delta_1\phi), \tag{26}$$

$$B = 2r_1 \cos\phi \cos\varphi_1 + r_d \cos(\varphi_d - \delta_2\phi) - r_o \cos(\varphi_o - \delta_1\phi). \tag{27}$$

The angle ϕ *is given by* $2\phi = |\alpha_1 - \alpha_2| = |\beta_1 - \beta_2|$.

Next, the following theorem confirms that in the case of the Euclidean Plane, the meeting point m and the separation point s are in the convex hull of the origins and the destinations. We define the convex hull of a set of points as

$$\mathrm{Conv}(\{x_i\}) = \left\{ \sum_i a_i x_i \mid \sum_i a_i = 1 \right\}.$$

Theorem 4 *The meeting point* m *and the separation point* s*, as defined in Theorem 3 are in the convex hull of the origins and the destinations, i.e.* $s, m \in \mathrm{Conv}(\{o_1, o_2, d_1, d_2\})$. *In particular, convex combinations are given, when* $\delta_1\delta_2 = -1$*, by*

$$m = \frac{\lambda_2\theta}{w_{11}} o_1 + \frac{\lambda_1(\mu_1 + \theta)}{w_{11}} o_2 + \frac{\lambda_1\lambda_2}{w_{11}} d_1, \tag{28}$$

$$s = \frac{\mu_1\mu_2}{z_{22}} o_2 + \frac{\mu_2(\lambda_2 + \theta)}{z_{22}} d_1 + \frac{\mu_1\theta}{z_{22}} d_2, \tag{29}$$

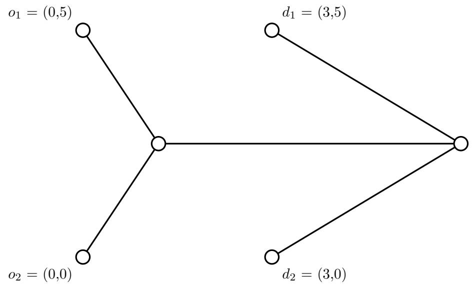


Figure 3: Example of a sparse network on which the trucks would automatically platoon

and, when when $\delta_1\delta_2 = +1$, *by*

$$m = \frac{\lambda_2(\theta + \mu_1)}{w_{21}} o_1 + \frac{\lambda_1\theta}{w_{21}} o_2 + \frac{\lambda_1\lambda_2}{w_{21}} d_1, \tag{30}$$

$$s = \frac{\mu_1\mu_2}{z_{21}} o_2 + \frac{\mu_2\theta}{z_{21}} d_1 + \frac{\mu_1(\lambda_2 + \theta)}{z_{21}} d_2, \tag{31}$$

where

$$\theta = \frac{\sigma}{2\rho},$$

$$w_{ij} = \lambda_i\mu_j + \lambda_1\lambda_2 + \lambda_1\theta + \lambda_2\theta,$$

$$z_{ij} = \lambda_i\mu_j + \mu_1\mu_2 + \mu_1\theta + \mu_2\theta,$$

for $i, j \in \{1, 2\}$.

The proof of Theorem 4 is given in Appendix A.

We now investigate how the solution of the PPRP on the Euclidean plane translates to the PPRP. This would hinge on the density of the network underlying the PPRP. For instance, the Euclidean solution could indicate that the truck pair cannot feasibly form a platoon but they might be able to on a sparse network. For example, given the network in Figure 3, the only possible paths would automatically create a platoon.

On the other hand, in a dense network one would expect the Euclidean solution being a good approximation for the network solution. However, this is not necessarily the case. Consider the example in Figure 4 where δ defines the grid density. In this grid, the lengths of shortest path between the points (0,0) and (2,1) can be calculated as $1 + \sqrt{2}$ while the corresponding value on the Euclidean plane is $\sqrt{5}$. Note that this shortest path on the grid is independent of grid density δ. This shows us that making the grid denser does not influence the ratio of the distances in the network and the Euclidean plane.

We use this to build up an example we show in Figure 5. Let the fuel consumption factor $\rho = \frac{2}{\sqrt{5}}$. Here, we can calculate the costs of platooning in the Euclidean plane as approximately 17.88 while the corresponding costs on a the network are 18.18. Comparing these figures to the

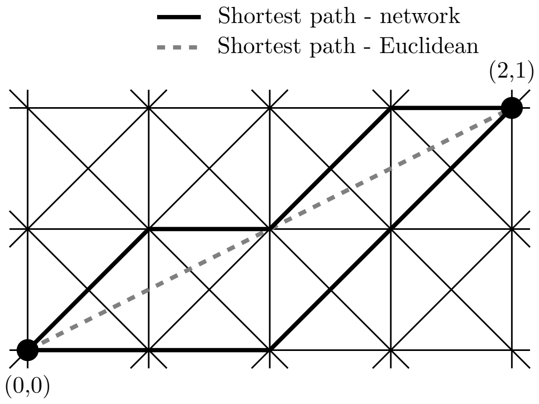
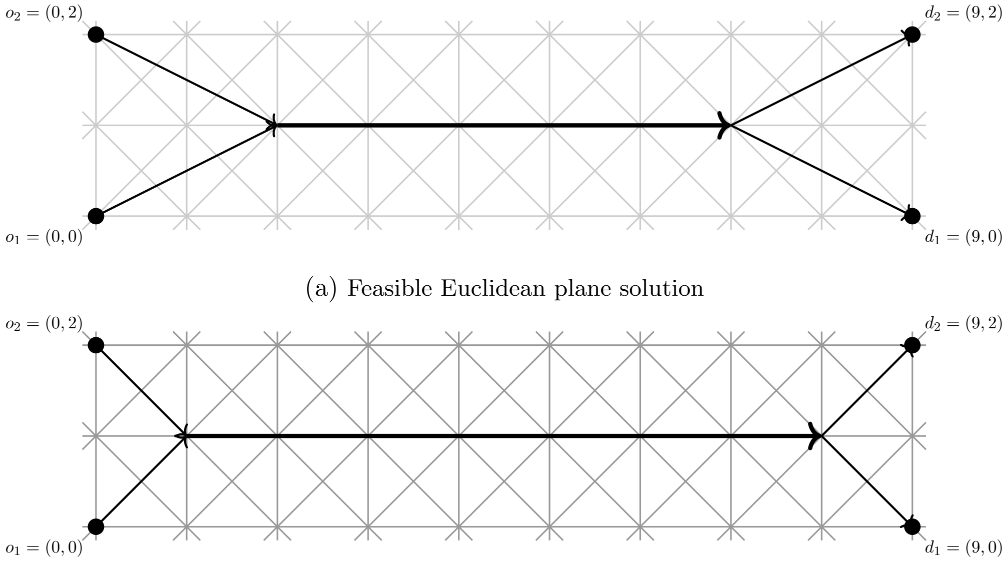


Figure 4: Shortest paths on the network and Euclidean plane



(a) Feasible Euclidean plane solution

(b) Infeasible network solution

Figure 5: Example of the optimal platoon for a truck pair being feasible on the Euclidean plane but being infeasible on a dense network; $\rho = \dfrac{2}{\sqrt{5}}$

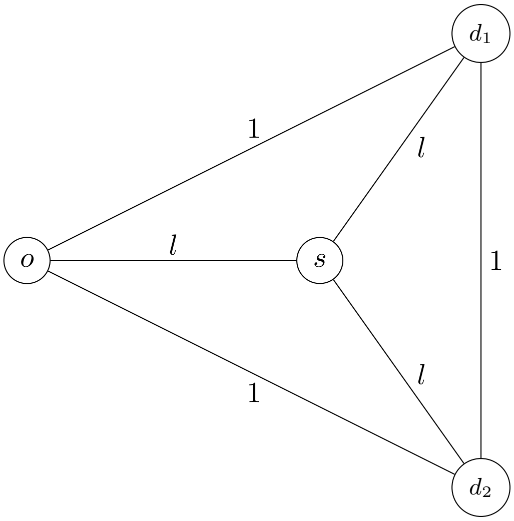


Figure 6: Counter example in case of networks

direct trucking costs of 18 indicates that a platoon is feasible on the Euclidean plane but not on the dense network.

Therefore, the theoretical translation of the Euclidean solution to the network case is not straightforward. We investigate this further in our numerical experiments on a practical case in Section 7.

Next, we evaluate if a similar convexity result as in Theorem 4 holds true in the case of a network graph. In a graph setting, the notion of geodesic convexity has been prominently used (Farber and Jamison 1987). In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, a set of nodes $\mathcal{C} \subseteq \mathcal{V}$ is geodesically convex if $\mathcal{C}$ contains all nodes in the shortest path between $u, v \in \mathcal{C}$. The geodesic convex hull of $\mathcal{C}$ is then the smallest convex set that contains $\mathcal{C}$.

In Figure 6, we present an example in which split point is NOT contained in the geodesic convex hull of the origins and destinations. Here, we have the geodesic convex hull $conv_g(o, d_1, d_2) = \{o, d_1, d_2\}$. We can easily calculate the value of l to be $\frac{1}{3}\sqrt{3}$. We have two trucks $k \in \{1, 2\}$ traveling from the same origin o to their destinations d_1 and d_2. Let $L(o, x)$ denote the costs of both trucks driving to their destinations splitting up at node x. Here, x can either o, s, and d_k. We can calculate the costs of each -

- $L(o, o) = \sum_k d_{od_k} = 2$
- $L(o, s) = 2 \cdot \rho \cdot d_{os} + \sum_k d_{sd_k} = \frac{2}{3}\sqrt{3}(1 + \rho)$
- $L(o, d_k) = 2 \cdot \rho \cdot d_{od_k} + d_{d_1 d_2} = 2\rho + 1$

Since $\rho > 1/2$ for a 2-truck platoon (see Section 3), we will always have $L(o, d_k) > L(o, o)$ so the trucks would never choose this route to minimize costs. For $L(o, s) < L(o, o)$, we need

$\frac{2}{3}\sqrt(3)(1+\rho) < 2$ or $\rho < \sqrt{3}-1 \approx 0.73$ so for any $0.5 < \rho < 0.73$, the trucks will split up at $s \notin conv_g(o, d_1, d_2)$.

6 Heuristics

In this section, we present a number of heuristics for the PRP-TW based on the special cases discussed in Section 5. The general idea is to find a 2-truck platooning solution first and then use an *improvement* procedure to potentially find additional (larger) platoons.

6.1 Shortest path heuristic

The shortest-path heuristic (SPH) creates a set of 2-truck platoons. As described in Section 5, we can formulate the associated truck pair routing problem as a general matching problem on a platoon pair graph $\mathcal{G}_{\mathcal{P}}$ to minimize total costs. To determine the costs of a platoon between truck pair $i, j \in \mathcal{K}$ we use a heuristic approach based on the procedure described in Section 5.1.

As a first step, we use the shortest path approach to find the best routes for every 2-truck pair of trucks $i, j \in K$ that satisfy Property 2 to form a platoon. This involves finding the appropriate meet and split points. If the shortest path algorithm does not find a feasible platoon, we do not include this pair in the matching. If a beneficial platoon is found, we test for time feasibility. That is, we check if the trucks can both reach their destinations in time when meeting as early as possible. If the platoon is time feasible we include these costs in the platoon pair graph $\mathcal{G}_{\mathcal{P}}$. If the optimal spatial platoon route is not beneficial or time feasible, we assume the trucks can not form a platoon. To denote this, we set the costs of the associated pair to be the sum of their direct distances. Note that this is a heuristic for the time constrained two truck case as we ignore the possibility that there could exist a time feasible platooning route that is different from the beneficial route that ignored time windows.

6.2 Euclidean heuristic

As in the previous section, we construct a general matching problem to create 2-truck platoons. In the Euclidean heuristic (EUH), we apply the analytical results of Section 5.2 to determine possible platoon pairs. In particular, we use expressions (28) - (31) to find the potential meeting and split points on the Euclidean plane for each pair of trucks $i, j \in \mathcal{K}$. We then 'round' these Euclidean coordinates to the closest discrete nodes on the network $\mathcal{G}$. That is, for both the meet and split point, we find the nearest node on the network in terms of Euclidean distance. If this platoon route is both beneficial and time-feasible, we include these platoon costs for truck pair i, j in the platoon pair graph $\mathcal{G}_{\mathcal{P}}$ and if not, we use the sum of their direct distances.

6.3 Improvement procedure

The above approaches generate platooning solutions with 2-truck platoons. It may, however, be possible and beneficial to still create larger platoons by merging some of the trucks that drive alone to a platoon. Therefore, we run an improvement procedure that is inspired by the shortest path approach in Section 5.1.

For a given platoon solution, let $\mathcal{P}$ be the set of trucks that are part of a platoon, and $\mathcal{Q}$ be the set of trucks that are not part of a platoon. In the graph, we now modify the costs of traveling along the path of an existing platoon to capture the potential savings of joining that platoon. We do this by multiplying the weights of the edges along all these paths by ρ. Let the network with the modifications for all trucks $i \in \mathcal{P}$ be $\mathcal{G}'$.

We then route trucks in $\mathcal{Q}$ on $\mathcal{G}'$ and check if there are platoon possibilities. For each truck $j \in \mathcal{Q}$, we check if its least cost route in the modified network $\mathcal{G}'$ overlaps with existing platoon routes and evaluate for time feasibility. In case of time feasibility, truck j can join a platoon in $\mathcal{P}$ to decrease total travel costs. Note that, Property 3 implies that we do not need to explicitly check for the platoon size constraint N for $N \geq 3$.

We can apply this improvement procedure on both SPH and EUH giving us two additional heuristics which we call SPH$^+$ and EUH$^+$.

7 Numerical experiments

In this section, we describe and report the results of a set of computational experiments on different instances. We determine the benefits of different platooning configurations by varying the maximum number of trucks per platoon and the number of platoons per truck trip. As we mentioned in Section 1, platoons of three trucks in practice already cause a lot more disruption to traffic than 2-truck platoons. Keeping this in mind, we restrict ourselves to arrangements in which a maximum of three trucks are allowed in a platoon. Note that increasing the maximum allowed platoon size will not create more platoons as an implication of Property 3. In addition, the cost structure we use (see Section 3) would keep the total costs the same.

We then follow this up by assessing the performance of the different heuristics. Since the heuristics take advantage of structural properties of restricted cases, we use these results to provide more detailed insights. We also conduct certain sensitivity analyses to assess the effects of different parameters on the total platooning benefits. In the MIP formulation, we use a discretization parameter of 15 minutes. Tests show that this value provides a good balance between performance and computational efficiency (see Appendix B). All results presented in this section have been averaged over five random truck trip instances and were implemented in Python 3.6.3 with Gurobi 7.0 as the IP solver on an Intel i7-6820HQ machine with 2.70Ghz processor.

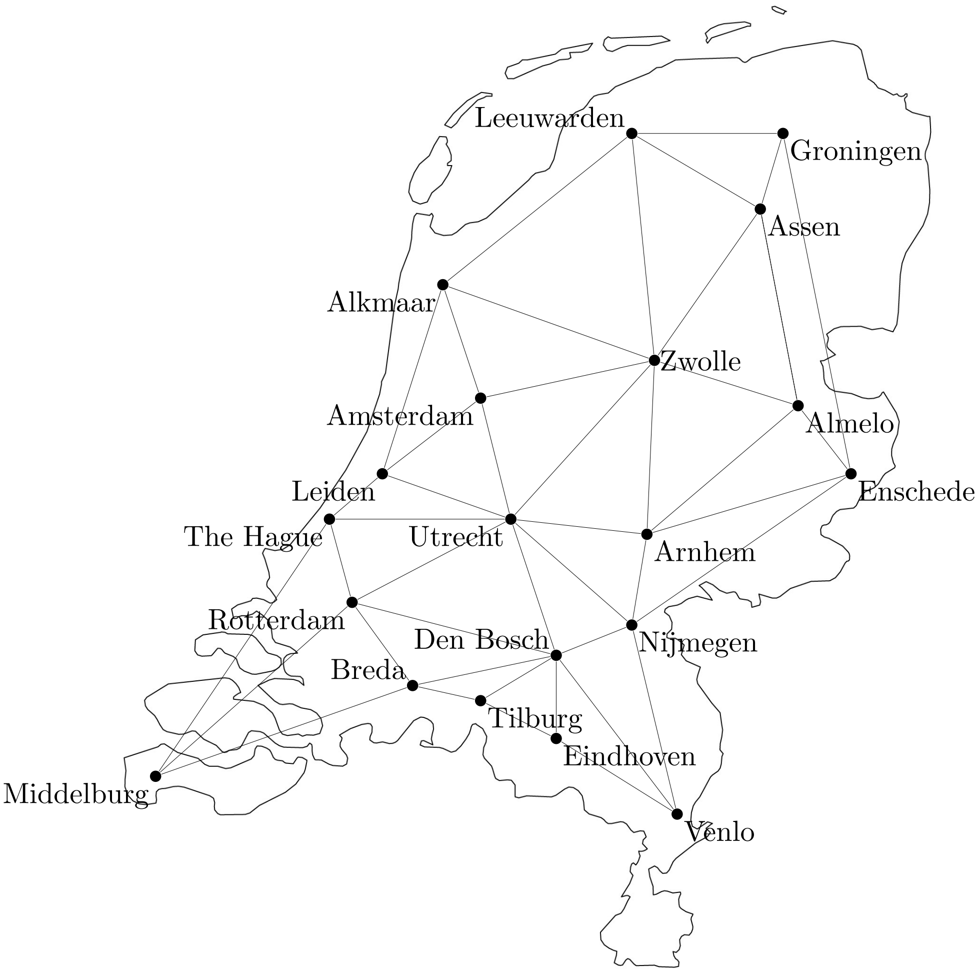


Figure 7: Representation of the Dutch highway network

7.1 Description of the instances

We generate instances on a network that represents the Dutch highway system with 20 cities represented by nodes and 45 road segments represented by edges. Figure 7 shows the network. The weights of the edges represent the travel costs between two cities and are proportional to the real distances (Google maps). In particular, we generate three sets of trip instances with different distributions of origins and destinations. In the first set of instances, UNIFORM, each node in the network is equally likely to be an origin and a destination for a truck. The next set of instances is inspired by our collaboration with the Port of Rotterdam. Here, all trips either start (ONEORIG) or end at one node (ONEDEST) in the network, i.e., the node that represents Rotterdam.

For all instances, we generate the earliest start times randomly in the one hour window between 07.00 and 08.00, which is inspired by our collaboration with various transportation service providers that operate between the Port of Rotterdam and the hinterland. We calculate their latest arrival times by adding the available flexibility to their shortest path times or $l_k = a_k + t_{o_k d_k} + f_k$ for truck $k \in \mathcal{K}$. We use a fuel consumption factor ρ of 0.90 which corresponds to 10% savings of driving in a platoon (McAuliffe et al. 2018). To facilitate easier interpretation of the results, we use a constant flexibility of 30 minutes for all trucks.

7.2 Impact of constraints

Here, we consider the impact of the different problem constraints on the solutions and run times. In particular, we look at the maximum number of trucks ($N \geq 2$) per platoon and the number of platoons per trip (one or unrestricted - $M \in \{1, \infty\}$). Recall that we denote each setting with the tuple (N, M). Therefore, we determine optimal platoons for a number of settings. For $(3, \infty)$ and $(2, \infty)$, we use the MIP from Section 4. For $(2, 1)$, we use the procedure from Section 5.1.

In Table 1, we report the percentage fuel savings, the total platooning kilometers as a percentage of the total direct distance, the proportion of trucks that are part of at least one platoon, the percentage of platoons with at least one truck making a detour, and the total detour distance expressed as a percentage of the total direct distance for the trucks with a detour. We calculate the percentage fuel savings as follows. Let Z^0 be the total costs if all trucks took their shortest paths and Z^* be the total costs with platooning. The percentage fuel savings are then

$$\text{Percentage savings} = 100 \cdot \frac{Z^0 - Z^*}{Z^0}$$

As expected, less constrained $(3, \infty)$ allows for more platoon savings than the more constrained 2-truck cases. However, we see that the differences are very small, for instance, a 0.2 percentage point difference between $(3, \infty)$ and $(2, \infty)$ in UNIFORM. This also applies to the percentage platoon kilometers (1 p.p.) and also the percentage of trucks in a platoon. There is no difference between the 2-truck cases with both $(2, \infty)$ and $(2, 1)$ having fuel savings of 3.3%. This suggests that the value of allowing multiple platoons per trip is low.

Max. trucks per platoon	3	2	2
Platoons per trip	∞	∞	1
UNIFORM			
Fuel savings (%)	3.5	3.3	3.3
Platoon km (%)	37.3	36.3	35.6
Trucks in platoon (%)	60.0	56.7	53.3
Platoons with detour (%)	19.6	20.9	8.8
Detour distance (%)	2.5	3.0	2.1
ONEORIG			
Fuel savings (%)	9.5	8.8	8.8
Platoon km (%)	95.2	89.1	88.8
Trucks in platoon (%)	99.3	98.0	94.7
Platoons with detour (%)	1.0	2.8	0.7
Detour distance (%)	0.1	1.8	1.1
ONEDEST			
Fuel savings (%)	8.4	7.6	7.5
Platoon km (%)	83.6	76.7	76.4
Trucks in platoon (%)	88.0	86.7	84.0
Platoons with detour (%)	1.8	3.0	3.1
Detour distance (%)	0.1	3.5	3.3

Table 1: Platooning solution characteristics; 30 trucks; averaged over five random instances; $\rho = 0.9$

Moreover, we see that there are higher savings in ONEORIG and ONEDEST as compared to UNIFORM. ONEORIG results in 9.5% savings as compared to 3.5% for UNIFORM. The reason is that it is much easier for trucks to find platoon partners when they are all starting from the same origin or heading to the same destination. We see that almost all trucks are part of a platoon in most cases. Looking at the different constraints across the instances, it appears that the value of having fewer constraints (i.e., (3,∞)) is greater in ONEORIG and ONEDEST. One potential explanation is that it is easier to exploit the extra flexibility in these instances because the trips of all trucks are spatially similar.

Moreover, we see higher savings (and more platoon km) in ONEORIG than in ONEDEST. This is somewhat surprising as both instances have a very similar route structure. The reason for this difference is related to the distribution of the time windows. Recall that the earliest departure times of all trucks are uniformly distributed between 7.00 and 8.00. This means that all trucks leave the origin around the same time but arrive at their destination at different times depending on their travel distances. As a result, the variation in arrival times is larger than the variation in departure times, which means that trucks can more easily be matched into platoons at their origins that at their destinations.

We now look at the detour characteristics. The percentage of platoons with a truck making a detour in UNIFORM is the highest for $(2, \infty)$ at 20.9% compared to the 19.6% for $(3, \infty)$ and 8.8% for $(2, 1)$. This is due to the difference in freedom trucks have to join platoons. In $(3, \infty)$, trucks have more freedom and can find platoons without longer detours. In addition, the costs of a detour prohibit trucks from significantly deviating from their shortest paths. Therefore, they tend to find platoon partners closer to the vicinity of their direct routes. This would also hold true when trucks are spatially clustered like in ONEORIG and ONEDEST, where only 1% and 1.8% of the trucks make a detour.

Note that although the additional fuel savings are small, the run times are much larger in the cases that allow for multiple platoons per truck. This is evident from Table 2 in which we report the average, maximum, and minimum run times of each setting for all sets of instances. For UNIFORM, $(3, \infty)$ requires 516 seconds on average while $(2, \infty)$ and $(2, 1)$ require 316 seconds and 280 seconds respectively.

Among the different instances, ONEORIG mostly has the largest run times. This is because of the higher number of possible platooning opportunities as a result of the trucks starting at the same node with less variability in their departure times than ONEDEST like we discussed. However, the run times of $(2, 1)$ are unexpectedly high for ONEDEST. This is again due to the higher variability in the arrival times of the trucks at the common destination. Because all trucks are heading to the same destination, there are many routes in close spatial proximity. As a result, there will be many beneficial platoon pairs but because of the variability in time, they are likely to be time-infeasible. Recall from Section 5.1 that the MIP needs to be called for these pairs, which adds to the run times.

Max. trucks per platoon	3	2	2
Platoons per trip	∞	∞	1
UNIFORM			
Average	516	316	280
Maximum	662	389	345
Minimum	420	248	209
ONEORIG			
Average	2453	773	416
Maximum	4951	850	627
Minimum	1196	633	266
ONEDEST			
Average	1172	576	846
Maximum	2160	768	1022
Minimum	538	444	635

Table 2: Run times in seconds for different instances; 30 trucks; averaged over five random instances; $\rho = 0.9$

7.3 Solution structure

The solution in which a truck can be part of multiple platoons along the route creates interdependencies between different platoons in time and space. To provide more insights into these interdependencies, we study the structure of the solution by looking at the larger platoon arrangements that have overlaps in the involved trucks.

To analyze these platoon arrangements, we create an undirected *truck connectivity graph* $\mathcal{P} = (\mathcal{K}', \mathcal{E})$. $\mathcal{K}'$ contains a node for each truck that is part of at least one platoon. Two nodes are connected by an edge $e \in \mathcal{E}$ if the associated trucks are in a platoon together. Each component in the truck connectivity graph represents a platoon arrangement. A component is a sub-graph in which every node is connected to every other node by a path. Two examples of platoon arrangements and their corresponding truck connectivity graphs are shown in Table 3. In the first platoon arrangement, trucks 1 and 2 are in a platoon together and are therefore connected in the truck connectivity graph. The following platoon arrangement has trucks 3,4,5 in a platoon together at first due to which they are all connected to each other in the corresponding truck connectivity graph. The platoon then loses truck 3 and truck 6 joins. Therefore, truck 6 is connected to trucks 4,5 in the truck connectivity graph but not to truck 3 since they are never in the same platoon together.

From the truck connectivity graph, we compute several statistics. The number of components provides more insight into the number of platoon arrangements in the solution. The size of these components is the number of trucks in their corresponding platoon arrangements. We also find the

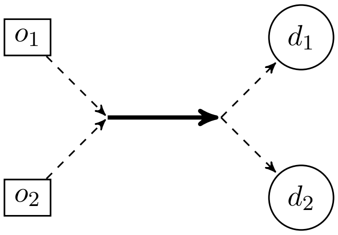
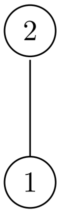
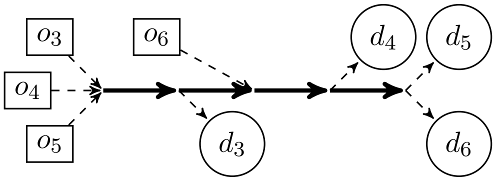
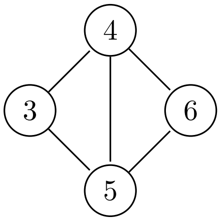
Platoon arrangement	Truck connectivity graph
	
 o_3 o_6 o_4 o_5 d_4 d_5 d_3 d_6	

Table 3: Examples of truck connectivity graphs

path between any two connected nodes in the truck connectivity graph to measure the separation between trucks. The separation indicates if two trucks are in a platoon together or if two trucks are linked by some common platoon partners and if so, how many. Therefore, the higher the separation is between any two trucks in a platoon arrangement, the higher the number of trucks there are in the arrangement, and the more frequently a truck joins or leaves a platoon in the arrangement.

Table 4 presents the number of arrangements, the average and largest arrangement sizes, and the largest separation. Note that the minimum component size is always 2. As expected, we see only components of size 2 in $(2, 1)$. Here, all platoons are independent. Comparing $(2, 1)$ and $(2, \infty)$ we see that the component sizes and the largest separation increases slightly when trucks are allowed to switch between different 2-truck platoons. The arrangement sizes and largest separation increase significantly for $(3, \infty)$. This suggest that trucks "piggyback" on existing 2-truck platoons close to their route. This is also the reason that we have fewer but larger arrangements in $(3, \infty)$. The higher number of trucks in these arrangements reiterates the requirement of the greater coordination efforts to plan these larger platoons.

For the different instances, we notice that the arrangement sizes are higher for ONEORIG and ONEDEST in $(3, \infty)$. In $(3, \infty)$, trucks have more freedom to join and leave platoons. The higher spatial proximity in ONEDEST and ONEORIG allows trucks to make more use of this greater freedom leading to larger platoon arrangements.

7.4 Performance of the heuristics

Here, we assess the performance of the heuristics presented in Section 6 by comparing their results with the optimal values of the $(3,\infty)$ solution. SPH and EUH represent the basic 2-truck

Max. trucks per platoon	3	2	2
Platoons per trip	∞	∞	1
UNIFORM			
Number of arrangements	6.6	8.4	8.0
Average arrangement size	2.5	2.1	2
Largest arrangement size	4.8	2.2	2
Largest separation	2.6	1.2	1
ONEORIG			
Number of arrangement	5.6	14.4	14.2
Average arrangement size	3.6	2	2
Largest arrangement size	14.4	2.2	2
Largest separation	6.8	1.2	1
ONEDEST			
Number of arrangement	8.0	12.8	12.6
Average arrangement size	2.9	2	2
Largest arrangement size	8.2	2.3	2
Largest separation	4.4	1.8	1

Table 4: Truck connectivity graph characteristics; 30 trucks; averaged over five random instances; $\rho = 0.9$

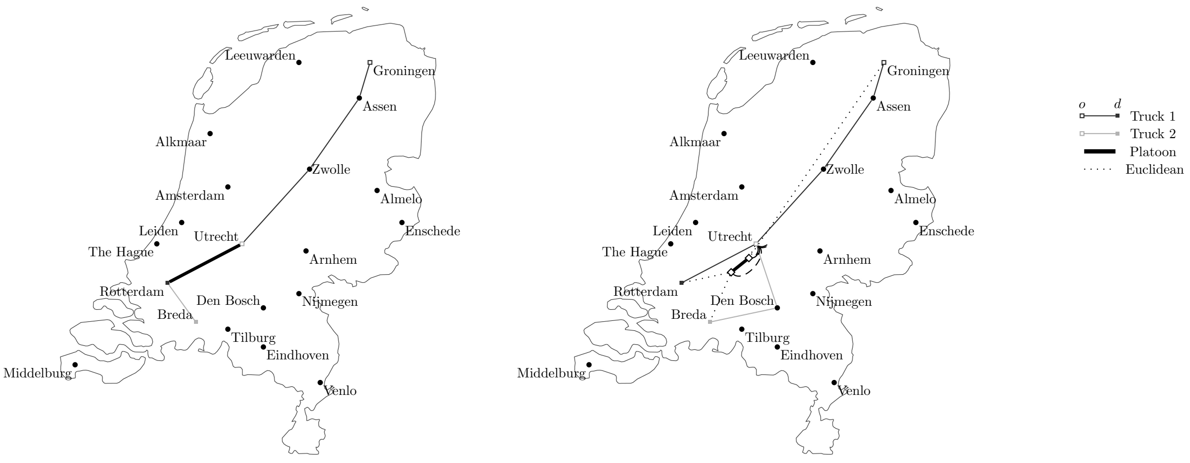
	SPH	SPH$^+$	EUH	EUH$^+$
UNIFORM				
Average (%)	0.34	0.15	2.42	2.09
Maximum (%)	0.77	0.24	3.52	3.52
Minimum (%)	0	0	1.59	1.08
# optimal	1	2	0	0
Run time (s)	1.79	1.81	0	0.01
ONEORIG				
Average (%)	0.79	0.55	2.10	1.97
Maximum (%)	0.95	0.90	2.20	2.06
Minimum (%)	0.59	0.32	2.05	1.90
# optimal	0	0	0	0
Run time (s)	1.83	1.84	0	0.01
ONEDEST				
Average (%)	1.31	0.25	2.23	2.03
Maximum (%)	1.46	0.81	2.41	2.27
Minimum (%)	1.18	0.12	2.12	1.87
# optimal	0	0	0	0
Run time (s)	1.75	1.77	0	0.01

Table 5: Optimality gaps and run times of heuristics; 30 trucks; averaged over five random instances; $\rho = 0.9$

platooning heuristics and SPH$^+$ and EUH$^+$ the versions in which we apply the improvement procedure. Table 5 reports the optimality gaps and run times of these four heuristics.

Overall, we see that SPH$^+$ performs best with optimality gaps of less than 1%. Even without the improvement step, SPH performs well, i.e., with optimality gaps of less than 1.46%. This is in line with our earlier observation that optimal (2,1) solutions are not so much worse than the less constrained optimal (3, ∞) solutions. However, the run times of the heuristics are far lower than the run times of our exact methods as shown in Table 2.

EUH and EUH$^+$ show even smaller run times but with a lower performance with 2-3% optimality gaps. A possible reason for the lower performance is that it can be difficult to translate the Euclidean solutions to a network solution as discussed in Section 5.2. To provide some more insight into this 'rounding' step, we show an example with two trucks in Figure 8. The first truck needs to travel from from Groningen to Rotterdam and the second from Utrecht to Breda. Figure 8a shows the optimal network solution in which the trucks platoon between Utrecht and Rotterdam. Figure 8b shows the Euclidean plane solution and the resulting rounded network solution. Both the meet and split points are rounded to Utrecht which means that no platooning takes place.



(a) Optimal 2-truck solution

(b) Euclidean 2-truck solution and rounded network solution

Figure 8: Example of optimal vs Euclidean solution

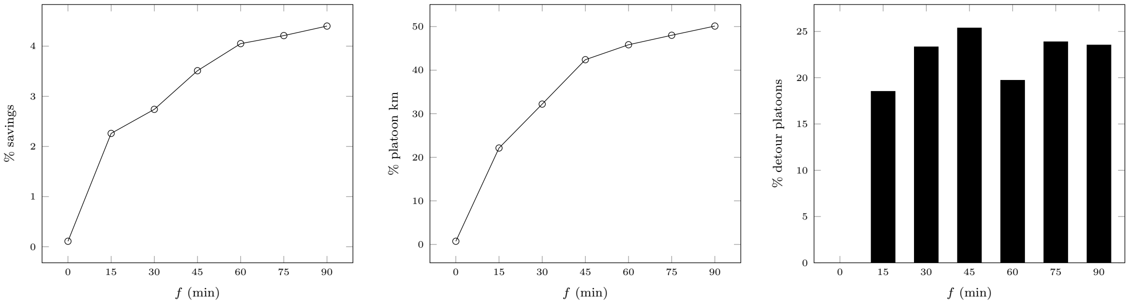
7.5 Impact of time flexibility, fuel consumption factor, and truck density

We now study the impact of the truck density (number of s) and their flexibility f on the performance of the system. In addition, we also look at the effect of the fuel consumption factor ρ. For these experiments, we use the best performing heuristic SPH$^+$. We run these sensitivity analyses using the UNIFORM instances.

We vary flexibility f between 0 and 90 minutes with increments of 15 minutes for instances with 30 trucks. Figure 9 shows the fuel savings, the platoon kilometers relative to the setting without platoons, and the percentage of platoons with a truck making a detour. From Figures 9a and 9b, we see that both the fuel savings and the percentage platoon kilometer values increase with higher flexibility. Moreover, we see a diminishing increase at higher flexibility. We observe that allowing a flexibility of 15 minutes makes a significant difference as compared to having no flexibility at all. From a practical perspective, our results suggest that we would only need a relatively small amount of time flexibility to achieve benefits from platooning.

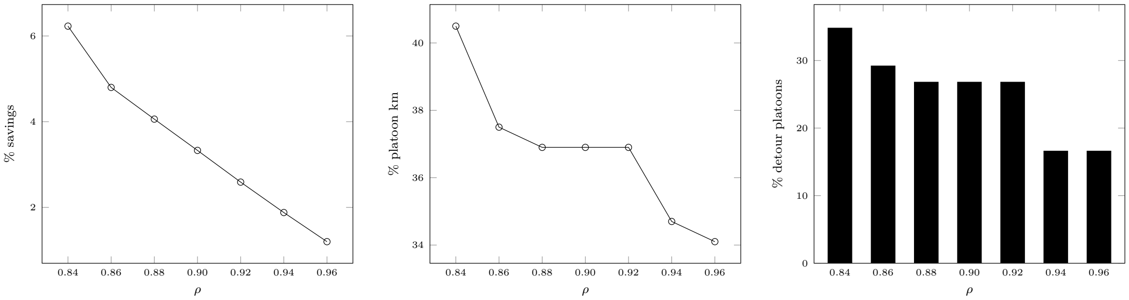
Interestingly, Figure 9c shows that the percentage of platoons in which a truck makes a detour is roughly the same for different values of flexibility. One explanation for this is that the detours are automatically minimized to maximize the platoon benefits. That is, long detours would offset fuel cost savings and thus render platoons non-beneficial. The experiments also show that the percentage detour distance also remains fairly constant with increasing flexibility.

In practice, the fuel consumption factor ρ depends on several factors like the surrounding traffic, terrain, or headway. In this experiment, we consider values between 0.84 and 0.96 in steps of 0.02. We again plot the percentage savings, the percentage platoon kilometers and the percentage of trucks in at least one platoon in Figure 10. We observe a roughly linear trend in the percentage



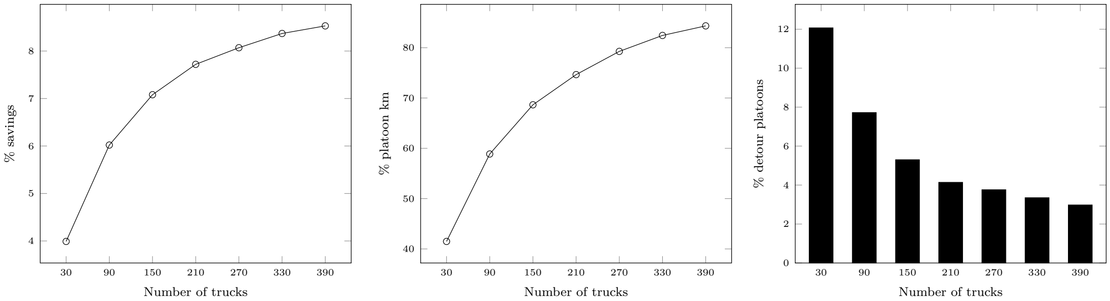
(a) Variation in % savings (b) Variation in % platoon km (c) Variation in % detour platoons

Figure 9: Effects of varying f; 30 trucks; averaged over five random instances; $\rho = 0.9$



(a) Variation in % savings (b) Variation in % platoon km (c) Variation in % detour platoons

Figure 10: Effects of varying ρ; 30 trucks; averaged over five random instances; flexibility = 30 min



(a) Variation in % savings (b) Variation in % platoon km (c) Variation in % detour platoons

Figure 11: Effects of varying the truck density; flexibility = 30 min; averaged over five random instances; $\rho = 0.9$

savings from Figure 10a. Figures 10a, as expected, indicates that the platoon kilometers increase with higher fuel savings, i.e., lower ρ. However, the differences are relatively small. We see a similar trend in Figure 10c with the percentage of trucks making a detour. Trucks can make more detours when the platoon savings are higher since the savings offset the additional detour costs.

Next, we vary the number of trucks between 30 and 390 in steps of 60. In Figure 11, we plot the percentage savings, the percentage platooning kilometers and the percentage of platoons with at least one truck making a detour. Figures 11a and 11b again show a diminishing increase. As expected, the number of platoons increases with the number of trucks. Hence, equipping more trucks with platooning technology at the deployment stages would make a significant difference.

Not only is it more likely for trucks to form platoons with a higher truck density but they also need less detours to do so. Figure 11c shows that the percentage of trucks that make a detour to form platoons goes down with an increase in the number of trucks.

8 Concluding remarks

Platooning technology virtually connects trucks and helps save fuel and reduce associated emissions by reducing aerodynamic drag. In this paper, we looked at the planning truck platoons by determining the paths and time schedules of s between their origins and destinations. In particular, we focused on a restricted case in which a truck can join at most one platoon and we only consider platoons of two trucks. We referred to this problem as the Platoon Pair Routing Problem (PPRP-TW and PPRP). We showed that this problem can be solved in polynomial time and provide several fast exact methods and heuristics. To evaluate the impact of restricting the possible platoons, we also present a novel MIP to solve the general platoon planning routing with time windows (PRP-TW).

To assess the impact of the restrictions and evaluate the heuristics, we ran a set of numerical experiments on instances based on the Dutch highway network. Our experiments showed that the restrictive setting with 2-truck platoons is associated with similar fuel savings as the less restricted settings. Moreover, it is computationally less challenging to create two truck platoons.

Our heuristics also provide very good results. They consistently produced results within a percent of the optimum.

In this paper, we focused on a static setting in which we plan the platoons in advance. A natural direction for further research is to consider dynamic trip announcements and travel time uncertainty.

9 Acknowledgements

Funding: This work was supported by the Netherlands Organization for Scientific Research (NWO) as part of the 'Spatial and Transport impacts of Automated Driving (STAD)' [Project number 438-15-161] projects.

References

Adler, A., Miculescu, D., Karaman, S., 2016. Optimal policies for platooning and ride sharing in autonomy-enabled transportation. URL: http://wafr2016.berkeley.edu/papers/WAFR_2016_paper_110.pdf. working paper - Massachussetts Institute of Technology.

Albinski, S., Crainic, T., Minner, S., 2020. The day-before truck platooning planning problem and the value of autonomous driving. Technical Report. CIRRELT.

Bergenhem, C., Schladover, S., Coelingh, E., 2012. Overview of platooning systems. Proceedings of the 19th ITS World Congress .

Bhoopalam, A.K., Agatz, N., Zuidwijk, R., 2018. Planning of truck platoons: A literature review and directionf for future research. Transportation Research Part B: Methodological 107, 212–228.

Bodwin, G., 2019. On the structure of unique shortest paths in graphs, in: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM. pp. 2071–2089.

Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. Operations research 65, 1303–1321.

Boysen, N., Briskorn, D., Schwerdfeger, S., 2018. The identical-path truck platooning problem. Transportation Research Part B: Methodological 109, 26–39.

Calvert, S., Schakel, W., van Arem, B., 2019. Evaluation and modelling of the traffic flow effects of truck platooning. Transportation research part C: emerging technologies 105, 1–22.

Colling, A., Hekkenberg, R., 2020. Waterborne platooning in the short sea shipping sector. Transportation Research Part C: Emerging Technologies 120, 102778.

Dantzig, G.B., 1960. On the shortest route through a network. Management Science 6, 187–190.

Farber, M., Jamison, R.E., 1987. On local convexity in graphs. Discrete Mathematics 66, 231–247.

Gabow, H.N., 1990. Data structures for weighted matching and nearest common ancestors with linking, in: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms, pp. 434–443.

Goble, K., 2018. Seven more states adopt rule changes for truck platoons. Online Editorial. URL: http://www.landlinemag.com/story.aspx?storyid=72230#.Wyj6vO6FNEY.

Van de Hoef, S., 2018. Fuel-Efficient Centralized Coordination of Truck platooning (Doctoral dissertation. Ph.D. thesis. KTH Electrical Engineering.

Kilcarr, S., 2016. Driverless trucks: Where they'll work, where they won't. URL: http://m.fleetowner.com/technology/driverless-trucks-where-they-ll-work-where-they-won-t.

Larsen, R., Rich, J., Rasmussen, T.K., 2019. Hub-based truck platooning: Potentials and profitability. Transportation Research Part E: Logistics and Transportation Review 127, 249–264.

Larson, J., Kramer, C., Lang, K.Y., Johannson, K.H., 2013. Coordinated route optimization for heavy duty vehicle platoons, in: Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), pp. 1196–1202.

Larson, J., Munson, T., Sokolov, V., 2016. Coordinated platoon routing in a metropolitan network, in: Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing, pp. 73–82. doi:10.1137/1.9781611974690.ch8.

Larsson, E., Sennton, G., Larson, J., 2015. The vehicle platooning problem: Computation complexity and heuristics. Transportation Research Part C: Emerging Technologies 60, 258–277. doi:10.1016/j.trc.2015.08.019.

Liang, K.Y., Mårtensson, J., Johansson, K.H., 2013. When is it fuel efficient for a heavy duty vehicle to catch up with a platoon?, in: IFAC Proceedings Volumes, pp. 738–743. doi:10.3182/20130904-4-JP-2042.00071.

Liang, K.Y., Mårtensson, J., Johansson, K.H., 2014. Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data, in: 2014 IEEE Intelligent Vehicles Symposium (IV), pp. 1061–1068. doi:10.1109/IVS.2014.6856540.

Lioris, J., Pedarsani, R., Tascikaraoglu, F.Y., Varaiya, P., 2017. Platoons of connected vehicles can double throughput in urban roads. Transportation Research Part C: Emerging Technologies 77, 292–305. doi:10.1016/j.trc.2017.01.023.

Luo, F., Larson, J., Munson, T., 2018. Coordinated platooning with multiple speeds. Transportation Research Part C: Emerging technlogies 90, 213–225.

van Maarseveen, S., 2017. Impacts of Trucks Platooning at Motorway On-ramps. Master's thesis. TU Delft.

Mahmassani, H.S., 2016. 50th anniversary invited article—autonomous vehicles and connected vehicle systems: Flow and operations considerations. Transportation Science 50, 1140–1162.

McAuliffe, B., Lammert, M., Lu, X.Y., Shladover, S., Surcel, M.D., Kailas, A., 2018. Influences on energy savings of heavy trucks using cooperative adaptive cruise control. Technical Report. SAE Technical Paper.

Nourmohammadzadeh, A., Hartmann, S., 2016. The fuel-efficient platooning of heavy duty vehicles by mathematical programming and genetic algorithm, in: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M. (Eds.), Theory and Practice of Natural Computing. TPNC 2016. Lecture Notes in Computer Science. Springer. volume 10071, pp. 46–57. doi:10.1007/978-3-319-49001-4_4.

Nourmohammadzadeh, A., Hartmann, S., 2019. Fuel-efficient truck platooning by a novel meta-heuristic inspired from ant colony optimisation. Soft Computing 23, 1439–1452.

Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E., 2018. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. Networks 72, 411–458.

Poikonen, S., Golden, B., 2020. The mothership and drone routing problem. INFORMS Journal on Computing 32, 249–262.

Poikonen, S., Golden, B., Wasil, E.A., 2019. A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS Journal on Computing 31, 335–346.

Richert, D., Cortés, J., 2013. Optimal leader allocation in uav formation pairs ensuring cooperation. Automatica 49, 3189–3198.

Schladover, S.E., Nowakowski, C., Lu, X.Y., Felis, R., 2015. Cooperative adaptive cruise control (CACC) definitions and operating concepts. Transportation Research Record: Journal of the Transportation Research Board 2489 , 145–152doi:10.3141/2489-17.

Scora, G., Barth, M., 2006. Comprehensive modal emissions model (CMEM), version 3.01. User guide. Centre for Environmental Research and Technology. University of California, Riverside .

Sokolov, V., Larson, J., Munson, T., Auld, J., Karbowski, D., 2017. Platoon formation maximization through centralized routing and departure time coordination. arXiv preprint arXiv:1701.01391 .

Sun, X., Yin, Y., 2019. Behaviorally stable vehicle platooning for energy savings. Transportation Research Part C: Emerging Technologies 99, 37–52.

Van Arem, B., Van Driel, C.J., Visser, R., 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. IEEE Transactions on Intelligent Transportation Systems 7, 429–436. doi:10.1109/TITS.2006.884615.

Winter, P., Zachariasen, M., 1998. Euclidean steiner minimum trees: An improved exact algorithm. Networks 30, 149–166. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/, doi:10.1002/(SICI)1097-0037(199710)30:3<149::AID-NET1>3.0.CO;2-L.

Zhang, W., Jenelius, E., Ma, X., 2017. Freight transport platoon coordination and departure time scheduling under travel time uncertainty. Transportation Research Part E: Logistics and Transportation Review 98, 1–23. doi:10.1016/j.tre.2016.11.008.

A Platooning on the Euclidean plane

We establish the proof of Theorem 3. Note that $L(m,s)$ as defined in (18) is differentiable whenever

$$(m,s) \in \mathcal{D} = \{(x,y) \in \mathbb{R}^2 \times \mathbb{R}^2 : x \neq y,\ o_k \neq x,\ y \neq d_k,\ k \in \{1,2\}\}.$$

We will demonstrate that $L(m,s)$ is strictly convex, so that there exists a unique (m,s) for which $L(m,s)$ attains its minimum value. When $L(m,s)$ has a critical point in $\mathcal{D}$, then $L(m,s)$ will attain its minimum value there. Otherwise, the minimum value will be attained at $\{(x,y) \in \mathbb{R}^2 \times \mathbb{R}^2 : x = o_k\}$ or $\{(x,y) \in \mathbb{R}^2 \times \mathbb{R}^2 : y = d_k\}$ for some $k \in \{1,2\}$. For a vector $x = (x_1, x_2)^T$, we define angle $\alpha = \angle(x) \in [0, 2\pi)$ such that $\cos\alpha = x_1/\|x\|_2$ and $\sin\alpha = x_2/\|x\|_2$. We set

$$\alpha_k = \angle(o_k - m),\ \beta_k = \angle(s - d_k) \text{ for } k = 1,2, \text{ and } \gamma = \angle(s - m). \tag{32}$$

We state the following lemma:

Lemma 1 *For minimum distance $L(m,s)$ as defined in (18) and attained in $\mathcal{D}$, angles as defined in (32), and with discount factor $0 \leq \rho < 1$, it holds true that*

$$\alpha_1 = \pi - \delta_1\phi + \gamma, \quad \alpha_2 = \pi + \delta_1\phi + \gamma, \tag{33}$$

and

$$\beta_1 = \pi - \delta_2\phi + \gamma, \quad \beta_2 = \pi + \delta_2\phi + \gamma, \tag{34}$$

where $\phi \in [0, \pi/2]$ satisfies $\cos\phi = \rho$, and $\delta_k \in \{-1, 1\}$ for $k = 1,2$.

Proof. We optimize L with respect to m and s coordinates in the plane; a critical point of $L(m,s)$ satisfies

$$\frac{\partial L}{\partial m_j} = \sum_k \frac{m_j - o_{kj}}{\|m - o_k\|_2} + 2\rho\frac{m_j - s_j}{\|m - s\|_2} = 0 \quad \text{for} \quad j = 1,2,$$

$$\frac{\partial L}{\partial s_j} = \sum_k \frac{s_j - d_{kj}}{\|s - d_k\|_2} + 2\rho\frac{s_j - m_j}{\|s - m\|_2} = 0 \quad \text{for} \quad j = 1,2.$$

With reference to the angles α_k, β_k for $k = 1,2$ and γ as defined in (32), we may write

$$\sum_k \cos\alpha_k = \sum_k \cos\beta_k = -2\rho\cos\gamma, \tag{35}$$

$$\sum_k \sin\alpha_k = \sum_k \sin\beta_k = -2\rho\sin\gamma. \tag{36}$$

Since (35) and (36) provide

$$\left(\sum_k \cos\alpha_k\right)^2 + \left(\sum_k \sin\alpha_k\right)^2 = 2 + 2\cos(\alpha_2 - \alpha_1) = 4\rho^2,$$

we arrive at $\cos(\alpha_2 - \alpha_1) = 2\rho^2 - 1$. If we set $\phi = \arccos \rho$, then $0 < \phi \leq \pi/2$ and $\cos(\alpha_2 - \alpha_1) = \cos 2\phi$. Similarly, we get $\cos(\beta_2 - \beta_1) = \cos 2\phi$. This implies

$$\alpha_2 = \alpha_1 + 2\delta_1\phi, \quad \delta_1 \in \{-1, +1\}, \tag{37}$$

$$\beta_2 = \beta_1 + 2\delta_2\phi, \quad \delta_2 \in \{-1, +1\}. \tag{38}$$

If we insert (37) into (35), we obtain $\cos(\alpha_1 + \delta_1\phi) + \cos\gamma = 0$, and if we insert (37) into (36), we obtain $\sin(\alpha_1 + \delta_1\phi) + \sin\gamma = 0$. We obtain similar results when we insert (38) into (35) and (36). This implies

$$\alpha_1 = \pi - \delta_1\phi + \gamma, \quad \alpha_2 = \pi + \delta_1\phi + \gamma, \tag{39}$$

$$\beta_1 = \pi - \delta_2\phi + \gamma, \quad \beta_2 = \pi + \delta_2\phi + \gamma, \tag{40}$$

which solves (35) and (36). Equations (37) - (40) should be read modulo 2π. □
This in particular implies that $|\alpha_1 - \alpha_2| = |\beta_1 - \beta_2| = 2\phi$, which corresponds to an observation about Steiner trees, where $\rho = 1/2$ and $2\phi = 2\pi/3$; see Winter and Zachariasen (1998).
We now state and prove the following Proposition and hence that $L(m, s)$ attains a unique minimum solution at the critical point for $0 < \rho < 1$, in case such critical point exists in $\mathcal{D}$. For $\rho = 0$, the minimum solution is not unique and minimum distance equals
$L(m, s) = \|o_1 - o_2\|_2 + \|d_1 - d_2\|_2$.

Proposition 2 *Whenever $0 < \rho < 1$, the function $(m, s) \mapsto L(m, s)$ is strictly convex, with exception of pairs from the set*

$$\begin{aligned} \mathcal{C} = \{\{(m_1, s_1), (m_2, s_2)\} \in \mathbb{R}^{2+2} \times \mathbb{R}^{2+2} : \angle(o_k - m_1, o_k - m_2) = 0, \\ \angle(m_1 - s_1, m_2 - s_2) = 0, \ \angle(s_1 - d_k, s_2 - d_k) = 0, \ k \in \{1, 2\}\}. \end{aligned} \tag{41}$$

The critical point of L in $\mathcal{D}$ is not part of any pair in $\mathcal{C}$.

Proof. To show that $L(m, s)$ is strictly convex, we take $(m_1, s_1) \neq (m_2, s_2)$, and write for $0 < \lambda < 1$,

$$L(\lambda m_1 + (1 - \lambda)m_2, \lambda s_1 + (1 - \lambda)s_2) =$$

$$\sum_k \|o_k - \lambda m_1 - (1 - \lambda)m_2\|_2 + 2\rho\|\lambda(m_1 - s_1) + (1 - \lambda)(m_2 - s_2)\|_2 +$$

$$\sum_k \|\lambda s_1 + (1 - \lambda)s_2 - d_k\|_2 \leq$$

$$\lambda \left\{ \sum_k \|o_k - m_1\|_2 + 2\rho\|m_1 - s_1\|_2 + \sum_k \|s_1 - d_k\|_2 \right\} +$$

$$(1 - \lambda) \left\{ \sum_k \|o_k - m_2\|_2 + 2\rho\|m_2 - s_2\|_2 + \sum_k \|s_2 - d_k\|_2 \right\} =$$

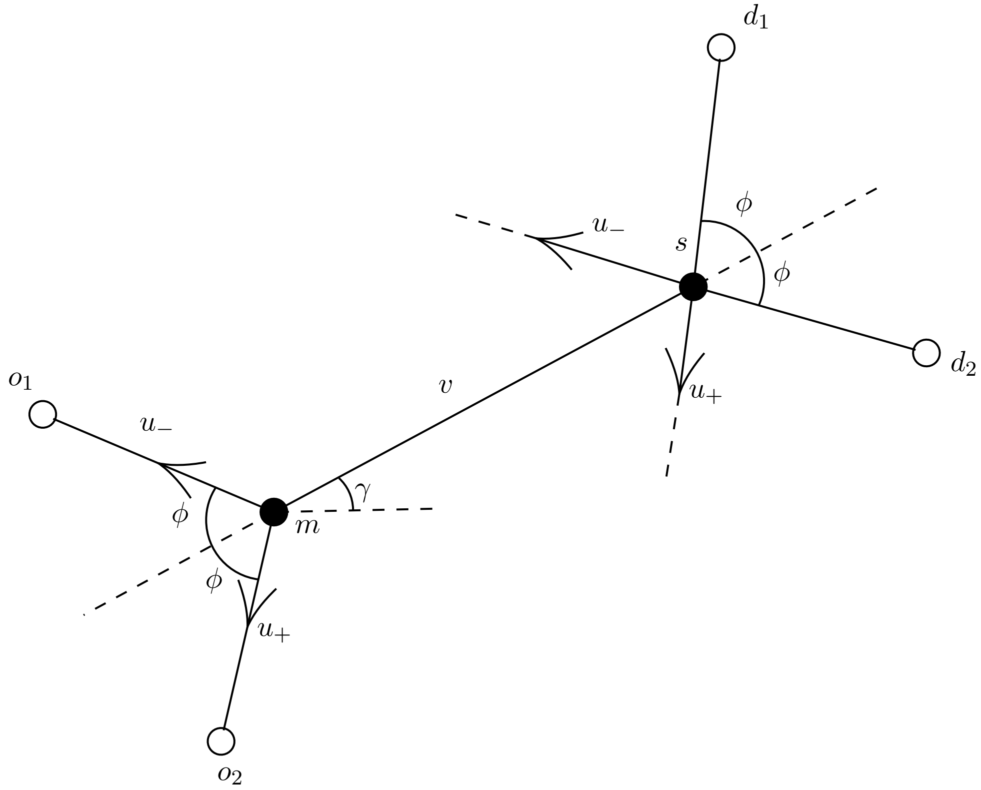


Figure 12: Platooning distance optimization in $\mathbb{C}$

$$\lambda L(m_1, s_1) + (1 - \lambda)L(m_2, s_2).$$

The inequality is strict when the pairs $((m_1, s_1), (m_2, s_2))$ satisfy either $\angle(o_k - m_1, o_k - m_2) \neq 0$, or $\angle(m_1 - s_1, m_2 - s_2) \neq 0$, or $\angle(s_1 - d_k, s_2 - d_k) \neq 0$ for some $k \in \{1, 2\}$, i.e., the pairs are outside the set $\mathcal{C}$.

The set $\mathcal{C}$ consists of those pairs $((m_1, s_1), (m_2, s_2))$ for which o_1, o_2, m_1, m_2 are on a single line M, s_1, s_2, d_1, d_2 are on another single line S, and m_1, s_1 and m_2, s_2 are on parallel lines. We prove that such points are never critical points in $\mathcal{D}$. In case either $(m_1, s_1) \in \mathcal{D}$ or $(m_2, s_2) \in \mathcal{D}$ is a critical point, then this results in a contradiction by Lemma 1, since for instance $\angle(o_1 - m_j, o_2 - m_j) = 2\phi$ for $j \in \{1, 2\}$, with $0 < 2\phi < \pi$. This implies that the pairs in set $\mathcal{C}$ never contain a critical point in $\mathcal{D}$. □

Proof of Theorem 3 We use the result of Lemma 1 in our representation of the platooning paths in the complex plane; we write

$$L = \lambda_1 + \lambda_2 + 2\rho\sigma + \mu_1 + \mu_2,$$

where $\lambda_k = \|o_k - m\|_2$, $\mu_k = \|s - d_k\|_2$ for $k = 1, 2$, and $\sigma = \|m - s\|_2$. We write unit numbers $u_\delta = e^{i(\pi+\delta\phi+\gamma)} = -e^{i(\delta\phi+\gamma)}$ with $\delta \in \{-1, +1\}$, and $v = e^{i\gamma}$; see also Figure 12.

This implies

$$o_1 = m + \lambda_1 u_{-\delta_1} = m - \lambda_1 e^{i(-\delta_1\phi+\gamma)},$$

$$o_2 = m + \lambda_2 u_{+\delta_1} = m - \lambda_2 e^{i(+\delta_1 \phi + \gamma)}.$$

We also write $s = m + \sigma e^{i\gamma}$, and

$$d_1 = s - \mu_1 u_{-\delta_2} = m + \sigma e^{i\gamma} + \mu_1 e^{i(-\delta_2 \phi + \gamma)},$$

$$d_2 = s - \mu_2 u_{+\delta_2} = m + \sigma e^{i\gamma} + \mu_2 e^{i(+\delta_2 \phi + \gamma)}.$$

The differences $\Delta_o = o_2 - o_1 = r_o e^{i\varphi_o}$, $\Delta_d = d_2 - d_1 = r_d e^{i\varphi_d}$, and $\Delta_1 = d_1 - o_1 = r_1 e^{i\varphi_1}$ are given by the inital data, and $\Delta_2 = d_2 - o_2 = \Delta_d - \Delta_o + \Delta_1$. Note

$$\Delta_o = o_2 - o_1 = (\lambda_1 e^{-i\delta_1 \phi} - \lambda_2 e^{i\delta_1 \phi}) e^{i\gamma} = r_o e^{i\varphi_o},$$

which implies

$$r_o e^{i(\varphi_o - \gamma)} = \lambda_1 e^{-i\delta_1 \phi} - \lambda_2 e^{i\delta_1 \phi}.$$

By taking real and imaginary parts of this equality, we get

$$\lambda_1 + \lambda_2 = -r_o \frac{\sin(\varphi_o - \gamma)}{\sin \delta_1 \phi}, \tag{42}$$

$$\lambda_1 - \lambda_2 = r_o \frac{\cos(\varphi_o - \gamma)}{\cos \delta_1 \phi}. \tag{43}$$

Solving this set of linear equations in λ_1 and λ_2, we arrive at

$$\lambda_1 = -r_o \frac{\sin(\varphi_o - \gamma - \delta_1 \phi)}{\sin 2\delta_1 \phi}, \tag{44}$$

$$\lambda_2 = -r_o \frac{\sin(\varphi_o - \gamma + \delta_1 \phi)}{\sin 2\delta_1 \phi}. \tag{45}$$

As we assumed $\lambda_k \geq 0$ for $k \in \{1, 2\}$, we need to ascertain that $\lambda_1 + \lambda_2 \geq 0$. This holds true by (42) if and only if $\delta_1 = -\operatorname{sgn} \sin(\varphi_o - \gamma)$. In addition, we need to show that $\lambda_1 \lambda_2 \geq 0$, which is equivalent to

$$r_o^2 \frac{\sin(\varphi_o - \gamma - \delta_1 \phi) \sin(\varphi_o - \gamma + \delta_1 \phi)}{\sin^2 \delta_1 \phi} \geq 0,$$

or, using trigonometric identities,

$$\sin^2(\varphi_o - \gamma) \cos^2 \phi - \cos^2(\varphi_0 - \gamma) \sin^2 \phi \geq 0,$$

which means that

$$\tan^2(\varphi_o - \gamma) \geq \tan^2 \phi = \frac{1 - \rho^2}{\rho^2}. \tag{46}$$

A similar analysis of Δ_d provides

$$\mu_1 = r_d \frac{\sin(\varphi_d - \gamma - \delta_2 \phi)}{\sin 2\delta_2 \phi}, \tag{47}$$

$$\mu_2 = r_d \frac{\sin(\varphi_d - \gamma + \delta_2\phi)}{\sin 2\delta_2\phi}. \tag{48}$$

For $\mu_1 + \mu_2 \geq 0$ it is necessary and sufficient that $\delta_2 = \operatorname{sgn}\sin(\varphi_d - \gamma)$, and $\mu_1\mu_2 \geq 0$ is equivalent to

$$\tan^2(\varphi_d - \gamma) \geq \tan^2\phi = \frac{1-\rho^2}{\rho^2}. \tag{49}$$

We have now expressed λ_k and μ_k for $k = 1, 2$ in terms of the initial data and γ. To express σ also in the same terms, and to solve for γ, we analyse Δ_1 as follows.

$$\Delta_1 = d_1 - o_1 = \sigma e^{i\gamma} + \mu_1 e^{i(\gamma - \delta_2\phi)} + \lambda_1 e^{i(\gamma-\delta_1\phi)} = r_1 e^{i\varphi_1}.$$

This implies

$$r_1 e^{i(\varphi_1 - \gamma)} = \sigma + \mu_1 e^{-i\delta_2\phi} + \lambda_1 e^{-i\delta_1\phi}.$$

If we take the real part and rearrange terms, we get

$$\sigma = r_1\cos(\varphi_1 - \gamma) - (\lambda_1 + \mu_1)\cos\phi. \tag{50}$$

If we take the imaginary part, we arrive at

$$r_1 \sin(\varphi_1 - \gamma) = -\mu_1 \sin\delta_2\phi - \lambda_1\sin\delta_1\phi.$$

If we insert (44) and (47) in this equation, and use trigonometric identities, we get
$A\cos\gamma = B\sin\gamma$ with

$$A = 2r_1\sin\varphi_1\cos\phi + r_d\sin(\varphi_d - \delta_2\phi) - r_o\sin(\varphi_o - \delta_1\phi), \tag{51}$$

$$B = 2r_1\cos\varphi_1\cos\phi + r_d\cos(\varphi_d - \delta_2\phi) - r_o\cos(\varphi_o - \delta_1\phi), \tag{52}$$

and hence

$$\gamma = \arctan\frac{A}{B} + \delta_3\pi, \tag{53}$$

where $\delta_3 \in \{-1, 1\}$. This proves the theorem. □

Observe that when γ is replaced by $\gamma + \pi$, equations (44) - (50) change signs. This will be used, together with choice of δ_1 and δ_2 to ensure that λ_k and μ_k are non-negative for $k = 1, 2$, while sign of σ may be negative in case platooning distance is larger than direct trucking distances.

We now finally consider the cases when $o_k = m$ or $s = d_k$ for some $k \in \{1, 2\}$, say $k = 1$. We discuss the case of $o_1 = m$ in detail. The other cases follow similar lines. If $o_1 = m$, then

$$L(m, s) = L(s) = \|o_1 - o_2\|_2 + 2\rho\|o_1 - s\|_2 + \sum_k \|s - d_k\|_2,$$

and we get

$$\frac{\partial L}{\partial s_j} = 2\rho\frac{s_j - o_{1j}}{\|s - o_1\|_2} + \sum_k \frac{s_j - d_{kj}}{\|s - d_k\|_2} = 0, \quad j = 1, 2.$$

This implies, with $\beta_k = \angle(s - d_k)$ and $\gamma = \angle(o_1 - s)$, that $\beta_1 = \pi + \delta\phi + \gamma$ and $\beta_2 = \pi - \delta\phi + \gamma$ for some $\delta \in \{1, 2\}$. We may write $L = r_0 + 2\rho\sigma + \mu_1 + \mu_2$ with μ_1 as in (47) and μ_2 as in (48). Note that

$$\Delta_1 = d_1 - o_1 = \sigma e^{i\gamma} - \mu_1 e^{i\beta_1} = \sigma e^{i\gamma} + \mu_1 e^{-i\delta+\gamma} = r_1 e^{i\varphi_1},$$

so

$$r_1 e^{i\varphi_1+\gamma} = \sigma + \mu_1 e^{-i\phi}.$$

Taking the real part provides $r_1 \cos(\varphi_1 - \gamma) = \sigma + \mu_1 \cos\phi$, and hence

$$\sigma = r_1 \cos(\varphi_1 - \gamma) - \mu_1 \cos\phi.$$

Taking the imaginary part results in $r_1 \sin(\varphi_1 - \gamma) = -\mu_1 \sin\delta\phi$, we get, after substitution of the RHS of (47) and trigonometric manipulations,

$$\{2r_1 \sin\varphi_1 \cos\phi + r_d \sin(\varphi_d - \delta\phi)\} \cos\gamma = \{2r_1 \cos\varphi_1 \cos\phi + r_d \cos(\varphi_d - \delta\phi)\} \sin\gamma.$$

With

$$A = 2r_1 \sin\varphi_1 \cos\phi + r_d \sin(\varphi_d - \delta\phi),$$

$$B = 2r_1 \cos\varphi_1 \cos\phi + r_d \cos(\varphi_d - \delta\phi),$$

we arrive at

$$\gamma = \arctan\frac{A}{B} + \delta'\pi,$$

with $\delta' \in \{-1, 1\}$.

Proof of Theorem 4. Given that $o_1 = m + \lambda_1 u_{-\delta_1}$, $o_2 = m + \lambda_2 u_{+\delta_1}$, $d_1 = m + \sigma v - \mu_1 u_{-\delta_2}$, and $d_2 = m + \sigma v - \mu_2 u_{+\delta_2}$, with $v = -\frac{1}{2\rho}(u_{-1} + u_{+1})$, we may write any convex combination (we write $\theta = \frac{\sigma}{2\rho}$) as

$$x = a_1 o_1 + a_2 o_2 + b_1 d_1 + b_2 d_2 =$$

$$m + a_1\lambda_1 u_{-1} + a_2\lambda_2 u_{+1} - (b_1 + b_2)\theta(u_{-1} + u_{+1}) - b_1\mu_1 u_{+1} - b_2\mu_2 u_{-1} =$$

$$m + \{a_1\lambda_1 - (b_1 + b_2)\theta - b_2\mu_2\}u_{-1} + \{a_2\lambda_2 - (b_1 + b_2)\theta - b_1\mu_1\}u_{+1}.$$

We have assumed here that $\delta_1 = +1$ and $\delta_2 = -1$. First, we find a convex combination for which $x = m$, which is a solution to

$$\begin{pmatrix} \lambda_1 & 0 & -\theta & -\theta - \mu_2 \\ 0 & \lambda_2 & -\theta - \mu_1 & -\theta \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Gaussian elimination gives

$$\begin{pmatrix} 1 & 0 & 0 & -z_{22}/w_{11} \\ 0 & 1 & 0 & z_{11}/w_{11} \\ 0 & 0 & 1 & w_{22}/w_{11} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \lambda_2\theta/w_{11} \\ \lambda_1(\mu_1+\theta)/w_{11} \\ \lambda_1\lambda_2/w_{11} \end{pmatrix},$$

with $(i, j \in \{1,2\})$

$$w_{ij} = \lambda_i\mu_j + \lambda_1\lambda_2 + \lambda_1\theta + \lambda_2\theta,$$

$$z_{ij} = \lambda_i\mu_j + \mu_1\mu_2 + \mu_1\theta + \mu_2\theta.$$

If we put $b_2 = 0$, we find that $a_1 = \lambda_2\theta/w_{11}$, $a_2 = \lambda_1(\mu_1+\theta)/w_{11}$, and $b_1 = \lambda_1\lambda_2/w_{11}$ are indeed coefficients of a convex combination, i.e., $a_1, a_2, b_1 \in [0,1]$ and $a_1 + a_2 + b_1 = 1$.
Second, we find a convex combination for which $x = s = m + \sigma v$, which is a solution to

$$\begin{pmatrix} \lambda_1 & 0 & -\theta & -\theta-\mu_2 \\ 0 & \lambda_2 & -\theta-\mu_1 & -\theta \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -\theta \\ -\theta \\ 1 \end{pmatrix}.$$

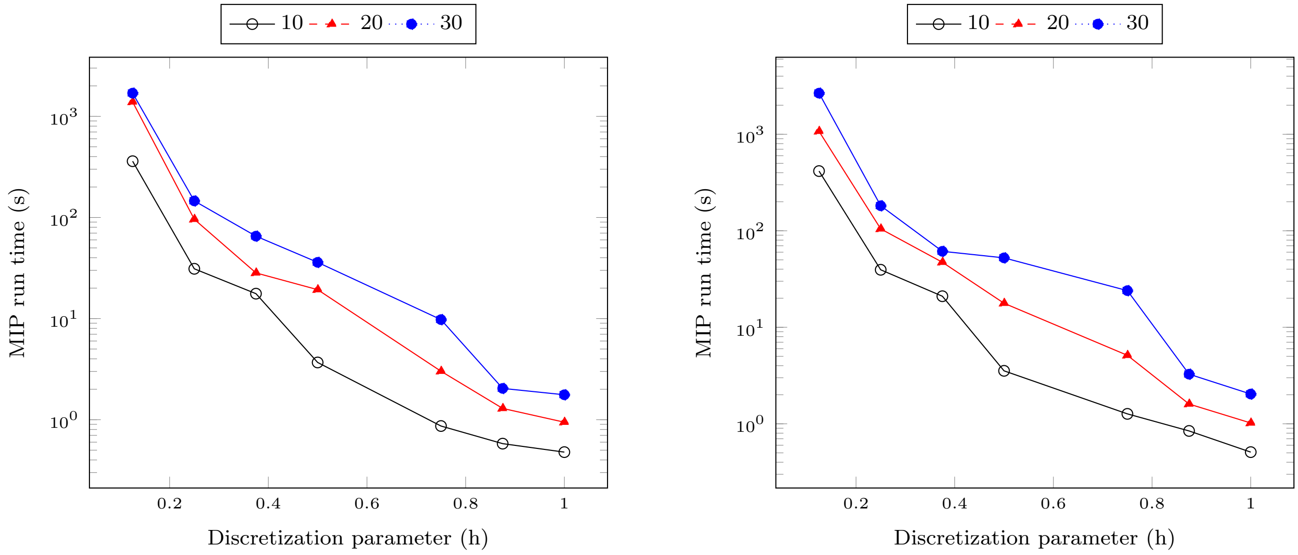
Gaussian elimination now gives

$$\begin{pmatrix} 1 & 0 & 0 & -z_{22}/w_{11} \\ 0 & 1 & 0 & z_{11}/w_{11} \\ 0 & 0 & 1 & w_{22}/w_{11} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} -\mu_1\theta/w_{11} \\ \mu_1(\lambda_1+\theta)/w_{11} \\ (\lambda_1\lambda_2+\lambda_1\theta+\lambda_2\theta)/w_{11} \end{pmatrix}.$$

If we put $b_2 = \mu_1\theta/z_{22}$, we find that $a_1 = 0$, $a_2 = \mu_1\mu_2/z_{22}$, and $b_1 = \mu_2(\lambda_2+\theta)/z_{22}$ are indeed coefficients of a convex combination, i.e., $a_2, b_1, b_2 \in [0,1]$ and $a_2 + b_1 + b_2 = 1$. We would have arrived at the same outcomes in the case when $\delta_1 = +1$ and $\delta_2 = -1$, as the roles of u_+ and u_- would have been switched.
We now do the analysis for $\delta_1 = \delta_2 = 1$ (and the analysis of $\delta_1 = \delta_2 = -1$ will yield the same results). Similar to above, if we put $x = m$ we can do the Gaussian elimination and arrive at $b_2 = 0$, $a_1 = \lambda_2(\theta+\mu_1)/w_{21}$, $a_2 = \lambda_1\theta/w_{21}$, and $b_1 = \lambda_1\lambda_2/w_{21}$, which again yields a convex combination.
If we put $x = m + \sigma v$, a convex combination is provided by $a_1 = 0$, $a_2 = \mu_1\mu_2/z_{21}$, $b_1 = \mu_2\theta/z_{21}$, and $b_2 = \mu_1(\lambda_2+\theta)/z_{21}$. This proves the theorem. □

B Impact of the discretization parameter

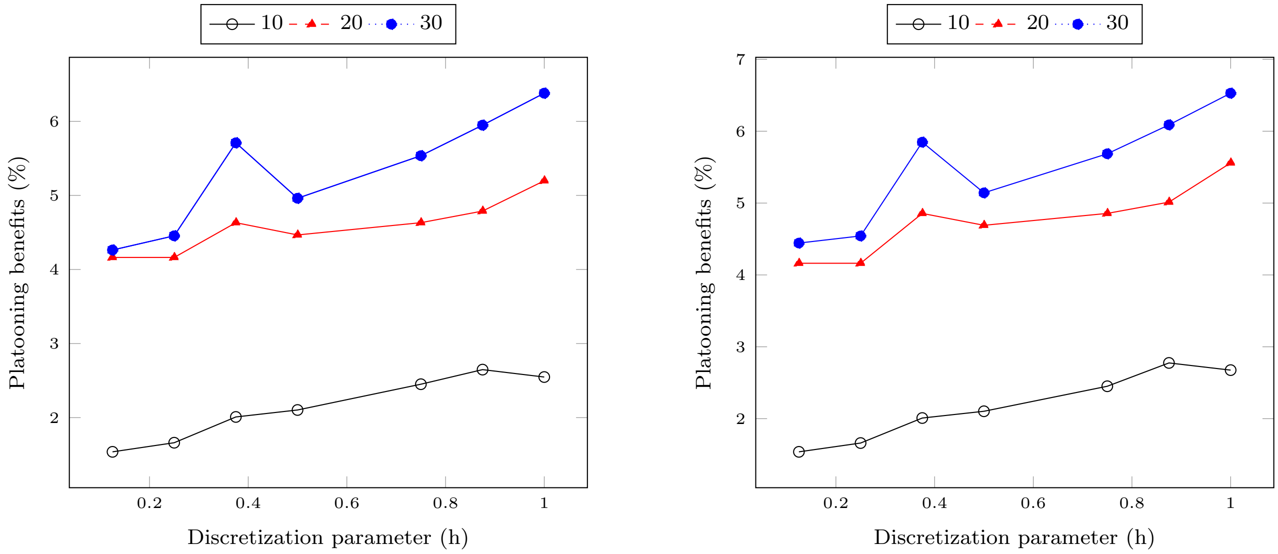
A finer discretization would provide a higher quality solution that is closer to the continuous time case but lead to an increase in the problem size, potentially to the point of intractability. On the



(a) $(2, \infty)$

(b) $(3, \infty)$

Figure 13: Run times for differing discretization parameters



(a) $(2, \infty)$

(b) $(3, \infty)$

Figure 14: Percentage benefits for differing discretization parameters; 30 trucks; $\rho = 0.9$

flip side, a less fine discretization level would keep the problem size in check while compromising on the solution quality. Here, we check the effect of the discretization parameter on the system savings. To do so, we run the MIP from phase 1 for different discretization parameters between 7.5 minutes and 60 minutes. We report percentage savings for the $(3, \infty)$ and $(2, \infty)$ settings in Figure 14 and run times in Figure 13. Note that we have used a logarithmic y-axis in Figure 13. From the run times in Figure 13, the increase in problem size with a finer level of discretization for both for both settings is evident. This increase is especially pronounced and becomes larger as we continue lowering the values of the discretization parameter. For example, the $(3, \infty)$ run time increases by a factor of fifteen from about 180 seconds to about 2700 seconds. This huge increase does not have much of an effect on the percentage benefits as we can see in Figure 14. This holds true for $(2, \infty)$ as well. Based on this, we use a discretization parameter of 0.25 in our experiments.