EUR Research Information Portal

Platoon Optimization Based on Truck Pairs

Published in:
INFORMS Journal on Computing

Publication status and date:
Published: 01/06/2023

DOI (link to publisher):
10.1287/ijoc.2020.0302

Document Version
Publisher's PDF, also known as Version of record

Document License/Available under:
Article 25fa Dutch Copyright Act

Citation for the published version (APA):
Bhoopalam, A. K., Agatz, N., & Zuidwijk, R. (2023). Platoon Optimization Based on Truck Pairs. *INFORMS Journal on Computing*, 1-19. https://doi.org/10.1287/ijoc.2020.0302

Link to publication on the EUR Research Information Portal

Terms and Conditions of Use
Except as permitted by the applicable copyright law, you may not reproduce or make this material available to any third party without the prior written permission from the copyright holder(s). Copyright law allows the following uses of this material without prior permission:

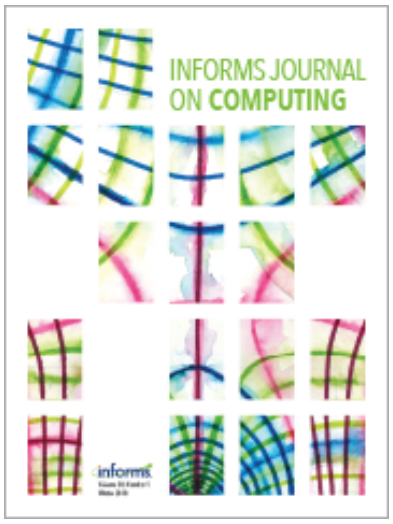
- you may download, save and print a copy of this material for your personal use only;
- you may share the EUR portal link to this material.

In case the material is published with an open access license (e.g. a Creative Commons (CC) license), other uses may be allowed. Please check the terms and conditions of the specific license.

Take-down policy
If you believe that this material infringes your copyright and/or any other intellectual property rights, you may request its removal by contacting us at the following email address: openaccess.library@eur.nl. Please provide us with all the relevant information, including the reasons why you believe any of your rights have been infringed. In case of a legitimate complaint, we will make the material inaccessible and/or remove it from the website.

Erasmus University Rotterdam
Making Minds Matter

This article was downloaded by: [145.5.176.14] On: 02 May 2024, At: 02:48
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA



INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Platoon Optimization Based on Truck Pairs

Anirudh Kishore Bhoopalam, Niels Agatz, Rob Zuidwijk

To cite this article:
Anirudh Kishore Bhoopalam, Niels Agatz, Rob Zuidwijk (2023) Platoon Optimization Based on Truck Pairs. INFORMS Journal on Computing 35(6):1242-1260. https://doi.org/10.1287/ijoc.2020.0302

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2023, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 35, No. 6, November–December 2023, pp. 1242–1260
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Platoon Optimization Based on Truck Pairs

Anirudh Kishore Bhoopalam,[a,*] Niels Agatz,[a] Rob Zuidwijk[a]

[a] Rotterdam School of Management, Erasmus University, 3062 PA Rotterdam, Netherlands
*Corresponding author
Contact: kishorebhoopalam@rsm.nl, https://orcid.org/0000-0003-3463-192X (AKB); nagatz@rsm.nl, https://orcid.org/0000-0003-3514-201X (NA); rzuidwijk@rsm.nl, https://orcid.org/0000-0002-1493-3974 (RZ)

Received: November 25, 2020
Revised: January 21, 2022; October 27, 2022; February 15, 2023; May 24, 2023
Accepted: May 24, 2023
Published Online in Articles in Advance: June 28, 2023

https://doi.org/10.1287/ijoc.2020.0302

Copyright: © 2023 INFORMS

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 associated with minimum detours and delays. In this paper, we focus on developing optimization technology to form truck 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 PRP-TW with two-truck platoons. Based on these special cases, we build several fast heuristics. An extensive set of numerical experiments shows that our heuristics perform well. Moreover, we show that simple two-truck platoons already capture most of the potential savings of platooning.

History: Accepted by Pascal van Hentenryck, Area Editor for Computational Modeling: Methods and Analysis.
Funding: This work was supported by the Netherlands Organization for Scientific Research (NWO) as part of the Spatial and Transport Impacts of Automated Driving [Grant 438-15-161] project.
Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplemental Information (https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2020.0302) as well as from the IJOC GitHub software repository (https://github.com/INFORMSJoC/2020.0302). The complete IJOC Software and Data Repository is available at https://informsjoc.github.io/.

Keywords: platooning • vehicle routing • optimization • matching • truck

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 to both cost savings and lower vehicle emissions (Scora and Barth 2006). Other benefits of platooning include improved road capacity utilization and traffic throughput (Van Arem et al. 2006, Schladover et al. 2015, Lioris et al. 2017). Moreover, as the following trucks break and steer automatically with the leader truck in front, the risk of rear-end collisions is reduced. Labor cost savings may also be possible when the technology allows the following trucks in a platoon to operate without (attentive) drivers (Kilcarr 2016).

Truck manufacturers across the world have conducted real-world tests with truck platooning (Bergenhem et al. 2012, Bhoopalam et al. 2018). Equally important are the challenges related to the planning and execution of platoons. Careful planning is especially important in the early implementation phases in which there are few trucks with compatible platooning technology. With low truck densities, it is unlikely that platoons emerge spontaneously (Liang et al. 2014, Sokolov et al. 2017). There is limited room for detours as they offset the small potential fuel savings, that is, between 4% and 13% (McAuliffe et al. 2018).

The planning of platoons involves determining the paths and time schedules of a set of trucks from their different origins to destinations so as to minimize the system-wide travel costs. We refer to the associated planning problem as the platoon routing problem with time windows (PRP-TW). We define a platoon as a set of two or more virtually connected trucks that drive together at short headways. Legal restrictions likely limit the size of a platoon because larger platoons may disrupt traffic flows at highway ramps (van Maarseveen 2017, Calvert et al. 2019). For example, the maximum platoon size was two during the tests in Kentucky and Missouri (Goble 2018).

One truck can potentially join different platoons along its journey. This creates interdependencies in time and space between different platoons. These interdependencies make the planning and coordinating of platoons more challenging. Larsson et al. (2015) show that the platoon routing problem is NP-hard even without deadlines and when all trucks start at the same origin.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

In this paper, we model and solve the PRP-TW but specifically focus on finding fast solution methods for problem settings with two-truck platoons in which all trucks are part of at most one platoon per trip. Here, we also look at the case without time windows to gain further insights. As the two-truck setting only requires coordination between two trucks, the issue of interdependencies is smaller, and we, therefore, believe that these platoons are easier to implement in practice. That is, a truck can wait without creating a cascade effect on the schedules of trucks in subsequent platoons.

The main contributions of this paper can be summarized as follows. First, we present polynomial-time algorithms to match trucks in two-truck platoons. Second, we introduce a novel mathematical programming formulation for 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 two-truck platoon setting. By doing so, we integrate and contribute to two literature streams that have thus far been disconnected: the literature on multitruck platoon planning systems (Larsson et al. 2015, Larsen et al. 2019, Luo and Larson 2021) and the stream on route planning of platoons between two trucks (Liang et al. 2013, Zhang et al. 2017, Xu et al. 2023). Finally, we conduct numerical experiments to evaluate the heuristics and the advantages and disadvantages of platoon restrictions on a large number of instances.

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 (MIP) formulation for the PRP-TW. In Section 5, we focus on various two-truck settings. Section 6 presents two heuristics and an improvement procedure for the PRP-TW based on the insights from the two-truck settings. 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 is a growing interest in the planning and operations of automated vehicles on the road (Mahmassani 2016), on water (Colling and Hekkenberg 2020), and in the air (Richert and Cortés 2013). In this section, we restrict ourselves to road-based platooning and discuss the relevant and recent literature on truck platoon planning. We refer to Bhoopalam et al. (2018) for an extensive overview.

A number of papers focus on platoon planning for trucks traveling on fixed paths, typically a single highway lane (Boysen et al. 2018) or a designated hub (Larsen et al. 2019, Adler et al. 2020). The papers in this area focus on speed control (Liang et al. 2013, Sun and Yin 2019), different platooning cost functions (Boysen et al. 2018), travel time uncertainty (Zhang et al. 2017), and dynamic planning (Van de Hoef 2018).

Similar to ours, other papers consider the path planning of trucks in the context of platooning. Luo and Larson (2021) build on the idea that trucks cannot deviate much from their shortest paths to form platoons. They propose an iterative route first, schedule second approach to plan platoons and develop a set of valid inequalities for both steps to reduce the problem size. Larson et al. (2013) set a threshold for possible truck detours. They consider a traffic junction and evaluate whether it is beneficial for approaching trucks to adjust their speed(s) to form a platoon. They propose a pairwise merging heuristic in which they continuously merge the most beneficial platoon pairs to generate multitruck platoons and show that the platooning savings increase in the case with detours.

Larsson et al. (2015) use a similar 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 that groups trucks based on their likelihood of traversing a certain edge. They formulate an MIP to evaluate their heuristics. In their subsequent work, Larson et al. (2016) focus on reducing the problem size by restricting the detour. 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 parts of the network. Nourmohammadzadeh and Hartmann (2016) also formulate an MIP that they use to evaluate a genetic algorithm-based meta-heuristic and then an ant-colony optimization meta-heuristic (Nourmohammadzadeh and Hartmann 2019).

Xu et al. (2022) study the restricted problem of planning paths and platoons for two trucks in isolation. In particular, they consider the problem of determining the paths, speeds, and schedules for two trucks with arrival deadlines. They show that the problem is weakly NP-hard and propose a dual subgradient algorithm to solve it. Their experimental results suggest that routing and scheduling have a much larger impact on potential platoon savings than speed control.

Xu et al. (2023) and Albinski et al. (2020) study the impact of introducing driver rest time regulations. Xu et al. (2023) propose an iterated neighborhood search heuristic to solve the problem. Their experiments show that mandatory breaks can lead to different truck path solutions. Albinski et al. (2020) also formulate an MIP and present a preprocessing procedure to reduce the problem size. They run computational experiments with different levels of driver involvement and show that the platoon savings can vary greatly.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

3. Problem Description and Properties

In this section, we define the PRP-TW and describe a couple of relevant theoretical properties.

3.1. Problem Description

The PRP-TW involves routing a set of trucks from their origins to their destinations in a way that minimizes system-wide travel costs by platooning, satisfying time window constraints. To form platoons—saving fuel costs and emissions—trucks need to align their routes in time and space. This specifically concerns decisions related to (i) determining the routes of all trucks between their origins and destinations; (ii) determining departure times of all trucks and potential waiting times along their routes; and (iii) identifying platoons between trucks, including the meet and split locations and times. The following constraints must be taken into account: (i) time window restrictions on the earliest departure time from the origin and the latest arrival time at the destination and (ii) a maximum platoon size.

We model the PRP-TW on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with nonnegative arc lengths. The set of vertices $\mathcal{V}$ represents the locations, that is, the possible origins and destinations of trucks. The set of arcs $\mathcal{A}$ represents the road segments that connect the different locations. We have a set of trucks $\mathcal{K}$ in which truck $k \in \mathcal{K}$ has origin $o_k \in \mathcal{V}$ and destination $d_k \in \mathcal{V}$. Let c_a or c_{uv} denote the cost of traversing arc a or $(u, v) \in \mathcal{A}$ outside of a platoon. Trucks that drive in a platoon can save fuel. The leader typically saves less than the followers (Winder 2016). Let ρ_l be the fuel consumption factor for a platoon leader and ρ_f the average fuel consumption factor for a follower ($\rho_l > \rho_f$). This means that $1 - \rho_l$ represents the fuel savings for a leader and $1 - \rho_f$ for a follower as compared with driving alone. As is common in the literature (Albinski et al. 2020, Luo and Larson 2021), we set the total combined cost of an n-truck platoon traversing a to be $c_a^n = (\rho_l + (n-1) \cdot \rho_f) \cdot c_a$. Note that $\rho_l, \rho_f \in (0.5, 1)$ as the total combined fuel consumption of a two-truck (or larger) platoon cannot be less than the fuel consumption of a truck driving alone.

Each truck has an earliest departure time a_k from its origin and a latest arrival time l_k at its destination. Let t_{uv} be the travel time needed to travel between nodes $u, v \in \mathcal{V}$. Therefore, to meet its deadline l_k, a truck $k \in \mathcal{K}$ has to depart at the latest by $l_k - t_{o_k d_k}$. We call the difference between this latest departure time and the earliest departure time a_k the time flexibility $f_k = l_k - t_{o_k d_k} - a_k$. This flexibility specifies the maximum possible waiting and detour times within a trip. Moreover, the maximum platoon size N specifies the maximum number of trucks in a platoon.

We make the following assumptions in our model:

1. Each truck travels at a given constant speed. This is a reasonable assumption as trucks typically drive at the general speed of the traffic flow.
2. Trucks can only start and end a platoon at a node. This also implies that trucks can only wait at a node to form a platoon.
3. We do not consider waiting costs, but consider hard time windows on the arrival time.
4. For ease of exposition, we ignore any fixed setup costs in forming a platoon. This models a setting in which platoon formations are fast and take place automatically.

3.2. Basic Properties

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

Let $\bar{c}_{uv}$ denote the costs of the least cost path between nodes $u, v \in \mathcal{V}$. Let ρ represent the average savings per truck in a platoon, that is, $\rho = (\rho_l + \rho_f)/2$.

Property 1. *Given an average minimum platoon savings 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* $\rho < \frac{\sum_{k \in \mathcal{K}'} \bar{c}_{o_k d_k} - \sum_{k \in \mathcal{K}'} \bar{c}_{o_k m}}{\sum_{k \in \mathcal{K}'} \bar{c}_{m d_k}}$.

Proof. Consider the case in which the trucks all meet at a single m. In this case, their platooning costs are at least $\sum_k \bar{c}_{o_k m} + \rho \cdot \sum_k \bar{c}_{m d_k}$. For this platooning solution to be beneficial, we need

$$\sum_k \bar{c}_{o_k m} + \rho \cdot \sum_k \bar{c}_{m d_k} < \sum_k \bar{c}_{o_k d_k}$$

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

$$\rho < \frac{\sum_k \bar{c}_{o_k d_k} - \sum_k \bar{c}_{o_k m}}{\sum_k \bar{c}_{m d_k}} \quad \square$$

Note that a similar property holds for split point s: $\rho < \frac{\sum_k \bar{c}_{o_k d_k} - \sum_k \bar{c}_{s d_k}}{\sum_k \bar{c}_{o_k s}}$.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

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

Property 2. *Node m is not a feasible meeting point for a platoon $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| > 1$ if at least one truck in $\mathcal{K}'$ cannot reach its destination before its latest arrival time after departing from m at the earliest time for all trucks in the platoon; that is, for any truck $i \in \mathcal{K}'$, $\max_{k\in\mathcal{K}'}(a_k + t_{o_k m}) + t_m d_i < l_i$.*

Proof. Consider a set of trucks $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| > 1$ that form a platoon together at node m. They can only depart as a platoon from m after all trucks have arrived at $\max_{k\in\mathcal{K}'}(a_k + t_{o_k m})$. If, for any truck $i \in \mathcal{K}'$, the shortest time from m to its destination d_i would go beyond its deadline l_k, it is infeasible, which means that m cannot be a meeting point.

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.

Property 3. *If the cost of an n-truck platoon traversing an arc a is $(\rho_l + (n-1)\cdot\rho_f)\cdot c_a$, where $\rho_l, \rho_f > 0$ and the maximum platoon size $N \geq 3$, all trucks that jointly traverse an arc at the same time are part of a platoon in the optimal solution, that is, there is never a truck driving by itself outside of a platoon.*

Proof. Consider k trucks that jointly traverse an arc a at the same time. If we matched into platoons all of size N, there is $\lfloor k/N \rfloor$. This leaves $k \bmod N$ unmatched trucks. When $k \bmod N \geq 2$, they can be all be in a platoon of size $< N$ together to save costs. When $k \bmod N = 1$, breaking up one of the N-truck platoons to allow the solitary truck to also be part of a smaller platoon reduces costs because $\rho_l > 0$. That is, trucks jointly traversing arcs can always be organized into platoons, which brings down total costs as compared with some of them driving alone. Therefore, in an optimal solution, all these trucks are 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 aim to find the minimum-cost paths of all trucks in the system taking platooning into account. Here, we focus on both the routing and scheduling of platoons and do not separate the decisions as in Luo and Larson (2021). In the second phase, we assign trucks to specific platoons associated with these paths.

4.1. Phase 1: Finding Paths of Trucks and Platoons

The PRP-TW involves the synchronization of truck routes in both space and time. To add the element of time, we modify $\mathcal{G}$ 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 $\mathcal{T} = \{0, \triangle, 2\triangle, 3\triangle, \ldots\ldots, H\triangle\}$. Note that the total number of time periods H is dictated by the latest l_k for truck $k \in \mathcal{K}$ as $H = \lceil max_{k\in\mathcal{K}} l_k/\triangle \rceil$. 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 $|\mathcal{V}_\mathcal{T}| = (H+1)\cdot|\mathcal{V}|$.

As in 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 $j = i + \lceil t_w/\triangle \rceil$. 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, that is, the truck waits.

Trucks only incur costs when they travel spatially; that is, we do not consider a cost associated with traveling on the waiting arcs. Recall that the cost of traveling on arc a outside of a platoon is c_a.

Let binary variable x_{ak}^{nt} be one when truck k begins traversing arc a at time t. When $n > 1$, it indicates that truck k is in a platoon of size n. We also define a variable y_a^{nt} to denote the overall truck and platoon flows on arc a at t. For $n = 1$, y_a^{nt} is the number of trucks that start traversing arc a at t outside of a platoon at cost c_a. For $n > 1$, y_a^{nt} is the number of n-platoons that start traversing arc a at t with costs of c_a^n per platoon. If we fix the values of y_a^{nt}, the model can be characterized as a continuous-time service network design problem by using, for practical purposes, a one-minute discretization (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.

$$\min \sum_{a\in\mathcal{A}_\mathcal{T}} \sum_{t\in\mathcal{T}} \sum_{n} c_a^n y_a^{nt}, \tag{1}$$

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

$$\sum_k x_{ak}^{nt} \le n \cdot y_a^{nt} \qquad \forall a \in \mathcal{A}_T, 1 < n \le N, \tag{3}$$

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

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

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. This problem is a generalization of the NP-complete integer-valued multicommodity flow problem (Even et al. 1975) and is, therefore, also NP-complete.

4.2. Phase 2: Finding Specific Trucks and Platoons per Path

Phase 1 provides the aggregate flow of trucks from their origins to their destinations. That is, 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 the specific platoons. For example, phase 1 may indicate that four trucks $\{1,2,3,4\}$ travel in two platoons without specifying the specific platoons, for example, {1, 3} and {2, 4}. Phase 2 aims to generate these specific platoons based on the output of phase 1.

There are many different possible specific platoons that can be created from the solution of phase 1. Among these, we aim to find the solution that minimizes the inconvenience for trucks in terms of the number of switches. Let $\mathcal{K}_a^{nt} = \{i \in \mathcal{K} : x_{ai}^{nt} = 1\}$ be the set of trucks on platoon arc a at time t in a platoon of size n. We write $\mathcal{K}_a^t = \cup_{1<n\le N} \mathcal{K}_a^{nt}$ as the set of trucks on platoon arc a at time t in a platoon of arbitrary size but not larger than the maximum platoon size N. We may and do suppress the time dimension t when no confusion should arise; for solutions obtained in phase 1, the time dimension has been properly accounted for.

For now, we consider the case of $n = 2$. Consider the graph $G = (V, A)$. For each arc $a \in A$, we define the set $\mathcal{K}_a$ of trucks that travel in a platoon on that arc. We assume that, for all $a \in A$, $\mathcal{K}_a \neq \emptyset$ and that $\mathcal{K} = \cup_{a\in A} \mathcal{K}_a$. For each subset $\mathcal{L} \subseteq \mathcal{K}$ of trucks, we define

$$A(\mathcal{L}) = \{a \in A : \mathcal{L} \subseteq \mathcal{K}_a\}.$$

We observe that, for a pair $\pi \subseteq \mathcal{K}$, $A(\pi)$ is a union of paths, and it need not be a single path. We also define $\mathcal{A}_a = \{\pi : \pi \subseteq \mathcal{K}_a, |\pi| = 2\}$ as the set of all pairs in $\mathcal{K}_a$ for $a \in A$, and define $\mathcal{A} = \cup_{a\in A} \mathcal{A}_a$.

Claim 1. *We have*

$$\mathcal{K}_a \subseteq \bigcup_{a\in A(\pi)} \pi = \bigcup_{\pi\in\mathcal{A}_a} \pi, \text{ for all } a \in A.$$

Proof. To prove the left-hand inclusion, we note that, if $k \in \mathcal{K}_a$, then there exists a platoon pair $\pi \ni k$ with $a \in A(\pi)$. To prove the right-hand equality, observe that $a \in A(\pi)$ is equivalent to $\pi \subseteq \mathcal{K}_a$, which means by definition that $\pi \in \mathcal{A}_a$. □

A set $\mathcal{C} \subseteq \mathcal{A}$ of pairs in $\mathcal{K}$ is called a cover whenever

$$\mathcal{K}_a \subseteq \bigcup_{\pi\in\mathcal{C}_a} \pi, \text{ for all } a \in A,$$

where $\mathcal{C}_a = \mathcal{C} \cap \mathcal{A}_a$ for $a \in A$. A cover can be associated with a number of switches. Each distinct platoon pair merges and dissolves, both of which represent switches. In this manner, the number of switches equals twice the size of the cover $|\mathcal{C}|$ as this number represents the number of distinct platoon pairs in the cover. To minimize the number of switches, we seek to find a cover of minimum size.

Claim 2. *If $\mathcal{K}_a = \pi$ is a pair for some $a \in A$ and $\mathcal{C}$ is a cover, then $\pi \in \mathcal{C}$.*

Proof. Note that $\mathcal{A}_a = \{\pi\}$. Because $\mathcal{K}_a = \pi \subseteq \cup_{\pi'\in\mathcal{C}_a} \pi'$, we obtain $\pi \in \mathcal{C}$. □

We observe that this set cover problem can be seen as a so-called edge cover problem of a number of cliques. To specify this type of edge cover, we identify a pair of vertices with an edge and define the graph $\mathcal{G}$ with vertices $\mathcal{K}$ and edges $\mathcal{A}$. Moreover, the graph $\mathcal{G}_a = (\mathcal{K}_a, \mathcal{A}_a) \subseteq \mathcal{G}$ is a complete subgraph or clique. The set $\mathcal{C} \subseteq \mathcal{A}$ is an edge cover of the set of cliques $\{\mathcal{G}_a\}_{a\in A}$ when, for each $a \in A$, the set $\mathcal{C}_a$ is an edge cover of $\mathcal{G}_a$. This definition is indeed equivalent with the previous definition of a cover. The minimum edge cover problem of a graph is solvable in polynomial time and

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

even trivial for complete graphs or cliques; the question is whether the edge cover problem of a given set of cliques in a graph is solvable in polynomial time as well.

For a complete graph with an even number of nodes, which is the case for $\mathcal{G}_a$, a minimum edge cover coincides with a perfect matching. Perfect matchings in a clique are very easy to find, but there are many of them: for the complete graph $\mathbf{K}_{2n}$, the number of perfect matchings equals

$$\frac{(2n)!}{n!2^n}.$$

Proposition 1. *If $\mathcal{C}$ is an edge cover of $\{\mathcal{G}_a\}_{a\in A}$ of minimum size, then there exist perfect matchings $\mathcal{M}_a \subseteq \mathcal{C}$ of $\mathcal{G}_a$ such that $\mathcal{C} = \cup_{a\in A}\mathcal{M}_a$.*

Proof. By definition, $\mathcal{K}_a = \cup_{\pi\in\mathcal{A}_a\cap\mathcal{C}}\pi$. There exist perfect matchings $\mathcal{M}_a \subseteq \mathcal{C}$ of $\mathcal{G}_a$, which implies that $\mathcal{K}_a = \cup_{\pi\in\mathcal{M}_a}\pi$. As a result, $\mathcal{C}' = \cup_{a\in A}\mathcal{M}_a$ is an edge cover of $\{\mathcal{G}_a\}_{a\in A}$, whereas $\mathcal{C}' \subseteq \mathcal{C}$. Because $\mathcal{C}$ is of minimum size, this implies $\mathcal{C}' = \mathcal{C}$. □

We can formulate this cover problem as a set cover problem without reference to its specific structure by defining the multidimensional sets $\boldsymbol{S}(\pi)$, where $\boldsymbol{S}_a(\pi) = \{\pi\}$ when $\pi \in \mathcal{A}_a$ and $\boldsymbol{S}_a(\pi) = \emptyset$ otherwise. The set $\boldsymbol{K}$ is defined as $\boldsymbol{K}_a = \mathcal{K}_a$ for $a \in \mathcal{A}$. The set cover problem seeks to find a cover $\{\boldsymbol{S}(\pi) : \pi \in \mathcal{C}\}$ for $\boldsymbol{K}$. We solve this set cover formulation with a commercial MIP solver.

Although the minimum edge cover problem can be solved in polynomial time, our conjecture is that the phase 2 problem is NP-complete. In Online Appendix A, for the interested reader, we include some examples that help create a further understanding of the intricacies of edge covers of a set of cliques.

5. Special Cases with Truck Pairs

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 and platoons have a size of two. We denote this special case as the PRP-TW (2,1). For a platoon with two trucks, we can use a single fuel consumption factor ρ that represents the average consumption of the leader and follower, that is, $\rho = (\rho_l + \rho_f)/2$. Given all possible truck platoon pairs and their respective costs, we can model this problem as a general matching problem. These special cases provide insights and building blocks to develop heuristics for the PRP-TW. We refer to the general problem of finding the meeting and split point that maximizes the savings of a platoon between two specific trucks as the meet and split routing problem with time windows (MSRP-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 subproblem 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 PRP-TW (2,1) can be obtained in $\mathcal{O}(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|) + |\mathcal{K}|^3)$.*

Proof. Consider the MSRP-TW for a single pair of trucks $i, j \in \mathcal{K}$. Consider an arbitrary pair of nodes $u, v \in \mathcal{V}$ to be the meeting and split points. 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). Assessing the time feasibility of each path can be done in constant time. Doing this for all $|\mathcal{V}|^2$ node pairs and computing the minimum costs L_{ij}, therefore, requires $\mathcal{O}(|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|))$. The next step is to compute the L_{ij} for all pairs $i, j \in \mathcal{K}$. Because we have $|\mathcal{K}|^2$ truck pairs, this can be achieved in $\mathcal{O}(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|))$. 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, $\mathcal{O}(|\mathcal{K}|^2|\mathcal{V}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|) + |\mathcal{K}|^3)$. □

Next, we focus on a relaxed version of the PRP-TW (2,1) in which the time windows are not binding or the PRP (2,1). The associated problem of finding the costs for a single pair of trucks here is the MSRP.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

5.1. Routing of Platoon Pairs Without Time Windows

The solution of the problem of finding the costs for a truck pair without time windows (MSRP) is a lower bound to the problem with time windows (MSRP-TW). In practice, we expect the optimal solutions of this relaxed problem to usually also be time feasible as they are often associated with small detours. The insights we gain from analyzing the MSRP 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.

We can transform the MSRP into a shortest path problem that aims to find a single least cost 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, that is, $d_\mathcal{E}(S, v) = \sum_{k=1,2} d(o_k, v)$. 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, that is, $d_\mathcal{E}(u, v) = 2 \cdot \rho \cdot d(u, v)$. 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, that is, $d_\mathcal{E}(v, T) = \sum_{k=1,2} d(v, d_k)$. We depict this idea in Figure 1.

Proposition 2. *MSRP 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, 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 PRP (2,1).

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

Proof. Consider the MSRP for a single pair $i, j \in \mathcal{K}$. Adding source S and sink T can be done in constant time. Modeling the first leg 1 requires us to add an edge and compute 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 $\mathcal{O}(|\mathcal{V}|(|\mathcal{V}| + Alog|\mathcal{V}|))$. Modeling the joint platoon 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 the last leg 3 has the same complexity as modeling the first leg 1. Therefore, the total time complexity to build transformed graph $\mathcal{G}_\mathcal{E}$ for a pair $i, j \in \mathcal{K}$ is $\mathcal{O}(|\mathcal{V}|(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|))$. 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 $\mathcal{O}(|\mathcal{V}|(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|))$. Following this, we can find the minimum weight matching in $\mathcal{O}(|\mathcal{K}|^3)$ making the overall complexity $\mathcal{O}(|\mathcal{V}||\mathcal{K}|^2(|\mathcal{V}| + |\mathcal{A}|log|\mathcal{V}|) + |\mathcal{K}|^3))$. □

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

We now outline how we can use the MSRP solution method to solve the PRP-TW (2,1). Because the MSRP solution is a lower bound to the MSRP-TW, we do not need to solve the MSRP-TW for truck pairs $i, j \in \mathcal{K}$ that do not have a beneficial MSRP 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 MSRP-TW 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 precompute or use parallel computing for the different steps to build the platoon pair graph $\mathcal{G}_\mathcal{P}$.

5.2. MSRP 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

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

split point s that minimize the platoon costs. Once we have these points, the travel costs may be determined as follows:

$$L(m,s) = \sum_k \|o_k - m\|_2 + 2\rho\|m - s\|_2 + \sum_k \|s - d_k\|_2, \tag{6}$$

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

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

which implies that platooning is not beneficial. Also, when $\rho = 1$, platooning never results 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 then always is 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 Online Appendix B.

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*

$$L(m,s) = \sum_k \lambda_k + 2\rho\sigma + \sum_k \mu_k. \tag{7}$$

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,

$$\lambda_1 = -r_o \frac{\sin(\varphi_o - \gamma - \delta_1\phi)}{\sin 2\delta_1\phi}, \tag{8}$$

$$\lambda_2 = -r_o \frac{\sin(\varphi_o - \gamma + \delta_1\phi)}{\sin 2\delta_1\phi}, \tag{9}$$

$$\mu_1 = r_d \frac{\sin(\varphi_d - \gamma - \delta_2\phi)}{\sin 2\delta_2\phi}, \tag{10}$$

$$\mu_2 = r_d \frac{\sin(\varphi_d - \gamma + \delta_2\phi)}{\sin 2\delta_2\phi}, \tag{11}$$

and

$$\sigma = r_1 \cos(\varphi_1 - \gamma) - (\lambda_1 + \mu_1)\cos\phi. \tag{12}$$

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

$$\gamma = \arctan\frac{A}{B} + \delta_3\pi, \tag{13}$$

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

$$A = 2r_1 \cos\phi \sin\varphi_1 + r_d \sin(\varphi_d - \delta_2\phi) - r_o \sin(\varphi_o - \delta_1\phi), \tag{14}$$

$$B = 2r_1 \cos\phi \cos\varphi_1 + r_d \cos(\varphi_d - \delta_2\phi) - r_o \cos(\varphi_o - \delta_1\phi). \tag{15}$$

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

Figure 1. Transforming Two Paths into a Single Path

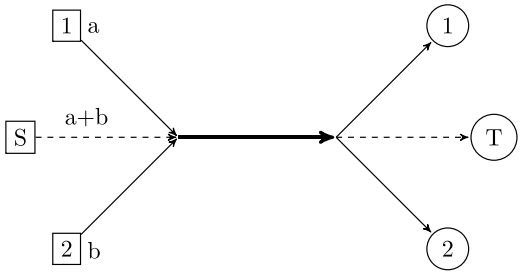
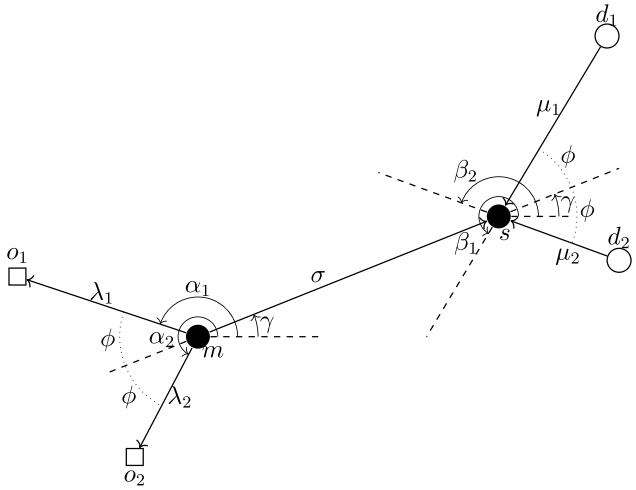


Figure 2. Platooning Distance Optimization in $\mathbb{R}^2$



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

$$\text{Conv}(\{x_i\}) = \left\{ \sum_i a_i x_i \middle| \sum_i a_i = 1 \right\}.$$

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, that is, $s, m \in \text{Conv}(\{o_1, o_2, d_1, d_2\})$. In particular, convex combinations are given when $\delta_1\delta_2 = -1$ by*

$$m = \frac{\lambda_2\theta}{w_{11}} o_1 + \frac{\lambda_1(\mu_1+\theta)}{w_{11}} o_2 + \frac{\lambda_1\lambda_2}{w_{11}} d_1, \tag{16}$$

$$s = \frac{\mu_1\mu_2}{z_{22}} o_2 + \frac{\mu_2(\lambda_2+\theta)}{z_{22}} d_1 + \frac{\mu_1\theta}{z_{22}} d_2, \tag{17}$$

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

$$m = \frac{\lambda_2(\theta+\mu_1)}{w_{21}} o_1 + \frac{\lambda_1\theta}{w_{21}} o_2 + \frac{\lambda_1\lambda_2}{w_{21}} d_1, \tag{18}$$

$$s = \frac{\mu_1\mu_2}{z_{21}} o_2 + \frac{\mu_2\theta}{z_{21}} d_1 + \frac{\mu_1(\lambda_2+\theta)}{z_{21}} d_2, \tag{19}$$

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 Online Appendix B.

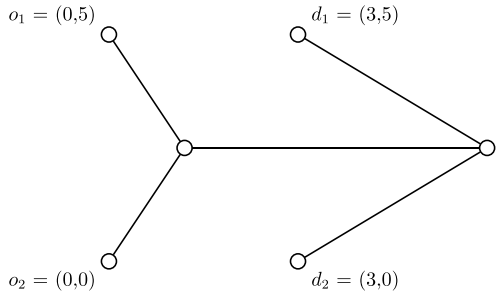
We now investigate how the solution of the MSRP on the Euclidean plane translates to the MSRP. This hinges on the density of the network underlying the MSRP. For instance, the Euclidean solution could indicate that the truck pair cannot feasibly form a platoon, but they might be able to on a sparse network.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

On the other hand, in a dense network, one expects the Euclidean solution to be a good approximation for the network solution. However, this is not necessarily the case. Consider the example in Figure 4 in which δ defines the grid density. In this grid, the lengths of shortest paths between the points (0,0) and (2,1) can be calculated as $1 + \sqrt{2}$, whereas the corresponding value on the Euclidean plane is $\sqrt{5}$. Note that this shortest path on the grid is independent of grid

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Figure 3. Example of a Sparse Network on Which the Trucks Automatically Platoon



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, whereas the corresponding costs on the network are 18.18. Comparing these figures to the 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 is 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 the 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 be o, s, and d_k. We can calculate the costs of each:

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

Because $\rho > 1/2$ for a two-truck platoon (see Section 3), we always have $L(o,d_k) > L(o,o)$, so the trucks 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 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 two-truck platooning solution first and then use an improvement procedure to potentially find additional (larger) platoons.

Figure 4. Shortest Paths on the Network and Euclidean Plane

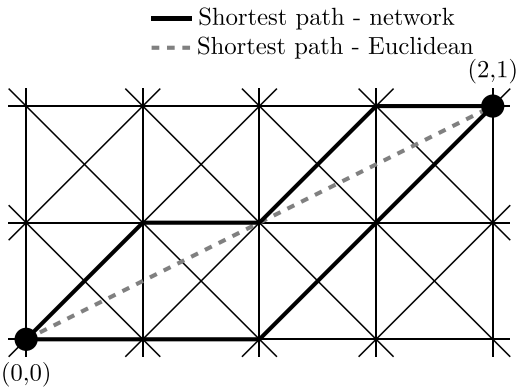
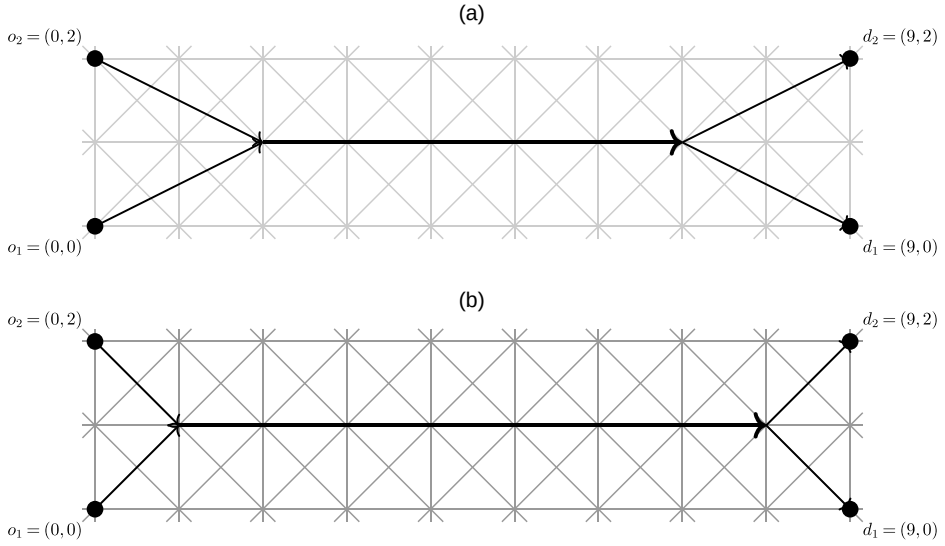


Figure 5. Example of the Optimal Platoon for a Truck Pair Being Feasible on the Euclidean Plane but Being Infeasible on a Dense Network; $\rho = \frac{2}{\sqrt{5}}$



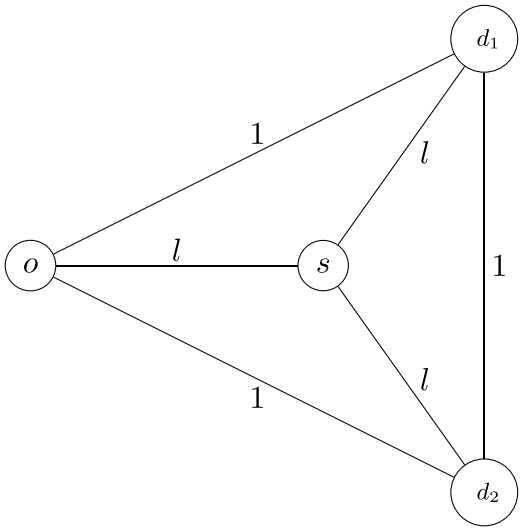
Notes. (a) Feasible Euclidean plane solution. (b) Infeasible network solution.

6.1. Shortest Path Heuristic

The shortest path heuristic (SPH) creates a set of two-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 two-truck pair of trucks $i, j \in K$ that satisfies 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 cannot 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.

Figure 6. Counterexample in Case of Networks



Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

6.2. Euclidean Heuristic

As in the previous section, we construct a general matching problem to create two-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 (16)–(19) 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 points, 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

These approaches generate platooning solutions with two-truck platoons. It may, however, be possible and beneficial to still create larger platoons by merging some of the trucks that drive alone with 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 that 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 mention in Section 1, platoons of three trucks, in practice, already cause a lot more disruption to traffic than two-truck platoons. Keeping this in mind, we restrict ourselves to arrangements in which a maximum of three trucks are allowed in a platoon.

We then follow this up by assessing the performance of the different heuristics. Because 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 integer programming (IP) formulation, we use a discretization parameter of 15 minutes. In practice, trucks typically travel long distances on highways and, therefore, have longer travel times. Furthermore, they are usually planned to depart on the half hour or hour, which this discretization parameter captures. All results presented in this section are averaged over five random truck trip instances and implemented in Python 3.6.3 with Gurobi 7.0 as the IP solver on an Intel i7-6820HQ machine with a 2.70-Ghz processor. All the data and results from this paper are available in the IJOC data repository (Bhoopalam et al. 2023).

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, that is, the node that represents Rotterdam.

For all instances, we generate the earliest start times randomly in the one-hour window between 7:00 and 8:00 am, 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 flexibility f_k to the shortest path time for truck k. This represents the amount of time available for truck k to wait or make a detour to form a platoon. We have $l_k = a_k + t_{o_k d_k} + f_k$ for truck $k \in \mathcal{K}$. We use fuel consumption factors $\rho_l = 0.95$ and $\rho_f = 0.89$, which correspond to 5% savings for the leader and 11% for a follower, respectively (McAuliffe et al. 2018). We consider instances with 20, 30, and 40 trucks and time flexibility f_k of 15, 30, and 45 minutes.

Figure 7. Representation of the Dutch Highway Network



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 IP from Section 4. For (2, 1), we use the procedure from Section 5.1.

In Table 1, we report the optimal percentage 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 trucks making a detour, and the total detour distance expressed as a percentage of the total direct distance for the trucks with a detour. We report these statistics for all combinations of instance sizes and time flexibility. We calculate the percentage fuel savings as follows. Let Z^0 be the total costs if all trucks take their shortest paths and Z^* be the total costs with platooning. The percentage fuel savings are then

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

Compared with UNIFORM, we see that there are higher savings in ONEORIG and ONEDEST. For the setting with 40 trucks and 45 minutes of flexibility ONEORIG results in 8.4% savings as compared with 4.2% 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. That is, (3,∞) generates relatively more benefits than the two-truck setting. 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 generally see higher savings (and more platoon kilometers) 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 am. 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 than at their destinations.

As expected, less constrained $(3, \infty)$ allows for more platoon savings than the more constrained two-truck (2,1) and $(2, \infty)$ cases. However, we see that the differences are quite small, for instance, a 0.4 percentage point (pp) difference between $(3, \infty)$ and $(2, \infty)$ in UNIFORM with 40 trucks and a flexibility of 45 minutes. These small incremental savings from larger platoons are consistent with the results from Albinski et al. (2020) for three- and five-truck platoons. For ONEORIG and ONEDEST, these differences are a bit larger at 1.2 and 1.4 pp, respectively. Again, this is because of it

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Table 1. Platooning Solution Characteristics Averaged over Five Random Instances

Number of trucks	20									30									40								
Flexibility, minutes	15			30			45			15			30			45			15			30			45		
Max trucks per platoon	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2	3	2	2
Platoons per trip	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1	∞	∞	1
UNIFORM																											
Fuel savings, %	1.3	1.3	1.3	1.9	1.9	1.8	2.6	2.4	2.4	2.3	2.2	2.1	2.8	2.8	2.7	3.7	3.4	3.1	2.5	2.3	2.3	3.4	3.1	2.9	4.2	3.8	3.4
Platoon kilometers, %	19.9	19.9	18.9	30.5	30.6	26.3	37.0	35.1	31.5	31.2	30.1	28.6	38.7	36.9	39.0	50.4	48.0	42.9	32.4	30.6	30.8	47.8	45.6	41.2	55.9	53.4	48.2
Trucks in platoon, %	37.0	37.0	38.0	48.0	47.0	44.0	55.0	54.0	54.0	43.3	44.0	41.3	54.7	54.0	52.0	64.0	64.0	58.7	45.5	45.5	45.0	66.5	64.5	60.0	71.0	67.0	65.0
Trucks with detour, %	11.4	11.4	4.0	11.7	11.8	4.0	8.7	6.7	4.0	15.0	14.0	4.7	12.7	11.2	7.3	13.7	14.7	6.0	8.7	7.5	3.0	10.5	10.9	7.5	11.5	13.0	6.5
Detour distance, %	2.7	2.7	2.2	4.1	4.1	4.2	4.2	4.1	1.6	2.3	2.1	1.9	2.5	2.0	2.1	2.4	2.3	2.3	1.5	1.6	3.5	4.5	5.3	2.7	4.1	3.9	2.4
ONEORIG																											
Fuel savings, %	6.8	6.0	6.0	7.5	6.5	6.5	7.8	6.7	6.7	7.1	6.3	6.2	7.4	6.5	6.5	7.9	6.8	6.8	7.6	6.7	6.7	8.1	7.1	7.1	8.4	7.2	7.2
Platoon kilometers, %	79.3	75.2	75.2	87.3	81.5	81.5	90.6	83.3	83.3	83.6	79.1	78.0	87.7	81.7	81.7	92.6	84.8	84.8	89.2	84.1	84.7	94.3	89.2	89.2	96.9	90.3	90.3
Trucks in platoon, %	86.0	84.0	78.0	96.0	91.0	86.0	99.0	93.0	90.0	94.4	93.3	84.0	98.3	93.3	90.7	98.3	93.3	92.0	95.0	95.0	93.0	98.0	96.0	95.0	98.0	98.0	95.0
Trucks with detour, %	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	1.0	0.0	0.0	0.5	0.0	0.0	0.5
Detour distance, %	0.0	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.3	0.0	0.0	0.2	0.0	0.0	0.2
ONEDEST																											
Fuel savings, %	4.7	4.5	4.5	6.4	5.8	5.8	6.8	6.0	5.9	6.5	5.8	5.8	7.4	6.6	6.6	8.0	6.9	6.9	6.9	6.1	6.1	7.6	6.5	6.4	8.2	6.8	6.8
Platoon kilometers, %	60.8	58.6	59.0	78.2	72.1	74.2	81.5	75.5	74.6	78.0	72.4	72.6	88.0	82.9	82.5	93.1	86.5	86.2	82.2	77.3	77.3	88.1	81.2	80.6	93.8	85.6	85.6
Trucks in platoon, %	68.0	64.0	66.0	85.0	81.0	74.0	90.0	87.0	78.0	81.7	76.7	77.3	93.3	86.7	89.3	93.3	91.7	89.3	91.0	86.5	84.0	96.0	90.5	88.0	98.0	95.5	93.0
Trucks with detour, %	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	0.0	0.0	0.5	0.0	0.0	0.5
Detour distance, %	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.2	0.0	0.0	0.2

being easier for trucks to find platoon partners when starting from the same origin or heading to the same destination. Similar results with regards to the benefits of platoon size are seen in Larsen et al. (2019), who consider trucks platooning from a single hub, and Boysen et al. (2018), who look at all trucks traveling the same route (with the same origins and destinations).

Across the different instance types, we hardly see any benefits of allowing more platoons per trip for the two-truck case. We see that $(2, \infty)$ and (2, 1) have similar fuel savings. This suggests that, in these instances, with relatively short trips along the day, the value of allowing multiple platoons per trip is low. Moreover, because we do not consider waiting costs, the results for this setting are optimistic estimates, that is, waiting costs can only reduce the value of allowing multiple platoons per trip.

We now look at the detour characteristics. In UNIFORM, the number of trucks making a detour is generally higher for $(2, \infty)$ than $(3, \infty)$. One potential reason for this is that, in $(3, \infty)$, the trucks are less constrained and need fewer detours to form platoons. In addition, the costs of a detour prohibit trucks from significantly deviating from their shortest paths. Therefore, they tend to find platoon partners along their shortest paths. This also holds true when trucks are spatially clustered as in ONEORIG and ONEDEST, in which there are few trucks with a detour.

Note that, although the additional fuel savings are small, the runtimes are much larger in the cases that allow for multiple platoons per truck. Table 2 shows the average, maximum, and runtimes for the settings with 30 trucks and 30 minutes of flexibility. For UNIFORM, $(3, \infty)$ requires 554 seconds, on average, whereas $(2, \infty)$ and (2, 1) require 426 and 321 seconds, respectively.

Among the different instances, ONEORIG mostly has the largest runtimes. 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 as discussed. However, the runtimes of (2, 1) are unexpectedly high for ONEDEST. This is, again, because of 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 are 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 runtimes.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Table 2. Runtimes in Seconds for Different Instances: 30 Trucks, 30 Minutes Flexibility Averaged over Five Random Instances

Max trucks per platoon	3	2	2
Platoons per trip	∞	∞	1
UNIFORM			
Average	554	426	321
Maximum	659	577	347
Minimum	427	326	224
ONEORIG			
Average	2,513	773	406
Maximum	4,750	886	577
Minimum	1,231	733	306
ONEDEST			
Average	1,194	613	935
Maximum	2,191	798	1,279
Minimum	680	474	650

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 connections, 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 subgraph in which every node is connected to every other node by a path. We include a couple of examples in Online Appendix C.

From the truck connectivity graph, we can compute several statistics. Table 3 shows the average arrangement sizes for all instances (see Online Appendix C for more statistics). The results show larger arrangements for $(3, \infty)$ as compared with $(2, \infty)$ and (2, 1). This suggests that trucks "piggyback" on existing two-truck platoons along their route. 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 opportunities to join and leave platoons. The higher spatial proximity in ONEDEST and ONEORIG allows trucks to better exploit these opportunities to form 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,∞) solution. SPH and EUH represent the basic two-truck platooning heuristics and SPH^+ and EUH^+ the versions in which we apply the improvement procedure. Table 4 reports the optimality gaps and runtimes of these heuristics. To gain some understanding of the value of taking route planning into account, we also compare our methods to a simple fixed-route heuristic (FR). Fixing the routes of the trucks to their shortest paths is common in the platoon planning literature; see, for example, Luo (2022), Larsen et al. (2019), Van de Hoef (2018), and Liang et al. (2016). The FR heuristic that we propose fixes the truck routes to their shortest paths and sorts the truck routes from the longest to the shortest one. Note that the truck time schedules can be adjusted to enable a platoon as long as we satisfy all time window constraints. We then iterate over the sorted list of trucks to create platoons as follows. Starting with truck $k \in \mathcal{K}$ with the longest route, we check for overlaps in routes and time schedules with other truck routes $k+1, k+2, ..|\mathcal{K}|$. When we find a feasible platoon match, we create this platoon. This is repeated until all trucks have been considered.

Table 3. Average Arrangement Sizes: 30 Trucks Averaged over Five Random Instances, $\rho = 0.9$

Max trucks per platoon	3	2	2
Platoons per trip	∞	∞	1
UNIFORM	3.1	2.1	2
ONEORIG	5.2	2	2
ONEDEST	4.5	2	2

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Table 4. Optimality Gaps and Runtimes of Heuristics: 30 Trucks Averaged over Five Random Instances, $\rho = 0.9$

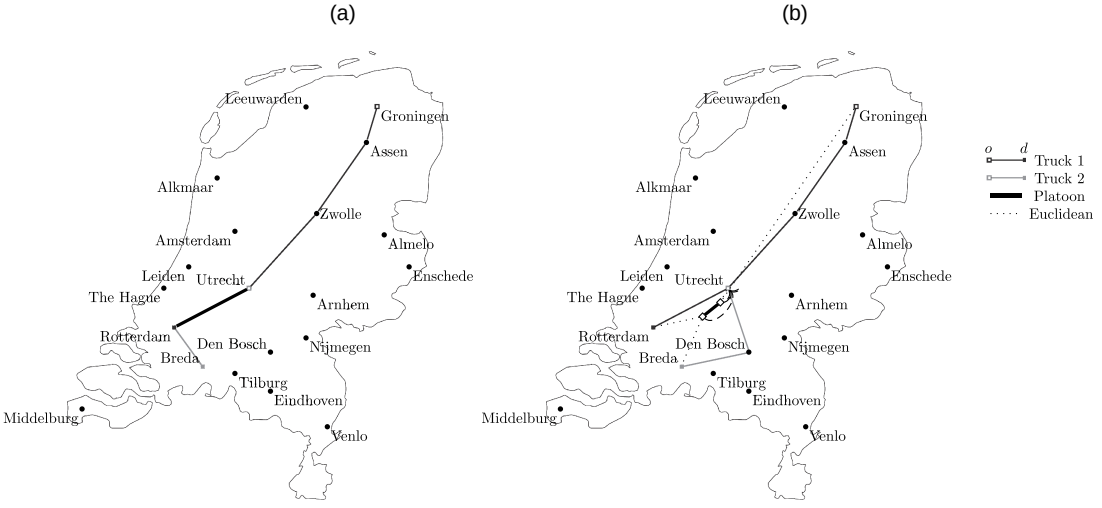
	SPH	SPH$^+$	EUH	EUH$^+$	FR
UNIFORM					
Average, %	0.43	0.30	1.45	1.45	1.34
Maximum, %	1.15	0.95	3.10	3.10	2.59
Minimum, %	0	0	0.12	0.12	0.21
Number optimal	1	1	0	0	0
Runtime, s	1.88	1.91	0	0.01	0.07
ONEORIG					
Average, %	1.07	1.04	2.01	1.91	4.40
Maximum, %	1.62	1.12	2.21	2.12	5.12
Minimum, %	0.78	0.71	1.91	1.61	3.69
Number optimal	0	0	0	0	0
Runtime, s	2.06	2.08	0	0.01	0.04
ONEDEST					
Average, %	1.11	1.01	2.01	1.75	3.84
Maximum, %	1.35	0.96	2.32	2.22	4.42
Minimum, %	0.91	0.71	1.85	1.85	3.59
Number optimal	0	0	0	0	0
Runtime, s	1.75	1.77	0	0.1	0.04

Overall, we see that SPH$^+$ performs best with optimality gaps of at most 1.1%. Even without the improvement step, SPH performs well, that is, with maximum optimality gaps up to 1.6%. 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 runtimes of the heuristics are far lower than the runtimes of our exact methods as shown in Table 2.

EUH and EUH$^+$ show even smaller runtimes 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 Groningen to Rotterdam and the second from Utrecht to Breda. Figure 8(a) shows the optimal network solution in which the trucks platoon between Utrecht and Rotterdam. Figure 8(b) 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.

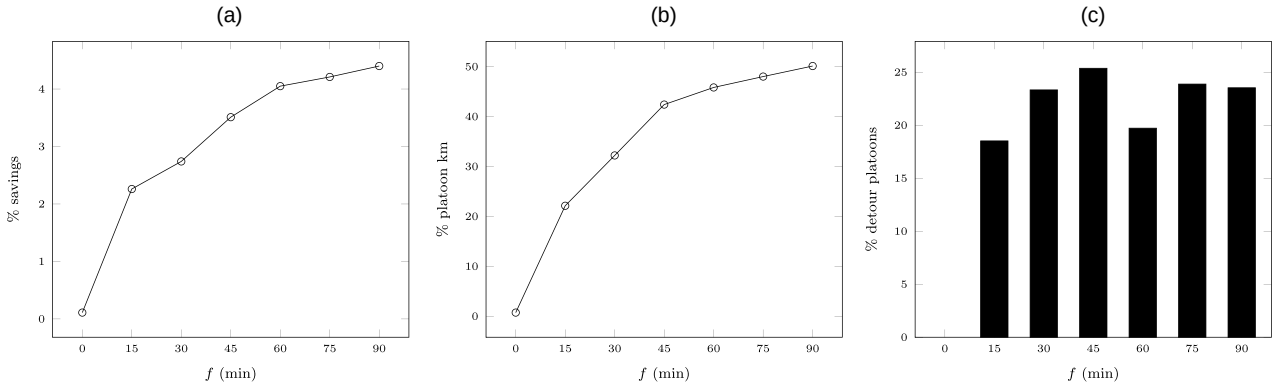
When comparing with FR, we see that SPH and SPH$^+$ outperform FR in all instances, whereas EUH and EUH$^+$ solely outperform FR for the ONEDEST and ONEORIG instances. The fact that FR specifically performs poorly on the ONEDEST and ONEORIG instances may appear counterintuitive as there is less geographic variation in these instances so one expects more route overlap in the shortest routes. However, it appears that, especially in these cases, a little flexibility in the specific route may be very beneficial and, thus, create better performance for our routing-based methods.

Figure 8. Example of Optimal vs. Euclidean Solution



Notes. (a) Optimal two-truck solution. (b) Euclidean two-truck solution and rounded network solution.

Figure 9. Effects of Varying f: 30 Trucks Averaged over Five Random Instances, $\rho = 0.9$



Notes. (a) Variation in percentage savings. (b) Variation in percentage platoon kilometers. (c) Variation in percentage detour platoons.

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.

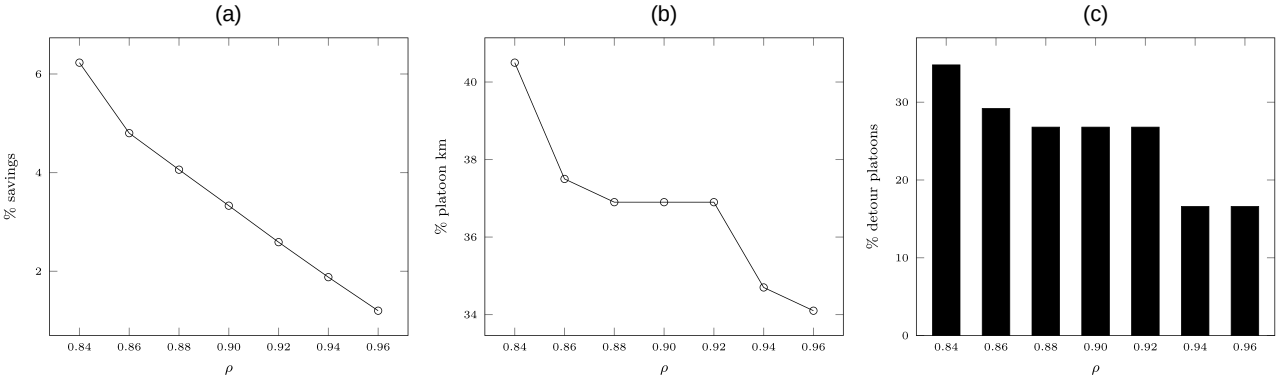
We vary flexibility f between 0 and 90 minutes with increments of 15 minutes for instances with 30 trucks. Figure 9 shows the fuel savings, the platoon kilometers relative to the setting without platoons, and the percentage of platoons with a truck making a detour. From Figure 9(a) and (b), we see that 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 with having no flexibility at all. From a practical perspective, our results suggest that we only need a relatively small amount of time flexibility to achieve benefits from platooning.

Interestingly, Figure 9(c) 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 offset fuel cost savings and, thus, render platoons nonbeneficial. 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, such as 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 savings from Figure 10(a). Figure 10(a), as expected, indicates that the platoon kilometers increase with higher fuel savings, that is, lower ρ. However, the differences are relatively small. We see a similar trend in Figure 10(c) with the percentage of trucks making a detour. Trucks can make more detours when the platoon savings are higher because 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. Figure

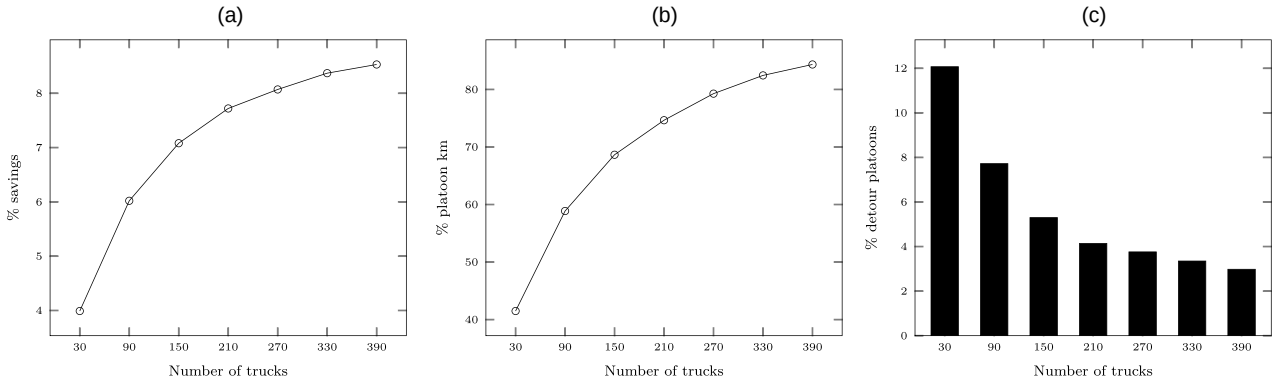
Figure 10. Effects of Varying ρ: 30 Trucks Averaged over Five Random Instances, Flexibility = 30 Minutes



Notes. (a) Variation in percentage savings. (b) Variation in percentage platoon kilometers. (c) Variation in percentage detour platoons.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Figure 11. Effects of Varying the Truck Density, Flexibility = 30 Minutes Averaged over Five Random Instances, $\rho = 0.9$



Notes. (a) Variation in percentage savings. (b) Variation in percentage platoon kilometers. (c) Variation in percentage detour platoons.

11(a) and (b), 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 makes a significant difference.

Not only is it more likely for trucks to form platoons with a higher truck density, but they also need fewer detours to do so. Figure 11(c) 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 look at the planning truck platoons by determining the paths and time schedules of trucks between their origins and destinations. In particular, we focus on a restricted case in which a truck can join at most one platoon, and we only consider platoons of two trucks. We show that this problem can be solved in polynomial time and present several fast exact methods and heuristics. To evaluate the impact of restricting the possible platoons, we also formulate a novel MIP to solve the general PRP-TW.

To assess the impact of the restrictions and evaluate the heuristics, we run a set of numerical experiments on instances based on the Dutch highway network. Our experiments show that the restrictive setting with two-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 produce results within approximately a percent of the optimum.

In this paper, we develop a two-phase approach to solve the general platoon routing problem using a commercial MIP solver. A direction of future research is to develop specific methods to solve these MIPs for larger instances. Another possible area for future research is to consider dynamic trip announcements. In practice, trucks may enter and leave the system or there may be disturbances necessitating an update in the planning. Similarly, traffic conditions could affect platoon plans by affecting travel times and, consequently, even choices for a truck's platoon partners. Therefore, incorporating stochastics in travel times is another promising future research avenue.

Acknowledgments

The authors thank the anonymous reviewers whose comments helped improve the manuscript.

References

Adler A, Miculescu D, Karaman S (2020) Optimal policies for platooning and ride sharing in autonomy-enabled transportation. *Proc. Twelfth Workshop on the Algorithmic Foundations of Robotics* (Springer, Cham, New York), 848–863.

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

Bergenhem C, Schladover S, Coelingh E (2012) Overview of platooning systems. Pfliegl R, Blervaque V, eds. *Proc. 19th ITS World Congress* (ITS International, Vienna).

Bhoopalam AK, Agatz N, Zuidwijk R (2018) Planning of truck platoons: A literature review and directions for future research. *Transportation Res. Part B Methodological* 107:212–228.

Bhoopalam AK, Agatz N, Zuidwijk R (2023) Platoon optimization based on truck pairs. http://dx.doi.org/10.1287/ijoc.2020.0302.cd, https://github.com/INFORMSJoC/2020.0302.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.

Boland N, Hewitt M, Marshall L, Savelsbergh M (2017) The continuous-time service network design problem. *Oper. Res.* 65(5):1303–1321.

Boysen N, Briskorn D, Schwerdfeger S (2018) The identical-path truck platooning problem. *Transportation Res. 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 Res. Part C Emerging Tech.* 105:1–22.

Colling A, Hekkenberg R (2020) Waterborne platooning in the short sea shipping sector. *Transportation Res. Part C Emerging Tech.* 120:102778.

Dantzig GB (1960) On the shortest route through a network. *Management Sci.* 6(2):187–190.

Even S, Itai A, Shamir A (1975) On the complexity of time table and multi-commodity flow problems. *16th Annual Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ), 184–193.

Farber M, Jamison RE (1987) On local convexity in graphs. *Discrete Math.* 66(3):231–247.

Gabow HN (1990) Data structures for weighted matching and nearest common ancestors with linking. *Proc. First Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 434–443.

Goble K (2018) Seven more states adopt rule changes for truck platoons. Online editorial. Accessed January 19, 2020, http://www.landlinemag.com/story.aspx?storyid=72230#.Wyj6vO6FNEY.

Kilcarr S (2016) Driverless trucks: Where they'll work, where they won't. Accessed June 15, 2017, http://m.fleetowner.com/technology/driverless-trucks-where-they-ll-work-where-they-won-t.

Larsen R, Rich J, Rasmussen TK (2019) Hub-based truck platooning: Potentials and profitability. *Transportation Res. Part E Logist. Transportation Rev.* 127:249–264.

Larson J, Munson T, Sokolov V (2016) Coordinated platoon routing in a metropolitan network. Gebremedhin AH, Boman EG, Bora U, eds. *Proc. Seventh SIAM Workshop Combin. Sci. Comput.* (SIAM, Philadelphia), 73–82.

Larson J, Kramer C, Lang KY, Johannson KH (2013) Coordinated route optimization for heavy duty vehicle platoons. *Proc. 16th Internat. IEEE Annual Conf. Intelligent Transportation Systems* (IEEE, Piscataway, NJ), 1196–1202.

Larsson E, Sennton G, Larson J (2015) The vehicle platooning problem: Computation complexity and heuristics. *Transportation Res. Part C Emerging Tech.* 60:258–277.

Liang KY, Mårtensson J, Johansson KH (2013) When is it fuel efficient for a heavy duty vehicle to catch up with a platoon? *IFAC Proc. Volumes* 46(21):738–743.

Liang KY, Mårtensson J, Johansson KH (2014) Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data. *IEEE Intelligent Vehicles Sympos.* (IEEE, Piscataway, NJ), 1061–1068.

Liang KY, Mårtensson J, Johansson KH (2016) Heavy-duty vehicle platoon formation for fuel efficiency. *IEEE Trans. Intelligent Transportation Systems* 17(4):1051–1061.

Lioris J, Pedarsani R, Tascikaraoglu FY, Varaiya P (2017) Platoons of connected vehicles can double throughput in urban roads. *Transportation Res. Part C Emerging Tech.* 77:292–305.

Luo F (2022) Coordinated vehicle platooning with fixed routes: Adaptive time discretization, strengthened formulations and approximation algorithms. Preprint, submitted May 23, https://arxiv.org/abs/2205.11043.

Luo F, Larson J (2021) A repeated route-then-schedule approach to coordinated vehicle platooning: Algorithms, valid inequalities and computation. *Oper. Res.* 70(4):2477–2495.

Luo F, Larson J, Munson T (2018) Coordinated platooning with multiple speeds. *Transportation Res. Part C Emerging Tech.* 90:213–225.

Mahmassani HS (2016) 50th anniversary invited article—Autonomous vehicles and connected vehicle systems: Flow and operations considerations. *Transportation Sci.* 50(4):1140–1162.

McAuliffe B, Lammert M, Lu XY, Shladover S, Surcel MD, Kailas A (2018) Influences on energy savings of heavy trucks using cooperative adaptive cruise control. SAE Technical Paper, Warrendale.

Nourmohammadzadeh A, Hartmann S (2016) The fuel-efficient platooning of heavy duty vehicles by mathematical programming and genetic algorithm. Martín-Vide C, Mizuki T, Vega-Rodríguez M, eds. *Theory and Practice of Natural Computing*, Lecture Notes in Computer Science, vol. 10071 (Springer) 46–57.

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

Richert D, Cortés J (2013) Optimal leader allocation in UAV formation pairs ensuring cooperation. *Automatica J. IFAC* 49(11):3189–3198.

Schladover SE, Nowakowski C, Lu XY, Felis R (2015) Cooperative adaptive cruise control (CACC) definitions and operating concepts. *Transportation Res. Rec.* 2489(1):145–152.

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. Preprint, submitted January 5, https://arxiv.org/abs/1701.01391.

Sun X, Yin Y (2019) Behaviorally stable vehicle platooning for energy savings. *Transportation Res. Part C Emerging Tech.* 99:37–52.

Van Arem B, Van Driel CJ, Visser R (2006) The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans. Intelligent Transportation Systems* 7(4):429–436.

Van de Hoef S (2018) Fuel-efficient centralized coordination of truck platooning. Unpublished PhD thesis, KTH Electrical Engineering, Stockholm, Sweden.

van Maarseveen S (2017) Impacts of trucks platooning at motorway on-ramps. Unpublished master's thesis, TU Delft, Netherlands.

Winder A (2016) Study of intelligent transport systems for reducing CO_2 emissions and increasing safety of heavy goods vehicles, buses and coaches. European Road Transportation Telematics Implementation Coordination Organisation, Brussels.

Xu M, Yan X, Yin Y (2022) Truck routing and platooning optimization considering drivers' mandatory breaks. *Transportation Res. Part C Emerging Tech.* 143:103809.

Xu W, Cui T, Chen M (2023) Optimizing two-truck platooning with deadlines. *IEEE Trans. Intelligent Transportation Systems* 24(1):694–705.

Zhang W, Jenelius E, Ma X (2017) Freight transport platoon coordination and departure time scheduling under travel time uncertainty. *Transportation Res. Part E Logist. Transportation Rev.* 98:1–23.

Downloaded from informs.org by [145.5.176.14] on 02 May 2024, at 02:48 . For personal use only, all rights reserved.