Transportation Research Part E 179 (2023) 103297

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

Maximizing truck platooning participation with preferences

Limon Barua, Bo Zou *, Pooria Choobchian

Department of Civil, Materials, and Environmental Engineering, University of Illinois Chicago, United States

ARTICLE INFO

Keywords:
Maximum stable truck platooning participation
Preference
Platform
Algorithmic approach
Rotation
Integer programming

ABSTRACT

Truck platooning refers to trucks traveling in convoy with longitudinal proximity. By reducing aerodynamic drag, platooning cuts truck energy use and operating cost along with other benefits to the freight transportation system. This research proposes a platform-based platooning system to maximize the participation of truck platooning considering stability of the formed platoons that arises from truck preferences of platooning partners. Preference-based platooning is of both scientific interest and practical importance to the trucking sector, which is highly fragmented especially in the US. The preferences depend on the benefits of fuel saving and schedule adjustment to coordinate the time for platoon formation. The formed platoons are stable in the sense that no two trucks in two platoons want to break away from their current platoons and platoon with each other. Tackling this operation planning problem, the proposed system involves a platform interacting with individual trucks in a way that reduces truck communication and computation burdens, mitigates truck privacy concerns, and overcomes trucks misreporting private information. The central methodological investigation of the interactive process is on how to solve the Maximum Stable Truck Platooning Participation (MS-TPP) problem, for which a two-phase algorithmic approach is proposed. The underlying idea of this approach is to progressively reduce the lengths of truck preference lists, by eliminating truck pairs that do not affect the MS-TPP solution presence and separating trucks in odd rotations – which are key constructs in this approach – from the rest of the truck population. Theoretical properties and computational complexity of this two-phase algorithmic approach are investigated, along with a comparison with an integer programming approach and extensive numerical experiments in the context of a northern Illinois road network in the US. We find that the proposed two-phase algorithmic approach is very efficient compared to the integer programming approach in forming a maximum number of truck platoons while ensuring stability. In addition, the percentage of trucks that end up platooning by solving MS-TPP is much greater than using a greedy approach. The algorithmic approach is computationally scalable for solving large problem instances. The advantages of MS-TPP, in terms of the percentage of trucks in platooning and the average utility gain, become even more prominent as we deal with a larger system with more trucks.

1. Introduction

Truck platooning refers to trucks traveling in convoy with longitudinal proximity, which is made possible thanks to the recent advances in connected and automated driving technologies. Truck platooning reduces aerodynamic drag and consequently truck

* Corresponding author.
E-mail address: bzou@uic.edu (B. Zou).

https://doi.org/10.1016/j.tre.2023.103297
Received 12 April 2023; Received in revised form 10 July 2023; Accepted 23 September 2023

Truck	Ordered preference of partner trucks for platooning
1	3, 2, 4
2	1, 3, 4
3	2, 1, 4
4	1, 2, 3

- ❑ If platoons are (1,2) and (3,4), then trucks 1 and 3 would find it better to platoon with each other;
- ❑ If platoons are (1,3) and (2,4), then trucks 2 and 3 would find it better to platoon with each other;
- ❑ If platoons are (1,4) and (2,3), then trucks 1 and 2 would find it better to platoon with each other.

Fig. 1. Illustration of non-existence of an outcome that all trucks platoon and the platoons are stable.

energy use, operating cost, and emissions (McAuliffe et al., 2018; Sun et al., 2021; Bhoopalam et al., 2018). Field studies suggest that truck fuel saving by traveling in a platoon can reach 10–15 % (Tsugawa et al., 2016; Bishop et al., 2017). Besides fuel saving benefits, trucks traveling in a platoon maintain a smaller separation which means more effective use of the road space and thus road capacity (Bergenhem et al., 2012; Lioris et al., 2017). In the US intercity trucking sector, the fuel saving benefits and road capacity improvement could translate into an annual cost reduction of about $900 million along with a reduced road infrastructure need worth $4.8 billion (Noruzoliaee et al., 2021). Truck platooning is further argued to improve traffic safety with reduced rear-end incidents, by enabling faster reaction and cutting human errors (Peloton, 2020a). To promote widespread platooning and reap the maximum benefits, attracting as many trucks as possible in platooning is a desired objective, for a number of reasons such as keeping trucks and trucking companies involved, spreading confidence and trust in a platooning system, stimulating a large participant pool, and helping gain more practical platooning experiences, all of which are valuable for the success of truck platooning (Bhoopalam et al., 2018). However, not much attention has been paid to the objective of maximizing truck platooning participation.

This study aims to fill this gap. We investigate how to maximize truck platooning participation from an operation planning perspective considering the individual preferences of a population of trucks. Planning truck platooning is shown to yield substantially greater benefits than otherwise opportunistic platooning, which pertains to real-time operations (Liang et al., 2014). Because of this, both the research community and the trucking industry have placed emphasis on the planning aspect of truck platooning (Janssen et al., 2015; Bhoopalam, 2018; Smartt, 2020; Albiński et al., 2020; Xu et al., 2023). As an operation planning problem, platoon formation will be performed before trucks leave their origins, for example, the day before operation or at the beginning of a day. An important feature of the planning process investigated in this paper is that truck preferences for partner trucks are explicitly considered. Preference-based platooning is not only of scientific interest but practically important to the US trucking sector which is highly fragmented, with about two million carriers operating 7.4 million trucks and 57 % of the carriers having just one truck (Federal Motor Carrier Safety Administration, 2022). Even for multi-truck carriers, considering individual truck preferences is still relevant to operation coordination given the heterogeneity in truck routing and schedules and that truck drivers do have preferences (unlike machines). Consequently, a platooning planning scheme without considering individual truck gains/losses may not be well received. For example, a truck would be unwilling to participate in a platoon if it is worse than traveling alone due to significant schedule adjustment to form the platoon.

We consider that maximization of truck platooning participation is performed by a platform, to which each truck submits its routing, preferred departure time, and truck configuration information. The platform computes truck platooning length and schedule adjustment for each possible platoon (termed "platooning opportunity"), and sends the computation results back to the trucks. Then, each truck constructs its preference list of partner trucks, and sends the list to the platform. Lastly, the platform performs maximization of platooning participation taking into account the truck preferences, and sends the platooning results to the involved trucks. The need of a platform can be justified by three reasons. First, absent the platform, each truck would need to have substantial inter-truck communications to exchange route, preferred departure time, and truck configuration information with numerous other trucks. Using the exchanged information, a truck then performs computation to identify its platooning opportunities, and platooning length and departure delay associated with each opportunity. This is in contrast to each truck communicating only with the platform and letting the platform perform most of the computation as in this paper, and may be constrained by the communication and computation capabilities onboard each truck. Second, absent the platform, private information of a truck would have to be widely shared with other trucks, raising significant privacy concerns as opposed to a truck only sharing information with the platform. Third, absent the platform, trucks would perform a "free-market" matching of which the outcome can be uncertain, not only because an overall stable outcome may not exist, but also due to market friction which prevents a stable outcome from being achieved (Echenique and Pereyra, 2016; Sun and Yin, 2019). In fact, having a platform for truck platooning has been argued for by both the research community and the industry (van de Hoef, 2016; European Automobile Manufacturers Association, 2017; Sokolov et al., 2017; Albiński et al., 2020; Sun and Yin, 2019; Trimble Transportation, 2023). As a concrete example, Peloton, an industry leader in truck platooning technologies, offers a cloud-based platform for managing platooning operations (Smartt, 2020; Peloton, 2020b).

With truck preference an essential consideration, the central research problem of this study is: how to form as many platoons as possible that are stable? We term this problem the Maximum Stable Truck Platooning Participation (MS-TPP) problem. An ideal outcome for MS-TPP would be that (1) all trucks platoon and are better off than traveling alone, and (2) there does not exist an unformed platoon, of which all component trucks find it better to form this platoon than in their current situations. These two points ensure that the maximum possible truck participation is achieved (i.e., all trucks participate) and no formed platoon will break up. In particular, the second point pertains to stability among the formed platoons, which is *core stability* in the taxonomy of coalition formation (Hajdukova, 2006). Unfortunately, such an ideal outcome often does not exist. Fig. 1 illustrates an example of four trucks for two-truck platooning (i.e., a platoon consists of two trucks). Each truck has a preference list of partner trucks. For truck 1, its most preferred partner truck is truck 3, followed by truck 2, and then truck 4. We assume that by platooning, a truck is always better than traveling alone. However, as described on the right of Fig. 1, no matter how the four trucks are paired to form two-truck platoons, there always exist two trucks from two different platoons which want to break away from their partners to form a new platoon, because by doing so the two trucks find more preferred partners for platooning.

Coupling platooning stability with maximum participation, we consider MS-TPP for which: (1) the number of platooning trucks is maximized; (2) the platooning trucks are all better off than traveling alone; and (3) stability is maintained among the platooning trucks. The consideration of stability among the platooning trucks is sensible, because even if the platooning trucks exchange information with each other, no existing platoon will break up and no new platoon will form among these trucks. On the other hand, non-platooning trucks will end up traveling alone with no information about other trucks as a truck only interacts with the platform. Without other trucks' information, the non-platooning trucks are unlikely to further explore platooning opportunities and disrupt the existing platoons.

In this paper we focus on two-truck platooning, which has been considered as the likely form of platooning at least in the near future (Janssen et al., 2015; Bishop et al., 2017; Bhoopalam 2018; Xu et al., 2023), for three reasons. First, many existing technologies (e.g., those developed by Peloton, Iveco, and MAN) are tested for or focus on two-truck platooning (Technavio, 2018; Hochschule Fresenius et al., 2019; Smartt, 2020; Peloton, 2021). Second, serious concerns about traffic interruptions and consequently safety can arise when a platoon has more than two trucks. Wang et al. (2019) show through simulation that three-truck platooning would cause significantly more merging problems than two-truck platooning. The merging problems are pertinent to both car and single truck drivers (Janssen et al., 2015; Crane et al., 2018). In contrast, two-truck platoons are unlikely to pose significant issues in free-flowing traffic (Bishop et al., 2017). Hassan et al. (2020) demonstrate that in peak traffic situations, platooning deteriorates traffic operations and safety to a much greater extent if a platoon has more than two trucks. Third, with heavier loading, longer platoons exacerbate pavement and bridge damage while also posing greater structural safety hazards (Kamranian, 2019; Al-Qadi et al., 2021; Ling et al., 2022). Because of this, the investigation of bridge suitability for truck platooning in Florida assumes two-truck platooning (Crane et al., 2018). In fact, several states in the US including Florida, Kentucky, Oklahoma, and Maryland have stipulated a platoon size of two as they begin implementation of truck platooning (Crane et al., 2018; Simon Law, 2019; Maryland Department of Transportation, 2021). Similar regulations have also been seen outside the US (Peloton, 2019). While two-truck is our focus, at the end of the paper we also discuss how the proposed approaches to MS-TPP may be extended to situations that allow for more than two trucks in a platoon.

In tackling the MS-TPP problem, this paper attempts to make three contributions. The first contribution is proposing a platform-based platooning system to maximize truck platooning participation while ensuring stability of formed platoons given individual truck preferences for platooning partners. To the best of our knowledge, this participation maximization problem has not been studied in the literature. However, as described earlier, the problem is practically relevant to promoting widespread truck platooning and ensuring that trucks are willing to participate. In doing so, the construction of truck preference lists accounts for not only fuel saving benefits from platooning, but also schedule adjustment to coordinate the time to form platoons. The proposed system involves a platform interacting with individual trucks in a way that reduces truck communication and computation burdens, mitigates truck privacy concerns of sharing sensitive proprietary information with other trucks, and overcomes the issue of trucks misreporting private information.

The second contribution is employing a two-phase algorithmic approach to finding a solution for the MS-TPP problem. The basic algorithmic principle of this approach is to progressively reduce the lengths of truck preference lists, by 1) eliminating truck pairs from the preference lists that do not affect MS-TPP solution presence and 2) separating truck groups that are independent from the rest of the truck population, so that in the end truck platoons can be easily formed while preserving the maximum participation and stability requirements. More specifically, the first phase pertains to trucks iteratively proposing to one's most preferred partner truck and the proposal recipient deleting trucks less preferred than the proposal. This results in reduced truck preference lists. Building on the reduced lists, the second phase iteratively identifies and separates/eliminates rotations, which are key constructs in the paper characterizing cyclic sequence of trucks in terms of preferences, from the truck preference lists. The iterations proceed until no more rotations can be identified. By exploiting properties of rotations, the second phase decomposes the truck population into two groups: (1) trucks that are only left with one platooning partner and (2) trucks that are in odd rotations (which are a specific type of rotations defined in subsection 4.2). Formation of platoons is then performed independently for group (1) and for trucks in each odd rotation in group (2). In the end, the resulting platoons give an MS-TPP solution.

From a theoretical perspective, the idea of the two-phase approach can be connected to stable roommate matching initially proposed by Irving (1985). However, Irving's algorithm can only find a stable matching if it exists; otherwise, the algorithm just reports non-existence of a stable matching outcome. A more closely related work to our study is Tan (1990) which looks into maximum stable matching. Still, our paper differs from Tan's study, in demonstrating key properties of odd and non-odd rotations in the second phase (see Definitions 3 and 5 for definitions of rotations and odd rotations). Specifically, we develop an intuitive inductive proof to show that each truck in an odd rotation has exactly two trucks on its preference list (Proposition 3), based on which we further argue that if an odd rotation is identified, an MS-TPP solution can be obtained by independently seeking an MS-TPP solution in the odd rotation and an MS-TPP solution from the remaining trucks (Proposition 4). Such a reasoning process is not in Tan (1990), at least not explicitly. Moreover, to show that any MS-TPP solution after eliminating a non-odd rotation is also an MS-TPP solution before eliminating the rotation, a three-step proving process is devised to demonstrate both "maximum" and "stability" in MS-TPP (see Lemmas 3–4 and Proposition 5). As opposed to this, Tan's approach is rather lengthy and involves making a new claim and considering whether there are common agents (in our context, trucks) in the two sets of agents in a non-odd rotation, both of which are unnecessary in our proof.

Table 1
Synthesis of the existing studies that focus on system optimization and platooning stability, and comparison with our work.

Focus	Study	Perspective	Truck OD consideration	Truck schedule adjustment	Approach
System optimization focused	Larsson et al. (2015)	Centralized	Multiple ODs	Allowed	Integer linear programs solved by heuristics
	Larson et al. (2016)	Centralized	Multiple ODs	Allowed	Mixed-integer linear program solved by commercial solver
	Nourmohammadzadehand Hartmann (2016)	Centralized	Multiple ODs	Allowed	Mixed-integer nonlinear program solved by genetic algorithm
	Luo et al. (2018)	Centralized	Multiple ODs	Allowed	Mixed-integer linear program solved by a decomposition-based heuristic
	Boysen et al. (2018)	Centralized	One road segment	Allowed but bounded by time window	Mixed-integer linear program solved by customized polynomial time algorithms
	Larsen et al. (2019)	Centralized	Multiple ODs	Allowed	Mixed-integer linear programs solved by heuristics
	Abdolmaleki et al. (2021)	Centralized	Multiple ODs	Allowed but bounded by time window	Mixed-integer nonlinear program solved by outer approximation and heuristics
	Chen et al. (2021)	Centralized	One OD	Allowed but bounded by time window	Mixed-integer second-order-cone program solved by a column generation-based heuristic
	Johansson et al. (2021)	Centralized	Multiple ODs	Allowed but bound by a budget	Integer linear programs
	Scholl et al. (2023)	Centralized	Multiple ODs	Allowed but bounded by time window	Mixed-integer program solved by metaheuristic
Stability focused	Sun and Yin (2019)	Centralized	One road segment	Not considered	Integer linear program solved by column generation and benefit redistribution based on a fair allocation mechanism
	Sun and Yin (2021a)	Peer-to-peer	One road segment (applicable to multiple ODs)	Not considered	Decentralized game-theoretical modeling with benefit redistribution
	Sun and Yin (2021b)	Peer-to-peer	One platoon	Not considered	Auction mechanism to determine leader–follower position and associated benefits in a platoon
	Bouchery et al. (2022)	Centralized	One OD (applicable to multiple ODs)	Allowed	Integer linear program solved by solver and heuristic, and cost allocation based on the core and Shapley value
	Our study	Centralized	Multiple ODs	Allowed	Two-phase algorithmic approach and integer linear program for exact solution that maximizes platooning participation with stability among participating trucks

A more detailed discussion of the difference between our proving process and Tan's approach is given in subsection 5.2 (after Proposition 5).

For the third contribution of the paper, it is about computational investigations of the two-phase algorithmic approach, which are entailed in three aspects. First, a theoretical examination of the computational complexity for the approach is conducted. Second, to verify the correctness of this approach and for the purpose of benchmarking solution time, an alternative integer programming approach to solve MS-TPP is presented. Third, extensive experiments are performed to numerically investigate the computational properties of the algorithmic approach. The results show significant advantage of the algorithmic approach over the integer programming approach, with much smaller and scalable computation time needed to find an MS-TPP solution. The numerical experiments further show the benefits of implementing MS-TPP compared to greedily forming platoons. These results suggest a promise of MS-TPP and the proposed system for real-world truck platooning implementation.

For the remainder of the paper, section 2 reviews the relevant literature, based on which we identify issues in the existing research. Section 3 gives a description of the problem setup and an overall picture of the proposed platooning system. In section 4, we discuss how truck preference lists are constructed. Section 5 presents the two-phase algorithmic approach, ensued by an integer programming formulation presented in section 6. Numerical experiments are reported in section 7. Section 8 summarizes the paper, discusses its limitations, and explores possibilities for future research.

2. Literature review

While truck platooning research has been conducted in various aspects such as fuel saving quantification, platooning system planning, platooning impacts on surrounding traffic, platooning vehicle control, and human factors, our literature review focuses on the planning aspect of truck platooning. In line with the observation of Bouchery et al. (2022), we find that the majority of the truck platooning planning studies seek to find optimal truck routing and schedules in a centralized setting, with the objective of minimizing total fuel consumption given a number of trucks (Larsson et al., 2015; Larson et al., 2016; Nourmohammadzadeh and Hartmann 2016; Luo et al., 2018; Boysen et al., 2018; Larsen et al. 2019; Abdolmaleki et al., 2021; Chen et al., 2021; Scholl et al. 2023). From the system optimization perspective, Larsson et al. (2015) show that the truck platooning problem is an NP-complete combinatorial optimization problem, suggesting that finding an exact optimal solution can be difficult even for small problems. As a result, approximation techniques have been considered such as genetic algorithm (Nourmohammadzadeh and Hartmann 2016), constructive and improvement heuristics (Larsson et al., 2015), decomposition (Luo et al., 2018), greedy insertion and destroy-repair heuristics (Larsen et al. 2019), adaptive large metaheuristic search (Scholl et al. 2023), outer approximation and dynamic programing-based heuristic (Abdolmaleki et al., 2021), and column generation (Chen et al., 2021). A more comprehensive review of the optimization-oriented truck platooning planning studies up to 2018 can be found in Bhoopalam et al. (2018).

However, finding a solution that is exactly or near system optimum can be practically questionable because, as mentioned in section 1, the truck industry is very fragmented, especially in the US, which means that trucks belonging to different operators have their own interests and may not follow the platoon formation outcome suggested by system optimization. Even within one truck operator, individual truck preferences are still relevant to platooning operation coordination given the heterogeneity in truck routing and schedule and that truck drivers do have preferences. Thus, understanding stability of the formed platoons by accounting for individual truck preferences is important. A few attempts have been made to look into the stability issue in truck platooning. In Sun and Yin (2019), a benefit redistribution mechanism is designed to incentivize trucks to form and maintain desired platoons while seeking the system optimum platoon formation. In Bouchery et al. (2022), heterogeneous truck departure times and origin–destination (OD) pairs are further considered. In addition to formulating and solving a system optimization model, the authors investigate cost sharing based on a consecutive platooning game and the Shapley allocation.

Different from Sun and Yin (2019) and Bouchery et al. (2022) for which platooning benefit is redistributed as a way to stabilize the formed platoons in a centralized setting, the issue of platooning stability issue in a peer-to-peer communication environment has also been explored (Sun and Yin, 2021a, b). In Sun and Yin (2021a), both one-to-one and many-to-many coordination among trucks to form stable platoons are investigated. For one-to-one coordination, each truck connects and communicates with another nearby truck. A game based on the classic bilateral trade model is proposed with expressed willingness-to-pay and inter-truck payment to determine the truck positions in the platoon. The many-to-many coordination is modeled as a one-sided matching problem with the matching stability achieved by a benefit redistribution mechanism. Focusing on the formation of one platoon, Sun and Yin (2021b) propose an auction mechanism which determines the leader–follower positioning and associated benefits in each position in a platoon, to facilitate the formation and maintain the behavioral stability of platoons in a distributed way. Linear monetary transfer functions are employed which result in an approximate equilibrium to assure behavioral stability. A synthesis of the existing studies that focus on system optimization and platooning stability is presented in Table 1.

A few issues are identified from the above review. First, while a large number of works are dedicated to system optimization, no study has looked into maximizing truck participation in platooning. Also, limited efforts are made to research stability in truck platooning, which is practically important given the fragmented nature of the trucking industry. Second, among the research focusing on stability, private information is assumed fully shared either with the platooning controller if in a centralized setting or with other trucks if by peer-to-peer communications. Although some private information (e.g., truck configuration and routing) can be verifiable, other information, such as truck fuel cost and value of time, is proprietary and sensitive. This gives rise to the question of whether trucks would be willing to share the information and the possibility that trucks strategically misreport private information to gain benefits, as recognized in other transportation contexts (e.g., Zou et al., 2015; Bian and Liu, 2019; Liang et al., 2020). Indeed, Sun and Yin (2021a) point out that one-sided matching as used in their study does not guarantee incentive compatibility in truck platooning.

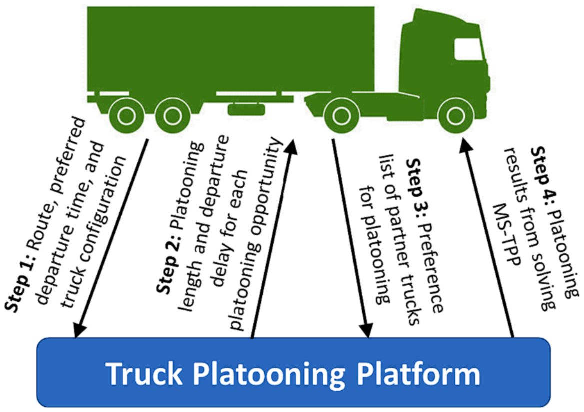


Fig. 2. The interactive process between a truck and the platform while computing an MS-TPP solution.

Third, the consideration of waiting time cost, which is rare in the stability-focused studies, could compromise the monotonicity of a truck's utility with respect to the total utility of the platoon to which the truck belongs (e.g., a platoon may have a large fuel saving due to long platooning length but little benefit to one of its trucks because of large schedule adjustment) and consequently platooning stability under benefit transfer (Sun and Yin, 2021a). Fourth, in peer-to-peer truck platooning systems, a truck may exchange information with many other trucks to identify the most suitable platooning opportunity. Significant communication and computation requirements on individual trucks will arise, which however may be limited by the communication and computation capabilities onboard trucks. The platform-based platooning system in this paper is intended to address/mitigate these issues.

3. Problem setup and overall picture of the platooning system

3.1. Problem setup

In this paper, we are interested in planning truck platooning for a time horizon, e.g., the next few hours or the next day. Consider n trucks each characterized by its route and preferred departure time from the origin. Given the route and preferred departure time, each truck also has estimates of travel speed on the road links its route traverses. For operation planning problems, assuming known routes, speed, and expected departure time/travel time window is quite common in the truck platooning literature (e.g., Larson et al., 2016; Boysen et al., 2018; Sun et al., 2021; Abdolmaleki et al., 2021; Johansson et al., 2021; Bouchery et al., 2022, to name a few). Two trucks can form and travel in a platoon along their overlapped length (i.e., platooning length). By traveling in a platoon, a truck saves fuel and consequently reduces its operating cost. Although fuel savings for the leading and the trailing trucks in a platoon are different, we consider that the fuel savings of a platoon, which have been well quantified in the literature (Browand et al., 2004; Bonnet and Fritz, 2000), will be evenly shared between the two trucks in the platoon, as advocated in practice (North American Council for Freight Efficiency, 2016). To do so, money/credit transfer between the two trucks may incur and be performed via the platform. An even sharing of fuel saving benefit creates an environment of equal treatment among trucks while keeping the platform revenue neutral. Note that an underlying assumption is that truck drivers will be fully engaged during platooning. If needed, the potential benefit of the driver resting in the trailing truck can also be included as part of the platooning benefit.

If traveling individually, two trucks that can travel in a platoon are likely to arrive at the start of the platooning length at different times. To form a platoon, at least one truck needs to deviate from its preferred schedule to coordinate the time of arrival at the start of the platooning length. We consider that between the two trucks, the one arriving earlier at the beginning of the platooning length if traveling individually will delay its departure from the origin. Although en-route waiting and speed adjustment might be alternative ways to adjust the arrival time at the start of the platooning length, these options are less practical as it is not always possible to find a safe place to wait while en route or change speed in the middle of traffic.

3.2. Overall picture of the platooning system

As mentioned in section 1, we consider a platooning system where a platform is present and seeks to maximize the number of platoons that can be formed and are stable among themselves. To do so, the platform needs to know all platooning opportunities and each truck's preference of partner trucks associated with the platooning opportunities. Uncovering the preferences requires computing truck utility change from the platooning opportunities, which involves sensitive proprietary information of each truck such as its unit fuel cost and the value of time. To mitigate the privacy concerns, an interactive process between each truck and the platform is conceived for information exchange. In this process, a truck only shares its private information that is essential for the platform to compute platooning opportunities (step 1 in Fig. 2), including route (to compute the platooning length with another truck), preferred departure time (to compute schedule deviation), and truck configuration (to compute fuel saving from platooning). Truck route and configuration information can be verified by the platooning partner if a truck ends up platooning. For preferred departure time, if a truck misreports its information, the truck would risk not meeting the partner truck at the coordinated time and thus disclosing misreporting. Considering these and that other private information on unit fuel cost and value of time is not shared, a truck misreporting its private information would be unlikely.

Using the information shared from trucks, the platform computes the platooning opportunities for each truck, and sends back to each truck the platooning length and departure delay for each platooning opportunity (step 2 in Fig. 2). After receiving the platooning opportunities, a truck computes its utility change for each opportunity compared to traveling alone, based on which an ordered list of the preferred partner trucks for platooning is constructed. The truck then sends its preference list to the platform (step 3 in Fig. 2). After receiving truck preference lists, the platform computes an MS-TPP solution and informs each truck of: 1) whether to platoon; and if yes, 2) which truck to platoon with; 3) where to start and end the platoon; and 4) departure delay if any (step 4 in Fig. 2). At each time when solving the MS-TPP, we consider that a truck platoons at most once. Nonetheless, the problem could be repeatedly solved in a rolling fashion to allow a truck to platoon more than once in the trip.

4. Constructing truck preference lists

As made clear in section 3, the key inputs for the platform to compute an MS-TPP solution are truck preference lists. Each truck constructs its preference list based on the utility changes of the truck under different platooning opportunities compared to traveling alone. Given the platooning length, fuel saving per unit distance in platooning (which depends on truck configuration), and departure delay, the utility change of truck u by platooning with another truck v, $\Delta U_{u(v)}$, is calculated as:

$$\Delta U_{u(v)} = \beta_u^1 \alpha_{u,v} l_o(u,v) - \beta_u^2 \Delta t_{u(v)} \tag{1}$$

where $l_o(u,v)$ is the platooning length when truck u platoons with truck v. $\Delta t_{u(v)}$ is the departure delay of truck u when platooning with truck v. $l_o(u,v)$ and $\Delta t_{u(v)}$ are computed by the platform (step 2 in Fig. 2). β_u^1 is the unit fuel cost in \$/gallon of truck u. $\alpha_{u,v}$ is half (as it is even split) of unit fuel saving in gallons/mile of the (u,v) platoon. While total fuel saving of the (u,v) platoon may differ depending on whether u or v is the leading truck, it is natural to consider that u and v will choose their positions in the platoon (only two choices) that yield the larger total fuel saving. β_u^2 is the value of time in \$/minute of truck u. Note that although the platform will know $\alpha_{u,v}$ based on the truck configuration information received, β_u^1 and β_u^2 are sensitive proprietary information and only known to truck u itself. That is why $\Delta U_{u(v)}$ can only be computed by each truck individually.

The platform can compute $l_o(u,v)$ and $\Delta t_{u(v)}$. To do so, each truck route is characterized as a sequence of traversed links. The platooning length is considered as the longest uninterrupted overlapping length, which is obtained by computing the longest weighted common substring of the two link sequences corresponding to u and v's routes weighted by link lengths. A recursive procedure adapted from the problem of computing the longest unweighted common substring (Knuth et al., 1977) is used to identify the longest uninterrupted overlapping length. Specifically, suppose that truck u's route is expressed as $\{a_1, a_2, \cdots, a_p\}$ and truck v's route expressed as $\{b_1, b_2, \cdots, b_q\}$, where $a_1, a_2, \cdots, a_p$ and $b_1, b_2, \cdots, b_q$ are the consecutive links forming u and v's routes. The length of a link is denoted by $d(\cdot)$. We use Eq. (2)-(3) to recursively construct two matrices OL and D of dimension $p \times q$. In the first matrix, each element $OL(i,j)$ is a vector storing the consecutive overlapping links of routes a and b that goes to the end of a_i and b_j. If a_i and b_j overlap ($a_i = b_j$), then a_i (b_j) will be added to the consecutive overlapping links that goes to the end of link a_{i-1} (b_{j-1}), which is expressed as $OL(i,j) = [OL(i-1,j-1)\ a_i]$. Note that $OL(i-1,j-1)$ can be empty if a_{i-1} and b_{j-1} do not overlap. If a_i and b_j do not overlap ($a_i \neq b_j$), then $OL(i,j) = []$ meaning there are no consecutive overlapping links that go to the end of a_i or b_j (Eq. (2)). In the second matrix, $D(i,j)$ denotes the length of $OL(i,j)$. $d(a_i)$ denotes the length of link a_i. Following the similar reasoning for computing $OL(i,j)$, elements in the second matrix D can be recursively computed, as shown in Eq. (3).

$$OL(i,j) = \begin{cases} [OL(i-1,j-1)\ a_i] & \text{if } a_i = b_j \\ [\ ] & \text{otherwise} \end{cases} \tag{2}$$

$$D(i,j) = \begin{cases} D(i-1,j-1) + d(a_i) & \text{if } a_i = b_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

To implement Eq. (2)-(3), $OL(i,j) = [\ ]$ is set to be empty if $i = 0$ and/or $j = 0$. Similarly, $D(i,j) = 0$ if $i = 0$ and/or $j = 0$. As computing OL and D involves all combinations of $i = 1, \cdots, p$ and $j = 1, \cdots, q$, the time complexity for computing OL and D is $O(pq)$.

Once OL and D are computed for trucks u and v, we pick $D(i,j)$ with the largest value as $l_o(u,v)$. It may be possible that two truck routes have two separate overlapping lengths. To see this, consider $l_o^1(u,v)$ is the first overlapping length of trucks u and v. $l_o^2(u,v)$ is the second overlapping length of u and v. From the end of $l_o^1(u,v)$ to the beginning of $l_o^2(u,v)$, truck u takes path 1 while truck v takes a different path 2. As the context of truck platooning is intercity transportation, two paths connecting the same starting point and the same ending point can differ significantly in distance. Thus, it is unlikely for either of the two trucks to incur substantial time deviation to be able to platoon on both overlapping lengths. Rather, it is plausible that different trucks have the same view about the most efficient path between the end of $l_o^1(u,v)$ and the beginning of $l_o^2(u,v)$, and consequently choose that same path. For example, if two truck routes follow their shortest-distance routes, then there will be only one overlapping length of the two routes.

Once $l_o(u,v)$ is identified, the planned arrival times of u and v at the start of the platooning length without platooning will be computed using each truck's preferred departure time, lengths of the links from the truck origin to the start of the platooning length, and expected average speeds on those links. Under the assumption that a truck will delay its departure to coordinate platoon formation,

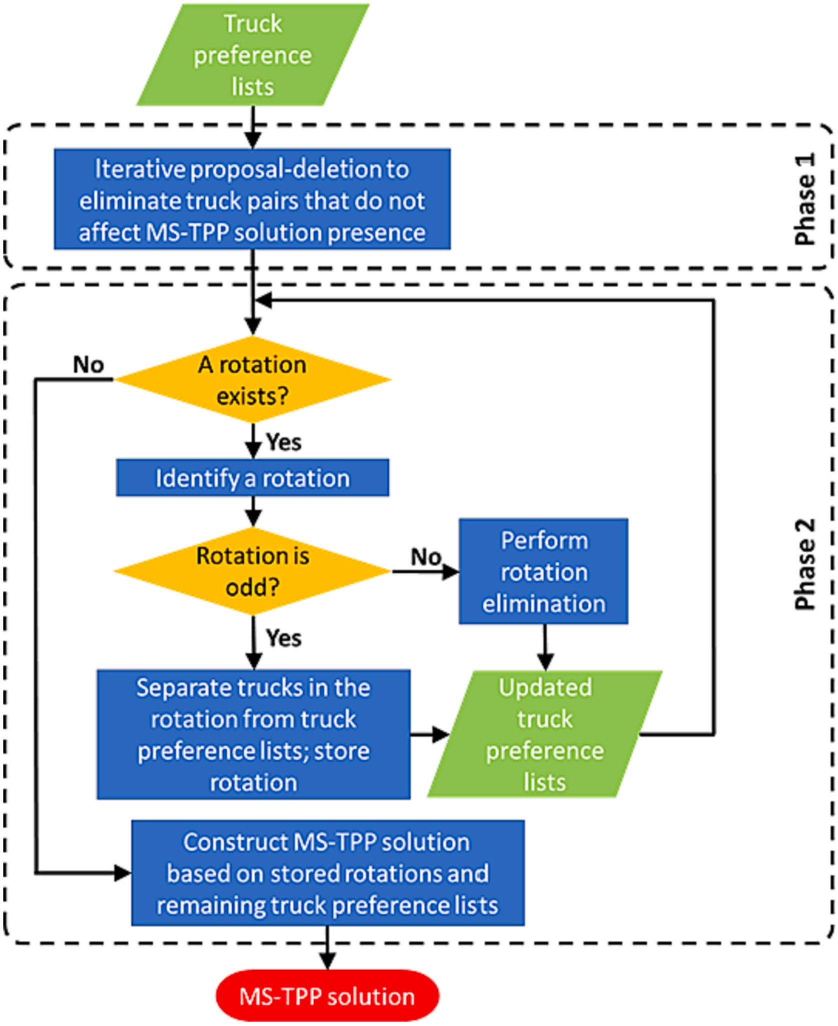


Fig. 3. Flowchart for the two-phase algorithmic approach to construct an MS-TPP solution.

$\Delta t_{u(v)}$ will be equal to the difference in the planned arrival time of the two trucks if u arrives earlier. If u arrives later, $\Delta t_{u(v)} = 0$, i.e., no departure delay for u.

With $l_o(u, v)$ and $\Delta t_{u(v)}$ computed, the information will be transferred from the platform to truck u, which calculates $\Delta U_{u(v)}$ and sorts $\Delta U_{u(v)}$'s in decreasing order. Because platooning should not make a truck worse than traveling alone, only v's with a positive $\Delta U_{u(v)}$ are kept and form the preference list of u. The preference list is then sent back to the platform. After collecting all trucks' preference lists, the platform cleans the preference lists so that preferences are symmetric: u is on v's list if and only if v is on u's list. In other words, a platooning opportunity should be utility improving compared to no platooning for both trucks in the platoon.

5. A two-phase algorithmic approach to MS-TPP solution

In this section, we investigate how the platform finds an MS-TPP solution given the truck preference lists by a two-phase algorithmic approach. The first phase is an iterative proposal-deletion process to eliminate truck pairs from the preference lists that do not affect MS-TPP solution presence. In the second phase, the truck preference lists are further reduced in length, by iteratively identifying and separating/eliminating rotations from the preference lists. The iterations proceed until no more rotations can be identified. By exploiting a series of properties of rotations, the second phase decomposes the truck population into two groups: (1) trucks that are only left with one platooning partner and (2) trucks that are in odd rotations. From this outcome, an MS-TPP solution can be easily constructed by forming platoons independently for trucks in group (1) and for trucks in each odd rotation in group (2). Fig. 3 illustrates the flow of the algorithmic approach. The rest of this section presents the methodological details. For brevity, in some places we use "list" as a short name for "preference list".

Table 2
An illustrative example of performing iterative proposal-deletion.

Step	Proposal	Updated preference lists					Status
0		Truck 1	2	3	4		
		Truck 2	3	4	1	5	
		Truck 3	1	4	2	5	
		Truck 4	1	3	2	5	
		Truck 5	2	3	4		
1	1→2	Truck 1	2	3	4		*1 is held by 2*
		Truck 2	3	4	1		
		Truck 3	1	4	2	5	
		Truck 4	1	3	2	5	
		Truck 5	3	4			
2	2→3	Truck 1	2	3	4		1 is held by 2
		Truck 2	3	4	1		*2 is held by 3*
		Truck 3	1	4	2		
		Truck 4	1	3	2	5	
		Truck 5	4				
3	3→1	Truck 1	2	3			1 is held by 2
		Truck 2	3	4	1		2 is held by 3
		Truck 3	1	4	2		*3 is held by 1*
		Truck 4	3	2	5		
		Truck 5	4				
4	4→3	Truck 1	2	3			1 is held by 2
		Truck 2	4	1			3 is held by 1
		Truck 3	1	4			*4 is held by 3*
		Truck 4	3	2	5		
		Truck 5	4				
5	2→4	Truck 1	2	3			1 is held by 2
		Truck 2	4	1			3 is held by 1
		Truck 3	1	4			4 is held by 3
		Truck 4	3	2			*2 is held by 4*
		Truck 5					*5's preference list is empty*

5.1. *Phase 1: Iterative proposal-deletion*

The first phase, which is characterized by a proposal-deletion process, is conceptually simple: we iteratively let each truck u propose to the first truck on its current preference list. Consider the first truck on u's list is v. If v accepts and subsequently holds the proposal, v will delete all trucks from its list that are less preferred than u, i.e., deleting all w's with $w \prec_v u$. By symmetry, truck v is also deleted from the lists of the w's.

At the beginning of this process, no truck holds a proposal. So when a truck receives a proposal, it will accept and hold the proposal. As the process continues, it can occur that when a truck u proposes to truck v, v holds another truck u' which proposed to v earlier. In this case, it must be that $u' \prec_v u$. Then v will reject u', and accept and hold u. As u' is part of the w's that $w \prec_v u$ on v's list, u' will be deleted from v's list. So will v be deleted from the list of u'. At this point, u' becomes unheld and will propose to the first truck on its current list, which no longer has v. Note that it is not possible to have $u \prec_v u'$, because if this were true u would have been deleted earlier from v's list when v accepted the proposal from u'. This iterative proposal-deletion process continues until every truck either is held by another truck or has an empty preference list. Algorithm 1 describes the process.

Algorithm 1: Proposal-deletion process

input: truck preference lists
while there exists a truck u that is not held by another truck and has a non-empty preference list
    u proposes to the first truck v on its current preference list
    if v holds another truck u', **do**
        v rejects u', and accepts and holds u
    else
        v accepts and holds u
    end if
    Delete all w's that $w \prec_v u$ from the preference list of v
    Delete v from the preference lists of the w's
end while

Table 2 gives an example to illustrate the proposal-deletion process. In Table 2, the first column indicates the steps. The second column shows the proposal made at each step. The third column shows the updated preference lists after proposal-deletion is performed at the step. Let us use $\mathbb{L}_0$ to denote the initial preference lists, as shown in the third column in step 0. In $\mathbb{L}_0$, the preference of truck 1 is: $2\succ_1 3\succ_1 4$. Truck 5 is assumed not able to platoon with truck 1, so not on truck 1′s list. The last column documents the status at the end of each step, in terms of which truck has been held by which truck, and what preference list(s) are empty. The status update in each step is italicized.

- In step 1, truck 1 proposes to the first truck on its current list, which is truck 2. Truck 2 accepts and holds truck 1. At the same time, truck 2 deletes the truck less preferred than truck 1 from its list, which is truck 5. By symmetry, truck 2 is also deleted from truck 5′s list. Status update: truck 1 is held by truck 2.
- In step 2, truck 2 proposes to the first truck on its current list, which is truck 3. Truck 3 accepts and holds truck 2. At the same time, truck 3 deletes the truck less preferred than truck 2 from its list, which is truck 5. By symmetry, truck 3 is also deleted from truck 5′s list. Status update: truck 2 is held by truck 3.
- In step 3, truck 3 proposes to the first truck on its current list, which is truck 1. Truck 1 accepts and holds truck 3. At the same time, truck 1 deletes the truck less preferred than truck 3 from its list, which is truck 4. By symmetry, truck 1 is also deleted from truck 4′s list. Status update: truck 3 is held by truck 1.
- In step 4, truck 4 proposes to the first truck on its current list, which is truck 3. As truck 3 holds truck 2, truck 3 updates the truck it accepts and holds as truck 4. At the same time, truck 3 deletes the truck less preferred than truck 4 from its list, which is truck 2. By symmetry, truck 3 is also deleted from truck 2′s list. Status update: truck 4 is held by truck 3; truck 2 is no longer held by any truck.
- In step 5, truck 2 proposes to the first ruck on its current list, which is truck 4. Truck 4 accepts and holds truck 2. At the same time, truck 4 deletes the truck less preferred than truck 2, which is truck 5. By symmetry, truck 4 is also deleted from truck 5′s list. Status update: truck 2 is held by truck 4; in addition, the preference list of truck 5 becomes empty.

Now that trucks 1–4 are each held by a truck and truck 5 has an empty list, there does not exist a truck that is not held by another truck and has a non-empty preference list. Per Algorithm 1, the iterations terminate.

As opposed to $\mathbb{L}_0$, we use $\mathbb{L}_1$ to denote the preference lists at the end of the proposal-deletion process. If a truck's list is empty in $\mathbb{L}_1$, by symmetry the truck will not be on any other trucks' lists. In other words, that truck can be excluded from platooning consideration.

The motivation for performing the above proposal-deletion process is to obtain $\mathbb{L}_1$ which has reduced lengths for some (or all) of the preference lists compared to $\mathbb{L}_0$. On the other hand, this process ensures that an MS-TPP solution in $\mathbb{L}_1$ is also an MS-TPP solution in $\mathbb{L}_0$. This is shown in Proposition 1 below. But before presenting Proposition 1, we first introduce formally the notions of blocking pair and stability of an MS-TPP solution, which are needed in proving the proposition.

Definition 1. *Blocking pair. In a platooning solution, if there exist two trucks u and v in two different platoons that u and v prefer each other to their respective current platooning partners, i.e., u and v want to break away from their current partners and form a new platoon with each other, we call that u and v form a blocking pair.*

In the example shown in Fig. 1, if the platooning solution is $(1,2)$ and $(3,4)$ being the formed platoons, then $(1,3)$ presents a blocking pair (as stated in the first bullet point in Fig. 1). Similarly, for the platooning solution $(1,3)$ and $(2,4)$, $(2,3)$ presents a blocking pair (second bullet point in Fig. 1). For the platooning solution $(1,4)$ and $(2,3)$, $(1,2)$ presents a blocking pair (third bullet point in Fig. 1).

Definition 2. *Stability of an MS-TPP solution. An MS-TPP solution is stable if there is no blocking pair, i.e., there do not exist two trucks u and v in two different platoons in the solution that u and v prefer each other to their current partners. In the taxonomy of coalition formation, this stability is termed core stability* (Hajdukova, 2006).

Proposition 1. *An MS-TPP solution in $\mathbb{L}_1$ is also an MS-TPP solution in $\mathbb{L}_0$.*

Proof. Per the definition of an MS-TPP solution, the proposition holds if we can prove the following two parts. First, the platoons in an MS-TPP solution in $\mathbb{L}_1$ are stable among themselves in $\mathbb{L}_0$. Second, the number of platoons in an MS-TPP solution in $\mathbb{L}_1$ is no less than the number of platoons in an MS-TPP solution in $\mathbb{L}_0$. Proving the first part is trivial: The formed platoons are stable among themselves, regardless of whether they are in $\mathbb{L}_1$ or $\mathbb{L}_0$. So the focus is on proving the second part.

Since the proposal-deletion process is iterative, our proof of the second part takes an inductive approach. Let us use $\mathbb{L}_1^{k-1}$ and $\mathbb{L}_1^k$ to denote the preference lists before and after the k th iteration (so $\mathbb{L}_1^0 = \mathbb{L}_0$). Suppose that the k th iteration pertains to a truck u proposing to the first truck on its current list, say v. v must accept u because otherwise, v must hold a more preferred truck which, per the proposal-deletion process, must have led to deletion of u from v's list and in turn deletion of v from u's list earlier. The difference between $\mathbb{L}_1^{k-1}$ and $\mathbb{L}_1^k$ is that truck pairs (w,v) with $u\succ_v w$ are in $\mathbb{L}_1^{k-1}$ but not in $\mathbb{L}_1^k$.

Now consider an MS-TPP solution M^{k-1} in $\mathbb{L}_1^{k-1}$. If M^{k-1} does not contain any platoon (w,v) with $u\succ_v w$, then M^{k-1} is obviously also an MS-TPP solution in $\mathbb{L}_1^k$. If M^{k-1} contains platoon (w_0,v) where w_0 is one of such w's, we can construct a new MS-TPP solution in $\mathbb{L}_1^{k-1}$ from M^{k-1} that is also in $\mathbb{L}_1^k$. To do so, we note that u must not be in a platoon in M^{k-1} because in $\mathbb{L}_1^{k-1}$, v is u's most preferred truck. If u is in a platoon, then its partner must be less preferred than v. Considering further that v prefers u to w_0, u and v would form a blocking pair, which means that M^{k-1} is not stable and thus a contradiction.

From M^{k-1}, we construct a new solution $\widetilde{M}^{k-1} = M^{k-1}\backslash(w,v)\cup(u,v)$. From M^{k-1} to $\widetilde{M}^{k-1}$, u gets its most preferred partner (v); v also gets more preferred partners (u instead of w). As M^{k-1} is stable in $\mathbb{L}_1^{k-1}$ and the remaining platoons in $\widetilde{M}^{k-1}$ are the same as in $M^{k-1}\backslash(w,v)$ – which is also stable, by adding (u,v) to $\widetilde{M}^{k-1}\backslash(w,v)$ neither u nor v would be better off by breaking away and forming a new platoon with another truck from an existing platoon. Thus, $\widetilde{M}^{k-1}$ is also an MS-TPP solution in $\mathbb{L}_1^{k-1}$. Noting further that $\widetilde{M}^{k-1}$ is in $\mathbb{L}_1^k$, $\widetilde{M}^{k-1}$ is a stable truck platooning participation (S-TPP) solution in $\mathbb{L}_1^k$. Suppose that M^k is an MS-TPP solution in $\mathbb{L}_1^k$. Then, the number of platoons in $\widetilde{M}^{k-1}$ must be no greater than that in M^k, i.e., $\left|\widetilde{M}^{k-1}\right| \leq \left|M^k\right|$. Since $\left|M^{k-1}\right| = \left|\widetilde{M}^{k-1}\right|$, we have $|M^{k-1}| \leq |M^k|$. By inductively performing the above, we can arrive at the conclusion that the number of formed platoons in $\mathbb{L}^1$ is no less than the number of formed platoons in $\mathbb{L}_0$. Thus, the second part is proven. This completes the proof. ■

As a result of the proposal-deletion process, an important property of $\mathbb{L}_1$ is that v is the first truck on u's preference list if and only if u is the last truck on v's preference list in $\mathbb{L}_1$. This property will be very useful when performing phase 2. Below we give a proof of this property.

Lemma 1. *For any $\mathbb{L}_1$ obtained after the proposal-deletion process, v is the first truck on u's preference list if and only if u is the last truck on v's preference list.*

Proof. We need to prove both directions. First, if v is the first truck on u's preference list in $\mathbb{L}_1$, per the proposal-deletion process, it must be that u proposed to v and v holds the proposal of u. Then again according to the proposal-deletion process, all trucks less preferred than u will be deleted from v's preference list. In other words, u is v's last truck on v's preference list.

Now we prove the other direction. If u is the last truck on v's preference list in $\mathbb{L}_1$, then per the proposal-deletion process, it must be that v accepted and holds the proposal of u as u proposed to v, which means v is the first truck on u's preference list. This completes the proof. ■

This property can be easily verified in $\mathbb{L}_1$ (the preferences shown in step 5) of the example in Table 2. Truck 2 is the first one on truck 1′s list, while truck 1 is the last one on truck 2′s list. Truck 4 is the first one on truck 2′s list, while truck 2 is the last one on truck 4′s list. Truck 1 is the first one on truck 3′s list, while truck 3 is the last one on truck 1′s list. Finally, truck 3 is the first one on truck 4′s list, while truck 4 is the last one on truck 3′s list.

In general, the preference lists of some trucks in $\mathbb{L}_1$ contain multiple trucks. However, a special case is that each truck has at most one truck on its preference list. In this case, we can readily identify an MS-TPP solution without proceeding to phase 2. Proposition 2 shows this.

Proposition 2. *If each truck has at most one truck on its preference list in $\mathbb{L}_1$, an MS-TPP solution can be readily identified, without proceeding to phase 2.*

Proof. Let us randomly pick a truck u whose preference list in $\mathbb{L}_1$ has only one truck v. Then u and v form a platoon in $\mathbb{L}_1$. We repeatedly do this for all trucks whose preference lists in $\mathbb{L}_1$ have only one truck. By symmetry, the number of these trucks is even, say $2n$. So they end up forming n platoons, which is the maximum possible number of platoons that can be formed out of the $2n$ trucks. Note that no two trucks from two different platoons want to break away from their current platooning partners and form a new platoon with each other, since no such two trucks are on each other's lists. So, the n formed platoons are stable among themselves, which give an MS-TPP solution in $\mathbb{L}_1$. Using Proposition 1, this MS-TPP solution in $\mathbb{L}_1$ is also an MS-TPP solution in $\mathbb{L}_0$. This completes the proof. ■

If some truck preference lists in $\mathbb{L}_1$ have multiple trucks, phase 2 will be needed, to find an MS-TPP solution in $\mathbb{L}_0$.

5.2. Phase 2: Rotation identification and separation/elimination

When some preference list(s) in $\mathbb{L}_1$ have more than one truck, phase 2 proceeds which involves further reducing $\mathbb{L}_1$ up to a point where an MS-TPP solution can be easily constructed. Phase 2 starts by checking if there is any truck with an empty preference list in $\mathbb{L}_1$. As mentioned in subsection 5.1, such trucks will not platoon with any other trucks. Thus, their preference lists can be removed. The resulting reduced preference lists is termed $\mathbb{L}_2^0$. If no truck has an empty list in $\mathbb{L}_1$, $\mathbb{L}_2^0 = \mathbb{L}_1$. Then, phase 2 continues to iteratively identify and separate/eliminate "rotations", each of which is an artificial construct of a cyclic sequence of trucks, until no more rotation can be identified. During this iterative process, we use $\mathbb{L}_2^k$ to denote the preference lists after k iterations, i.e., after k rotations are identified and separated/eliminated. The preference lists at the end of the iterative process is denoted by $\mathbb{L}_2$.

Because rotation plays a central role in the iterative process in phase 2, our exposition below starts with defining a rotation.

Definition 3. *Rotation exposed in $\mathbb{L}_2^k$. A rotation $\rho = (u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is a cyclic sequence of distinct trucks $u_1, u_2, \cdots, u_r$, where truck $v_i, i = 1, 2, \cdots, r$ is the most preferred truck of u_i and also the second most preferred truck of u_{i-1}, based on the preferences specified by $\mathbb{L}_2^k$. i is taken modulo r. So, truck v_1 is the most preferred truck of u_1 and the second most preferred truck of u_r.*

For the example in Table 2, $(1,4)|(2,3)$ is a rotation exposed in $\mathbb{L}_2^0$ (in this example, $\mathbb{L}_2^0$ consists of the preference lists of trucks 1–4). In $\mathbb{L}_2^0$, truck 2 is the most preferred truck of truck 1, and the second most preferred truck of truck 4. Truck 3 is the most preferred truck of truck 4, and the second most preferred truck of truck 1.

Given $\mathbb{L}_2^k$ in the iterative process, we need to first determine if a rotation can be identified. Lemma 2 shows that as long as there is a truck whose preference list has more than one truck, a rotation exposed in $\mathbb{L}_2^k$ exists and can be identified.

Table 3
$\mathbb{L}_2^1$ after eliminating rotation $(1,4)|(2,3)$ exposed in $\mathbb{L}_2^0$ of Table 2.

Truck 1	3
Truck 2	4
Truck 3	1
Truck 4	2

Lemma 2. *Given $\mathbb{L}_2^k$, if there is a truck whose preference list has more than one truck, then a rotation exposed in $\mathbb{L}_2^k$ exists and can be identified.*

Proof. Suppose that such a truck is u_1. Let us use $f(u_1), s(u_1), l(u_1)$ to denote the first, the second, and the last truck on L^k_{2,u_1}, which denotes u_1's list in $\mathbb{L}_2^k$. If L^k_{2,u_1} has only two trucks, $s(u_1) = l(u_1)$. Now, we identify the least preferred truck of $s(u_1)$, denoted as $u_2 = l(s(u_1))$. Note that by symmetry and Lemma 1, u_1 will be on $s(u_1)$'s list but is not the least preferred truck. So $s(u_1)$ also has more than one truck on its preference list $L^k_{2,s(u_1)}$. From u_2, we do the same and identify $u_3 = l(s(u_2))$. This process repeats. Because the number of trucks is finite, we will encounter a truck u_r that $l(s(u_r)$ is a previously visited truck, i.e., $l(s(u_r) \in \{u_1, u_2, \cdots, u_r\}$. Without loss of generality, suppose $l(s(u_r) = u_1$. Then, trucks $u_1, u_2, \cdots, u_r$ are distinct and form a cycle: $u_{i+1} = l(s(u_i), i = 1, 2, \cdots, r$ modulo r. Modulo r means $u_1 = l(s(u_r)$. From Lemma 1, we know that $f(u_{i+1}) = s(u_i)$. So by Definition 3, $(u_1, u_2, \cdots, u_r)|(s(u_r), s(u_1), \cdots, s(u_{i-1}))$ is a rotation. This completes the proof. ■

Given a rotation, we now further introduce the definition of rotation elimination.

Definition 4. *Rotation elimination. Suppose that $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is a rotation exposed in $\mathbb{L}_2^k$. So $v_1 = f(u_1) = s(u_r)$, $v_2 = f(u_2) = s(u_1)$, etc. The rotation is said eliminated if for each $v_{i+1}, i = 1, \cdots, r$, we delete all trucks from $L^k_{2,v_{i+1}}$ that are less preferred than u_i (i is taken modulo r), i.e., delete all $w \in L^k_{2,v_{i+1}}$ that $w \prec_{v_{i+1}} u_i$. By symmetry, we also delete truck v_{i+1} from the preference lists of these w's.*

Let us use the rotation $(1,4)|(2,3)$ exposed in $\mathbb{L}_2^0$ of Table 2 to illustrate how rotation elimination works. For truck 2, we delete all trucks in truck 2′s list that are less preferred than truck 4. So, truck 1 is deleted from truck 2′s list. By symmetry, truck 2 is deleted from truck 1′s list. Then we move to truck 3. We delete all trucks in truck 3′ list that are less preferred than truck 1. So, truck 4 is deleted from truck 3′s list. By symmetry, truck 3 is deleted from truck 4′s list. After the deletion (eliminating the rotation), the resulting $\mathbb{L}_2^1$ is shown in Table 3.

The rotation elimination can be viewed as a modified proposal-deletion process applied to the rotation, with the modification that each u_i in a rotation proposes to the second truck on its preference list. While there seems no intuitive meaning for rotation elimination at first sight, Propositions 3–5 below show that rotation elimination plays a key role in reducing $\mathbb{L}_2^0$ toward the final $\mathbb{L}_2$. These propositions involve whether a rotation is odd, which we define below.

Definition 5. *Odd rotation. If after eliminating a rotation $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$, some truck $u_i(i = 1, \cdots, r)$'s preference list becomes empty, then the rotation is called an odd rotation. Otherwise, the rotation is a non-odd rotation.*

Let us again use the rotation $(1,4)|(2,3)$ exposed in $\mathbb{L}_2^0$ of Table 2 to illustrate. The rotation is non-odd, since no preference list becomes empty after eliminating the rotation (as shown in Table 3). As we will see in Proposition 3 below, the name "odd rotation" is related to the fact that such a rotation has an odd number r. Also, it can be shown that the set of u's and the set of v's in an odd rotation are the same and have the same ordering, although u_i and $v_i, i = 1, \cdots, r$ are different. Moreover, each u_i has exactly two trucks on its preference list. These properties are proven below.

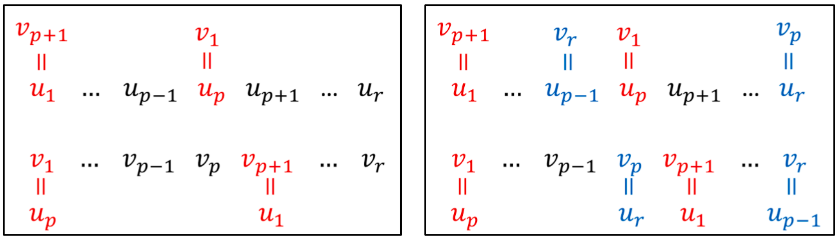
Proposition 3. *Suppose that $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is an odd rotation exposed in $\mathbb{L}_2^k$. After eliminating this rotation, some truck u_i's preference list becomes empty. Then the following are true:*

(i) For each truck $u_i, i = 1, \cdots, r$, its preference list has exactly two trucks.
(ii) $u_i = v_{p+i}, \forall i = 1, \cdots, r$ (subscript modulo r), where $2p - 1 = r$.

Proof. Without loss of generality, suppose that u_1's list becomes empty after eliminating the rotation. Before eliminating the rotation, v_2 is the second truck on u_1's list. v_2 is deleted from u_1's list only if u_1 is the second truck on some truck u_p's list and $u_p \succ_{u_1} v_2$. (Note that it cannot be that v_2 is the second truck on u_p's preference list and $u_p \succ_{v_2} u_1$, because in this case, $v_{p+1} = v_2$. Per the rotation definition and Lemma 1, u_{p+1} is the last truck on v_{p+1} (v_2)'s list and u_2 is the last truck on v_2's list. But by definition of a rotation, u_{p+1} and u_2 should be distinct. Thus a contradiction.).

Given $u_p \succ_{u_1} v_2$ and that v_2 is the second truck on u_1's list, u_p must be the first truck on u_1's list, which is v_1. So, $u_p = v_1$. Moreover, given that u_1 is the second truck on u_p's list and per the definition of a rotation, $u_1 = v_{p+1}$. Considering further that u_1 is the last truck on v_1 (which is equal to u_p)'s list, u_p's list has only two trucks, v_p and v_{p+1}.

The above shows: 1) u_p's list has exactly two trucks; 2) $u_1 = v_{p+1}$; and 3) $v_1 = u_p$ for a specific p (red color in Fig. 4(a)). So, v_1's list, which is the same as u_p's list, also just has two trucks. Per the definition of rotation, these two trucks are u_1 and u_r. As u_p's list has v_p and v_{p+1}, and $u_1 = v_{p+1}$, we must have $u_r = v_p$. Consequently, u_r ($= v_p$) is the first truck on v_1 ($= u_p$)'s list, and per Lemma 1 v_1 is the last



(a) (b)

Fig. 4. Illustration of equalities between u's and v's.

truck on u_r's list. By rotation definition, v_1 is the second truck on u_r's list. Thus, u_r's list also has exactly two trucks. Considering that u_r's list has only two trucks v_r and v_1, v_p $(= u_r)$'s list must also have only two trucks, which are u_{p-1} and u_p. Since $v_1 = u_p$, it must be that $v_r = u_{p-1}$. In summary, we have shown further that: 1) u_r's list has exactly two trucks; 2) $u_r = v_p$; and 3) $v_r = u_{p-1}$ (blue color in Fig. 4(b)).

By inductively performing the above procedure, we can conclude that u_i's list has exactly two trucks for all $i = 1,\cdots,r$. In addition, we will have two sets of equalities. The first set is: $u_1 = v_{p+1}, u_2 = v_{p+2}, \cdots$ (subscript modulo r). The second set is: $u_p = v_1, u_{p+1} = v_2, \cdots$ (subscript modulo r). Because the two sets describe the same relationships between u_i's and v_i's, we have from the second set $u_1 = v_{r-p+2}$ (note $u_r = v_{1+(r-p)}$. Because subscript is modulo r, we have $u_1 = u_{r+1} = v_{r-p+2}$), which is also equal to v_{p+1} from the first set. Thus, $r - p + 2 = p + 1$, which means $2p - 1 = r$. In other words, we have $u_i = v_{p+i}, \forall i = 1,\cdots,r$ (subscript modulo r), where $2p - 1 = r$. This completes the proof. ■

From Proposition 3, we can see that the preference lists of trucks in an odd rotation are independent of trucks not involved in the rotation. This is because any truck u_i in an odd rotation has only two trucks on its list: v_i and v_{i+1}, which are u_{i-p} and u_{i-p+1} (subscript modulo r) in the odd rotation. Thus, if an odd rotation exposed in $\mathbb{L}_2^k$ is identified, the preference lists of trucks in the odd rotation can be separated from $\mathbb{L}_2^k$, leading to $\mathbb{L}_2^{k+1}$ which correspond to the preference lists of the remaining trucks. An MS-TPP solution in $\mathbb{L}_2^k$ can be sought by seeking an MS-TPP solution in the odd rotation and an MS-TPP solution in $\mathbb{L}_2^{k+1}$ separately, and then combine the two solutions. We formalize this as Proposition 4.

Proposition 4. *If an odd rotation exposed in $\mathbb{L}_2^k$ is identified, an MS-TPP solution in $\mathbb{L}_2^k$ can be sought by independently seeking an MS-TPP solution in the odd rotation and an MS-TPP solution in $\mathbb{L}_2^{k+1}$, and then combine the two solutions.*

It is also possible that a rotation exposed in $\mathbb{L}_2^k$ is non-odd, i.e., no truck's list after rotation elimination becomes empty. In this case, suppose that eliminating the rotation leads to $\mathbb{L}_2^{k+1}$. Then, we claim that any MS-TPP solution in $\mathbb{L}_2^{k+1}$ is also an MS-TPP solution in $\mathbb{L}_2^k$. To show this, we first note that going from $\mathbb{L}_2^{k+1}$ to $\mathbb{L}_2^k$, which means adding back the eliminated truck pairs from $\mathbb{L}_2^{k+1}$ to $\mathbb{L}_2^k$, the formed platoons in an MS-TPP solution in $\mathbb{L}_2^{k+1}$ do not change. So the stability among the formed platoons is intact. What remains to be shown is that the number of formed platoons in an MS-TPP solution in $\mathbb{L}_2^{k+1}$ is the same as the number of formed platoons in an MS-TPP solution in $\mathbb{L}_2^k$, which is true if at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$. We show this by introducing two lemmas below.

Lemma 3. *Suppose that $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is a non-odd rotation exposed in $\mathbb{L}_2^k$. In addition, suppose that there exists an MS-TPP solution M in $\mathbb{L}_2^k$ where no u_i and v_i form a platoon with each other for $i = 1, \cdots, r$. After eliminating the rotation, the preference lists become $\mathbb{L}_2^{k+1}$. Then, at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$.*

Proof. The idea of the proof is similar to that for Proposition 1. We want to show that: (1) the formed platoons in an MS-TPP solution in $\mathbb{L}_2^{k+1}$, M', remain stable among themselves in $\mathbb{L}_2^k$; and (2) $|M'| \geq |M|$. The first part is trivial. So the focus of our proof is on the second part.

Two cases are possible for the formed platoons in M with respect to the rotation elimination:

Case 1. M does not contain any platoon (w,v_i), $i = 1,\cdots,r$, where $w \prec_{v_i} u_{i-1}$. In this case, the formed platoons in M remain intact after eliminating the rotation, i.e., these platoons remain intact and stable among themselves in $\mathbb{L}_2^{k+1}$. Per the "maximum" definition of an MS-TPP solution in $\mathbb{L}_2^{k+1}$, it must be that $|M'| \geq |M|$.

Case 2. M contains at least one formed platoon of (w,v_i), $i = 1,\cdots,r$, where $w \prec_{v_i} u_{i-1}$. In this case, we first claim that u_{i-1} must not be in a formed platoon in M. This is because in $\mathbb{L}_2^k$, v_i is u_{i-1}'s second most preferred truck. If u_{i-1} were in a formed platoon, then its partner would be less preferred than v_i considering that (u_{i-1}, v_{i-1}) is not a formed platoon in M and v_i platoons with $w \prec_{v_i} u_{i-1}$. Thus, (u_{i-1}, v_i) would present a blocking pair for M, which contradicts stability.

Now, we construct a new solution $M'' = M \backslash (w,v_i) \cup (u_{i-1},v_i)$. From M to M'', v_i gets a more preferred partner u_{i-1}. As M is stable in $\mathbb{L}_2^k$, v_i would not be better off by breaking away from u_{i-1} in M''. u_{i-1} also would not be able to break away from v_i. This is because the only situation that u_{i-1} breaks away is that (u_{i-1}, v_{i-1}) presents a blocking pair. But per the definition of a rotation, u_{i-1} is the least preferred truck of v_{i-1}. In other words, v_{i-1}, if in a formed platoon, would not want to break up with its current partner to platoon with u_{i-1}. So, M'' is stable among the formed platoons. As the number of formed platoons is unchanged from M to M'', M'' is an MS-TPP solution in $\mathbb{L}_2^k$. If there are multiple (w, v_i) pairs in M, the above procedure will be performed multiple times one for each pair to obtain M''.

Note that M'' is in $\mathbb{L}_2^{k+1}$. So, $|M'| \geq |M''|$. In other words, $|M'| \geq |M|$. Thus the second part is proven. This completes the proof. ■

Building on Lemma 3, a more general claim that at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$ after eliminating a non-odd rotation can be made, as in Lemma 4.

Lemma 4. *Suppose that $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is a non-odd rotation in $\mathbb{L}_2^k$. After eliminating the rotation, the preference lists become $\mathbb{L}_2^{k+1}$. Then, at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$.*

Proof. Let M denote an MS-TPP solution in $\mathbb{L}_2^k$. In M, three cases can be possible with respect to u_i and v_i $(i = 1, \cdots, r)$:

Case 1. No u_i and v_i, $i = 1, \cdots, r$, form a platoon with each other in M. In this case, Lemma 3 has shown that at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$.

Case 2. All u_i and v_i, $i = 1, \cdots, r$, form platoons with each other in M. In this case, let us consider a new platooning solution M' from M: we remove (u_i, v_i)'s and add (u_i, v_{i+1})'s for $i = 1, \cdots, r$. In other words, $M' = M \backslash \{(u_i, v_i)\}_{i=1,\cdots,r} \cup \{(u_i, v_{i+1})\}_{i=1,\cdots,r}$. Compared to M where each v_i platoons with its least preferred truck u_i, all v_i's in M' are better off by platooning with u_{i-1}. The only trucks which fare slightly worse in M' are u_i's, as each u_i now platoons with its second preferred truck v_{i+1} instead of the most preferred truck v_i. So, if any instability arises, it must be from (u_i, v_i)'s presenting blocking pairs. But now v_i platoons with a more preferred truck u_{i-1}. (u_i, v_i)'s cannot be blocking pairs. Thus, the new platooning solution M', which has the same number of formed platoons as M, is an MS-TPP solution in $\mathbb{L}_2^k$. Per Lemma 3, at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$.

Case 3. Some but not all (u_i, v_i)'s form platoons in M. Suppose that for some j, u_j and v_j form a platoon, but u_{j-1} and v_{j-1} do not. For u_{j-1}, there can be two possibilities. The first possibility is that u_{j-1} platoons with some truck $w \neq v_{j-1}$. For stability it must be that $w \succ_{u_{j-1}} v_j$ because otherwise, (u_{j-1}, v_j) would present a blocking pair (note that $u_{j-1} \succ_{v_j} u_j$). But per rotation definition, w can only be v_{j-1}, which is a contradiction. So the first possibility indeed cannot happen.

The second possibility is that u_{j-1} is not in a platoon in M. We consider a new matching $M'' = M \backslash (u_j, v_j) \cup (u_{j-1}, v_j)$. So in M'', u_j and v_j do not form a platoon. If M has more platoons like (u_j, v_j), the above is repeated until getting a matching M''' where no u_i and v_i, $i = 1, \cdots, r$, form a platoon with each other. Following the same argument as for case 2 in the proof of Lemma 3, it can be shown that M''' is an MS-TPP solution in $\mathbb{L}_2^k$. Per Lemma 3, at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$. This completes the proof. ■

With Lemma 4, we now show that any MS-TPP solution in $\mathbb{L}_2^{k+1}$ is also an MS-TPP solution in $\mathbb{L}_2^k$ if the eliminated rotation is non-odd.

Proposition 5. *Suppose that $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is a non-odd rotation in $\mathbb{L}_2^k$. After eliminating the rotation, the preference lists become $\mathbb{L}_2^{k+1}$. Then, any MS-TPP solution in $\mathbb{L}_2^{k+1}$ is also an MS-TPP solution in $\mathbb{L}_2^k$.*

Proof. Suppose M' is an MS-TPP solution in $\mathbb{L}_2^{k+1}$. Lemma 4 suggests that, when the eliminated rotation is non-odd, $|M'| = |M|$ where M denotes an MS-TPP solution in $\mathbb{L}_2^k$. Because the truck platoons in M' do not change after adding back the eliminated truck pairs, the stability among the formed platoons in M' remains intact in $\mathbb{L}_2^k$. So, M' is also an MS-TPP solution in $\mathbb{L}_2^k$. This completes the proof. ■

It is worth having a brief comparison of Lemmas 3–4 and Proposition 5 with a similar theorem (theorem 3.7) in Tan (1990). In our exposition, we first frame the objective as showing that any MS-TPP solution in $\mathbb{L}_2^{k+1}$ is also an MS-TPP solution in $\mathbb{L}_2^k$. Doing so requires showing both "maximum" and "stability". As "stability" can be intuitively seen, the focus of the proof is on the "maximum", for which three steps are taken. Step 1 considers just one case that an MS-TPP solution in $\mathbb{L}_2^k$ has no u_i and v_i in the non-odd rotation form a platoon, and shows that at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$ (Lemma 3). Step 2 builds and expands the conclusion of step 1, by showing more generally that after eliminating a non-odd rotation, at least one MS-TPP solution in $\mathbb{L}_2^k$ is also an MS-TPP solution in $\mathbb{L}_2^{k+1}$. In step 3, the conclusion from step 2 is leveraged to demonstrate that any MS-TPP solution in $\mathbb{L}_2^{k+1}$ is also an MS-TPP solution in $\mathbb{L}_2^k$. Different from this, Tan's theorem focuses on stability with a quite lengthy proof which involves making a new claim (which, when put in the truck platooning context, would be that an MS-TPP solution in $\mathbb{L}_2$ with no (u_i, v_i) forming a platoon exists) and considering whether u's and v's have common elements, both unnecessary in our proof.

With Propositions 4 and 5, we propose Algorithm 2 to seek an MS-TPP solution starting from $\mathbb{L}_2^0$. At the k th iteration, we identify a rotation from $\mathbb{L}_2^k$ (line 3). If the rotation is odd, we separate the rotation by storing it and removing the lists of all u's involved in the rotation from $\mathbb{L}_2^k$ (lines 5–6). If the rotation is non-odd, eliminate the rotation (line 8). Then we update the preference lists to $\mathbb{L}_2^{k+1}$ (line 11). The iterative procedure continues until no more rotation can be identified.

Algorithm 2: Seeking an MS-TPP solution in $\mathbb{L}_2^0$

(continued on next page)

(*continued*)

Algorithm 2: Seeking an MS-TPP solution in $\mathbb{L}_2^0$

1 **input:** preference lists $\mathbb{L}_2^0$ from phase 1
2 **set** $k = 0$
3 **while** a rotation exposed in $\mathbb{L}_2^k$ can be identified
4   **if** the rotation is odd, **do**
5   Store the rotation
6   Remove the lists of all u's involved in the rotation from $\mathbb{L}_2^k$
7   **else**
8   Eliminate the rotation
9   **end if**
10   **set** $\mathbb{L}_2^{k+1}$ to be the current preference lists
11   **update** $k = k + 1$
12 **end while**
13 Pair trucks based on the final preference lists $\mathbb{L}_2 = \mathbb{L}_2^k$
14 **for** each odd rotation $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$, **do**
15   Form truck pairs (u_1, u_p), (u_2, u_{p+1}), $\cdots$, (u_{p-1}, u_{2p-2}) where $2p - 1 = r$
16   Remove u_{2p-1}
17 **end for**
18 Combine the truck pairs obtained from lines 11–15, which form an MS-TPP solution in $\mathbb{L}_2^0$

After performing line 12 of Algorithm 2, we will have 1) a set of odd rotations and 2) a set of trucks each having only one truck on its list (if the preference list of any truck has more than one truck, per Lemma 2 a rotation exposed in the preference lists can be identified). From Propositions 4 and 5, an MS-TPP solution in $\mathbb{L}_2^0$ can be sought by independently seeking an MS-TPP solution in $\mathbb{L}_2$ (line 13) and in each odd rotation (lines 14–17), and then combining the MS-TPP solutions (line 18). In $\mathbb{L}_2$, each truck has only one truck on its list. So forming platoons is simply pairing trucks that indicate preference of each other. The formed platoons are stable, because no two trucks in two different platoons are on each other's lists.

For trucks in an odd rotation, recall that the number of trucks in an odd rotation is odd (Proposition 3(ii)). The maximum possible number of truck platoons that can be formed from an odd rotation $(u_1, u_2, \cdots, u_r)|(v_1, v_2, \cdots, v_r)$ is $(r - 1)/2$, by removing one truck and pairing the remaining trucks. Also recall that each u_i in an odd rotation has only two trucks on its list: v_i and v_{i+1} (Proposition 3(i)). If we pair u_1 with v_1 (which is u_p with $2p - 1 = r$ per Proposition 3(ii) and also shown in Fig. 4), then v_2, which is u_{p+1}, can only pair with u_2. By induction, the formed platoons are (u_1, u_p), (u_2, u_{p+1}), $\cdots$, (u_{p-1}, u_{2p-2}). Because each u_i ($i = 1, \cdots, p - 1$) platoons with its most preferred truck, these formed platoons are stable among themselves. This leaves u_{2p-1} (or u_r) unmatched, which will be removed. Consequently, (u_1, u_p), (u_2, u_{p+1}), $\cdots$, (u_{p-1}, u_{2p-2}) form an MS-TPP solution in the odd rotation. As u_i's and v_i's are cyclic (Proposition 3 (ii)), we can remove any truck depending on which truck is viewed as u_1 in the above pairing. This does not change stability among the formed platoons.

With the two-phase algorithmic approach fully described, we formalize the claim that this approach leads to an optimal solution, i.e., a solution that is MS-TPP.

Proposition 6. *The two-phase algorithmic approach yields an optimal solution, i.e., a solution that is MS-TPP.*

Proof. We prove optimality of the solution from the two-phase algorithmic approach in a backward fashion. We know that performing the first part of phase 2 (up to line 12 in Algorithm 2) leads to decomposition of the truck population into two groups: (1) trucks that are only left with one platooning partner and 2) trucks that are in odd rotations. From Propositions 4 and 5, an MS-TPP solution in $\mathbb{L}_2^0$ can be obtained by performing the second part of phase 2 (after line 12 in Algorithm 2), i.e., forming platoons independently for trucks in group (1) and for trucks in each odd rotation in group (2). Recall that $\mathbb{L}_2^0$ is $\mathbb{L}_1$ excluding trucks that have empty preference lists in $\mathbb{L}_1$. Thus, the obtained MS-TPP solution in $\mathbb{L}_2^0$ is also an MS-TPP solution in $\mathbb{L}_1$, which is the truck preference lists after performing phase 1. By further recalling Proposition 1, we can claim that this MS-TPP solution is an MS-TPP solution in $\mathbb{L}_0$, the original truck preference lists. So by performing the two-phase algorithmic approach, the obtained solution is an MS-TPP solution with respect to the original truck preference lists. This completes the proof. ■

5.3. Computational complexity investigation

After detailing the two phases, it is useful to gain some further understanding about the computational complexity of the overall approach. The detailed analysis is presented in Proposition 7 below.

Proposition 7. *For n trucks, the computational complexity of the two-phase approach while seeking an MS-TPP solution is $O(n^3)$.*

Proof. We investigate the computational complexity of each phase, based on which the overall computational complexity is concluded. Phase 1 consists of each truck u proposing to its most preferred truck v and v deleting trucks that are less preferred than u on v's list. A truck can propose to up to $(n - 1)$ trucks. So, the total number of proposals for the n trucks is bounded by $n(n - 1)$. A truck can delete up to $n - 1$ trucks from its list. So the total number of deletions for the n trucks is bounded by $n(n - 1)$. As each proposal and each deletion correspond to one operation, the computational complexity of phase 1 is $O(n^2)$.

As illustrated in Algorithm 2, phase 2 consists of three parts: rotation identification, rotation separation (odd rotation)/elimination (non-odd rotation), and formation of truck platoons. For rotation identification, the worst case is that we go through all trucks before uncovering a rotation (recall the proof of Lemma 2). As a result, the computational complexity for identifying one rotation is $O(n)$. We need to know how many rotations are identified. To do so, suppose that the length of the first identified rotation is r_1 (i.e., the rotation would look like $(u_1, u_2, \cdots, u_{r_1})|(v_1, v_2, \cdots, v_{r_1})$). If the rotation is odd, each of $u_1, u_2, \cdots, u_{r_1}$ has exactly two trucks on its preference list. The rotation separation is equivalent to deleting two trucks from each of $u_1, u_2, \cdots, u_{r_1}$'s lists. Thus in total $2r_1$ deletions. If the rotation is non-odd, at least $2r_1$ truck deletions will be performed (deleting u_i from v_i's list and vice versa, for $i = 1, \cdots, r_1$). Let M denote the total number of rotations identified in phase 2. $2r_1 + 2r_2 + \cdots + 2r_M$ is bounded by the total number of trucks in all trucks' preference lists, which is at most $n(n-1)$. Thus, $2r_1 + 2r_2 + \cdots + 2r_M \le n(n-1)$. Considering further that $r_m \ge 2, m = 1, 2, \cdots, M$, M is bounded by $n(n-1)/4$. So overall, the computational complexity for rotation identification is $O(Mn) = O(n^3)$.

For rotation separation (odd rotation)/elimination (non-odd rotation), since the total number of trucks in all trucks' preference lists is bounded by $n(n-1)$, the computational complexity of rotation separation/ elimination, which is about separating the preference lists of some trucks (for odd rotations) or deleting some trucks from truck preference lists (for non-odd rotations), does not exceed $O(n^2)$. For formation of truck platoons, given that the number of formed platoons is at most $n/2$, the computational complexity is $O(n)$. Putting the three parts together, the computational complexity of phase 2 is dominated by rotation identification. The overall computational complexity of the two-phase approach is dominated by phase 2, and equal to $O(n^3)$. This completes the proof. ■

6. An integer programming approach

This section provides an alternative approach to solve the MS-TPP problem by integer programming. While the two-phase algorithmic approach is theoretically guaranteed to lead to an MS-TPP solution, pursuing an integer programming approach serves for two purposes: first, the integer programming approach provides a further verification of the MS-TPP solution correctness of the two-phase algorithmic approach, in terms of the maximum number of platoons formed; second, the integer programming approach provides a benchmark for the solution time needed, which will show the computational advantage of the algorithmic approach. To our knowledge, no prior work exists on developing a mathematical programming model for the MS-TPP problem. On the other hand, inspired by earlier investigations of the relationship between stable matching and linear inequalities (Abeledo and Ruthblum, 1994; Teo and Sethuraman, 2000), we propose the following integer linear programming (ILP) model to solve MS-TPP:

$$\max \sum_{u:L_u \neq \emptyset} \sum_{v \in L_u} x_{u,v} \tag{4}$$

s.t.

$$\sum_{v \in L_u} x_{u,v} \le 1, \ \forall u : L_u \neq \emptyset \tag{5}$$

$$x_{u,v} = x_{v,u}, \ \forall (u,v): \ v \in L_u \text{ and } u \in L_v \tag{6}$$

$$\sum_{i \prec_u v, \ i \in L_u} x_{u,i} + \sum_{j \prec_v u, \ j \in L_v} x_{j,v} \le 1, \quad \forall (u,v): \ v \in L_u \text{ and } u \in L_v \tag{7}$$

$$x_{u,v} \in \{0,1\}, \quad \forall (u,v) \in \mathbb{L} \tag{8}$$

In the ILP, the decision variables are $x_{u,v}$'s which indicate whether truck u forms a platoon with truck v. The objective function (4) maximizes the number of platoons formed. We first sum over all trucks (u's) whose preference lists are non-empty. For each such truck u, we sum over all the partner trucks on its preference list. Constraint (5) says that any truck u whose preference list is non-empty can platoon with at most one truck on its preference list. Constraint (6) describes symmetry in truck pairing: for any v that is on u's preference list (in turn u is on v's preference list), $x_{u,v}$ and $x_{v,u}$ must be equal. This is intuitive as $x_{u,v}$ means that u forms a platoon with v, and $x_{v,u}$ means that v forms a platoon with u. Constraint (7) characterizes stability, ensuring that any two trucks u and v, which are in two different platoons, cannot break away from their current platoons and form a new one. This would happen if truck u platoons with a truck that is less preferred than v, and truck v platoons with a truck that is less preferred than u. So it cannot be that $\sum_{i \prec_u v, \ i \in L_u} x_{u,i} = 1$ and $\sum_{j \prec_v u, \ j \in L_v} x_{j,v} = 1$ at the same time. In other words, the left-hand side of constraint (7) cannot be two (but can be equal to one if only one of the equalities holds). Finally, constraint (8) specifies the binary requirement for the decision variables. Given that the problem of finding a stable matching that maximizes a linear function is NP-hard (Feder, 1992; Abeledo and Rothblum, 1994), ILP (4)-(8) is NP-hard.

7. Numerical experiments

In this section, the MS-TPP problem is solved by both the two-phase algorithmic and the integer programming approaches numerically in a regional road network setting (Fig. 5). We start by describing the experiment setup in subsection 7.1. In subsection 7.2, we discuss results from the base scenario with 1,000 trucks. The discussions include the computational aspects of the two-phase algorithmic approach in comparison with the integer programming approach, and the platooning outcomes by solving the MS-TPP

Fig. 5. The sketchy northern Illinois road network considered for the numerical experiments.

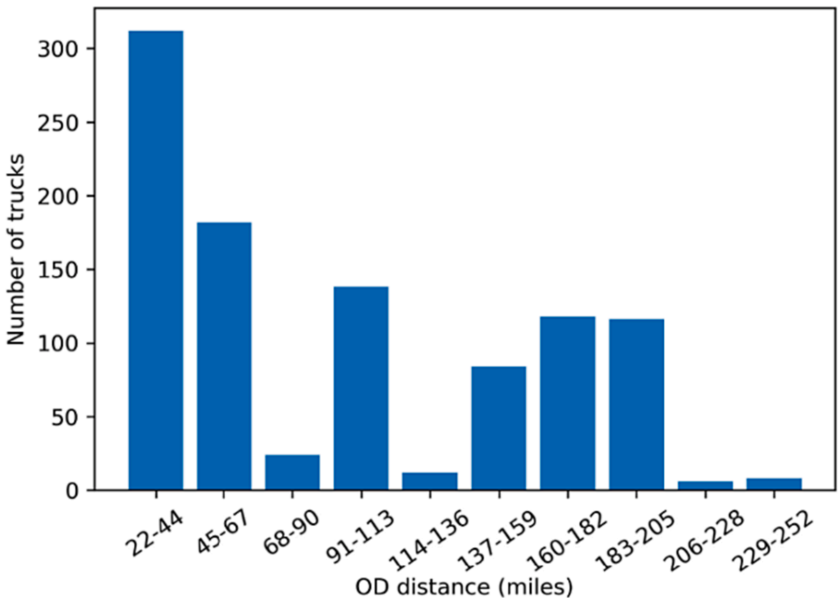


Fig. 6. Distribution of truck traffic by OD distance.

problem. As scalability is of particular interest, we devote subsection 7.3 to examining additional scenarios with larger problem sizes. All experiments are conducted on a PC with Intel Core (TM) i7 2600 3.40 GHz CPU, 16 GB RAM, and Windows 10 operating system. The two-phase algorithmic approach is coded in Python 3.6. The ILP model is coded and solved using branch-and-bound in CPLEX 12.10.

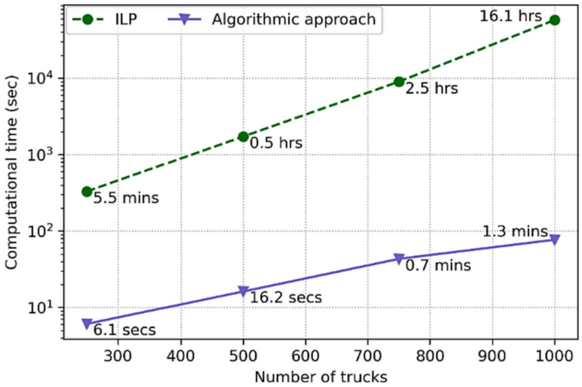


Fig. 7. Average computation time for the algorithmic and the integer programming approaches, over five randomly generated instances for each problem size.

7.1. Experiment setup

We consider a sketchy northern Illinois road network for our numerical experiments (Fig. 5). This network connects nine major cities in the state (Chicago, Aurora, Joliet, Rockford, Springfield, Peoria, Elgin, Champaign, and Waukegan) which serve as origins and destinations of truck trips. Thus, the network has 72 different ODs. Two cities are connected by one direct link or a set of consecutive links. Fig. 5 displays these cities (red triangles) and the connected links. The intersection points of the links are also shown as nodes in the network (grey nodes). In total, the sketchy northern Illinois road network has 17 nodes and 26 bidirectional links. The lengths of the links are presented in Table A1 of Appendix A.

While data on the actual truck traffic flow on the network are not available, we consider a base scenario of 1,000 trucks in the network for a period of four hours, and three alternative scenarios with 2,000, 3,000, and 4,000 trucks for the same period length. For the base scenario, we assign the 1,000 trucks to the 72 ODs based on weights of the ODs. The weight of an OD is constructed as the product of the origin and destination cities' population, divided by the square of the total population of the nine cities. The city population information is obtained from the World Population Review (2021). By doing so, population is used as a proxy for a city's ability to generate and attract truck traffic. The weights of the ODs and the truck traffic volume for the base scenario are shown in Tables A2 and A3 of Appendix A. As a crosscheck, the resulting OD truck traffic between Champaign and Chicago on a per hour basis is found comparable with a recent estimate using the Freight Analysis Framework (FAF) data from the US Federal Highway Administration (Noruzoliaee et al., 2021; FAF, 2019). Note that the actual truck traffic on the road links is higher because of traffic of other ODs, which is especially relevant to the Chicago metropolitan region which is a national freight hub. Fig. 6 shows the distribution of truck traffic by OD distance. A significant portion of the traffic is for short distances, which is attributed to the population concentration in Chicago and cities in the surrounding area (Waukegan, Elgin, Aurora, and Joliet).

Given the truck volume for each OD, we assume that each truck intends to travel along the shortest distance route. The preferred departure time of a truck is randomly generated during the four-hour period. Because of the randomness, for each scenario a number of problem instances are tested. In Section 7.2, 50 instances are tested except for (1) comparing with the integer programming approach, for which the average results are based on five instances; and (2) comparing with maximization of system utility gain, for which 20 instances are tested. The two exceptions are made because both involve solving ILP which takes a very long time. In Section 7.3, 20 instances are tested for each of the scenarios. We consider truck fuel efficiency at 6.5 miles per gallon (Schoettle et al., 2016). The average speed on the network is 60 mph. When platooning, the inter-truck distance is assumed at 0.2 s, which is associated with an average fuel saving per platooning truck at 7.1 % (McAuliffe et al., 2018). We use this average to compute fuel saving benefits for all the formed platoons in the experiments. The fuel cost is assumed at $5.5/gallon and the waiting time cost at $0.60/min.

7.2. Results from the base scenario (1,000 trucks)

This subsection presents the numerical experiment results for the base scenario with 1,000 trucks. We first focus on the computational aspects of the results for the two-phase algorithmic approach in comparison with the integer programming approach (Section 7.2.1). Then we compare the MS-TPP outcome with a greedy approach for platoon formation, and investigate the outcome in terms of truck departure delay, platooning length, fuel saving, and utility gain (Section 7.2.2).

7.2.1. Investigation of the computational aspects

Fig. 7 shows the average computation time needed for the algorithmic and the integer programming approaches, averaged over five randomly generated instances. To provide additional comparisons, we also solve smaller instances of 750, 500, and 250 trucks, each again with five randomly generated instances, and report the computation time. Because the variation of computation time across instances for each problem size is very small, only the averages are reported. For all problem instances, we find that the number of formed platoons is the same using the two approaches, which empirically validates the theoretical correctness of the algorithmic

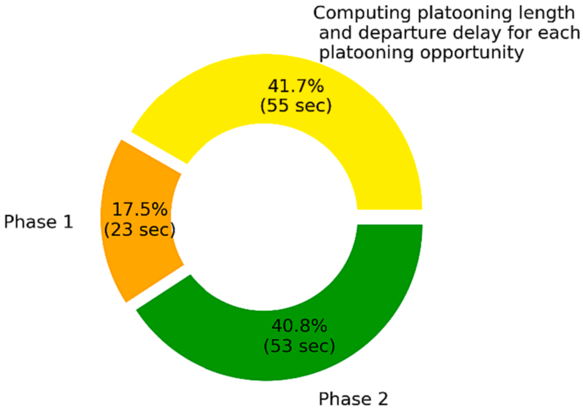


Fig. 8. Distribution of computation time needed by the platform.

Table 4
Truck pairs and rotations information when performing the two-phase algorithmic approach over 50 problem instances: average and standard deviation (in parentheses).

Start	Number of truck pairs in preference lists	1,281 (11.8)
Phase 1	Number of truck pairs eliminated	527 (5.3)
Phase 2	Number of odd rotations identified	13 (0.4)
	Number of non-odd rotations identified	73 (4.0)
	Number of truck pairs eliminated with non-odd rotations	414 (3.6)

approach.

Computationally, the algorithmic approach demonstrates a significant advantage: for 250 trucks the algorithmic approach only takes on average 6.1 s to solve the MS-TPP problem. By contrast, the integer programming approach takes 331 s (5.5 min), or 54 times the computation time. The contrast becomes even more prominent as the problem size gets larger. For 1,000 trucks the computation time for the algorithmic and the integer programming approaches is 77 s (1.3 min) and 57,893 s (16.1 h), or 751 times of difference. Noting that the vertical axis is in logarithmic scale, the computation time by the integer programming approach follows a highly nonlinear increasing trend (for the problem size from 250 to 1,000 trucks – a four-time increase, the computation time increase is 175 times; in contrast, the computation time increase for the algorithmic approach is only 13 times). Clearly, the algorithmic approach is the preferred choice for tackling the MS-TPP problem.

To further understand the computational aspects of the algorithmic approach, we decompose the computation time spent by the platform into three parts: 1) the time for computing platooning length and departure delay for each of the platooning opportunities; 2) the time for performing phase 1; and 3) the time for performing phase 2. These times are averages over 50 randomly generated instances. The decomposition results are shown in Fig. 8. It can be seen that computing platooning lengths and departure delays, which is performed following the procedure described in section 4 prior to executing the two-phase algorithm, takes a considerable portion (41.7 %) in the total computation time. However, as our computation deals with one platooning opportunity at a time, the computation time can be substantially reduced if parallel computing (i.e., multiple platooning opportunities are simultaneously examined) is employed. Between the two phases of the algorithm, phase 1 takes less than half the time of phase 2, which is not surprising as phase 1 performs simpler operations of proposing and deletion, while phase 2 involves operations of greater complexity including identifying

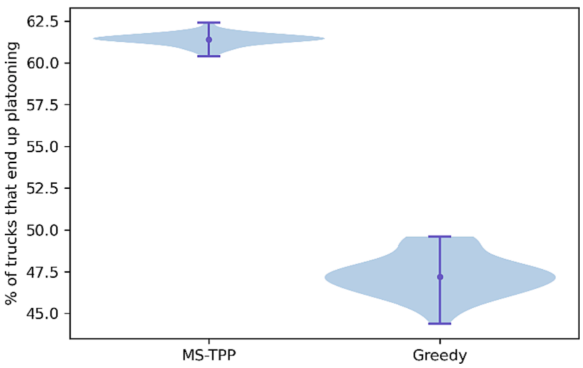


Fig. 9. Percentage of trucks in platooning with MS-TPP and a greedy approach under the base scenario.

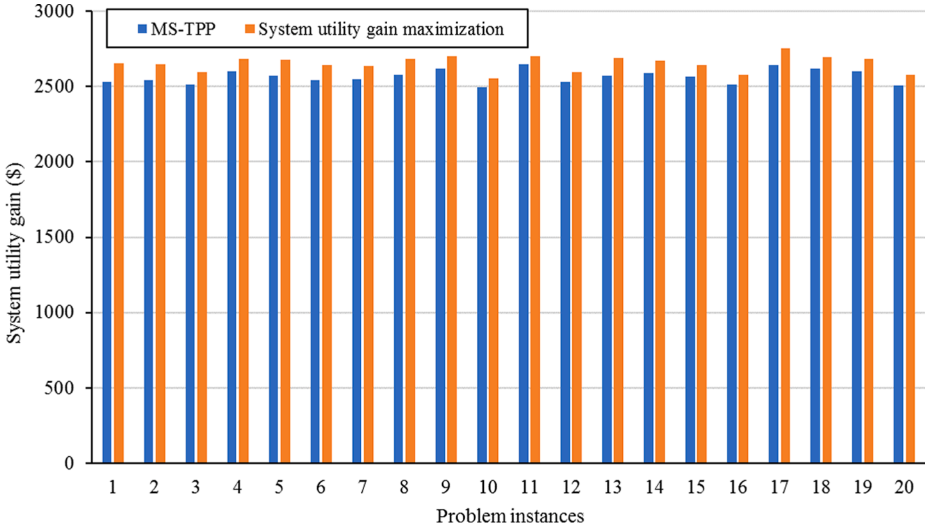


Fig. 10. Comparison of system utility gain under MS-TPP and system utility gain maximization.

rotations, determining whether a rotation is odd, and performing rotation separation (if odd) and elimination (if non-odd).

Table 4 provides additional information about the algorithmic computation, in terms of the number of truck pairs (i.e., platooning opportunities) and rotations. The numbers are again averages over the 50 instances, with the standard deviations in parentheses. It can be seen that the standard deviations are quite small compared to the averages. Through the proposal-deletion process in phase 1, about 40 % (527/1,281) of the truck pairs are eliminated. In phase 2, the number of non-odd rotations identified (73) is much greater than the number of odd rotations identified (13). From the non-odd rotations, another 32 % (414/1,281) of the truck pairs are eliminated.

7.2.2. Investigation of the platooning outcomes

While Section 7.2.1 focuses on the computational aspects of the two-phase algorithmic approach, this subsection looks into the outcomes from solving the MS-TPP problem. The left part of Fig. 9 shows a violin plot of the percentage of trucks that end up platooning across the 50 problem instances (where the curve corresponds to a kernel density estimation flipped vertically). The distribution has a relatively small range between about 60 % and 62.5 %, with the median percentage at 60.8 %.

For comparison, we also perform a greedy approach where a truck only myopically tries to form a platoon with a peer truck that has the same origin and the same destination, and is next to it in terms of preferred departure time. More specifically, for each OD, trucks are first sorted by their preferred departure time. Then, we investigate forming a platoon between the first two trucks, i.e., the trucks that have the earliest and the second earliest preferred departure time. If both trucks gain positive utilities, the platoon is formed. We then move to looking into the third truck platooning with the fourth truck. If the first truck does not gain a positive utility (due to the incurred departure delay cost exceeding the fuel saving benefit), the truck will not gain a positive utility by platooning with any later trucks since it would incur even greater departure delay. Thus, the first truck will travel alone. We move to looking into the second truck platooning with the third truck. This process continues until reaching the last truck of the OD.

The right part of Fig. 9 shows a violin plot of the percentage of trucks that end up platooning across the same 50 instances using the greedy approach. We see that the greedy approach yields a lower percentage of trucks in platooning, with the median value at 47.2 %. In addition, the variation of the percentage is greater, from below 45 % to about 50 %, which may be attributed to the fact that the greedy approach offers each truck fewer platooning opportunities. Furthermore, the formed platoons are not guaranteed to be stable. Given a larger number of platoons formed, less variation in the platoon formation results, and stability among the platooning trucks, taking an MS-TPP perspective is clearly superior to the greedy approach for truck platooning.

Another comparison we make is on the system utility gains between MS-TPP solutions and solutions that maximize system utility gain, which does not account for stability and is often considered in the truck platooning literature. The comparison is useful to understand the utility gain gap if the intent is to maximize the number of platooning trucks, rather than to maximize system utility gain. Given a problem instance, the maximization of system utility gain can be sought by formulating and solving the following ILP model:

$$\max \sum_{u:L_u \neq \varnothing} \sum_{v \in L_u} w_{u,v} x_{u,v} \tag{9}$$

s.t.

$$\sum_{v \in L_u} x_{u,v} \leq 1, \quad \forall u, L_u \neq \varnothing \tag{10}$$

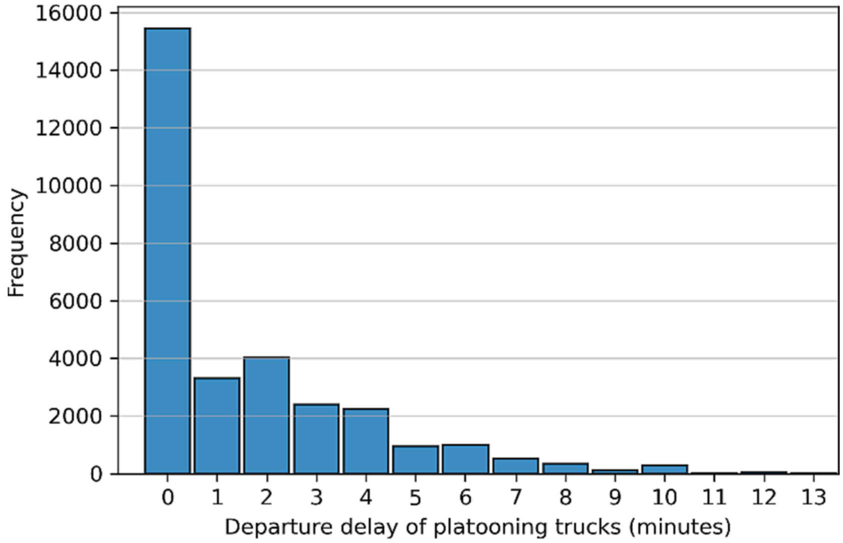


Fig. 11. Departure delay distribution among platooning trucks under the base scenario.

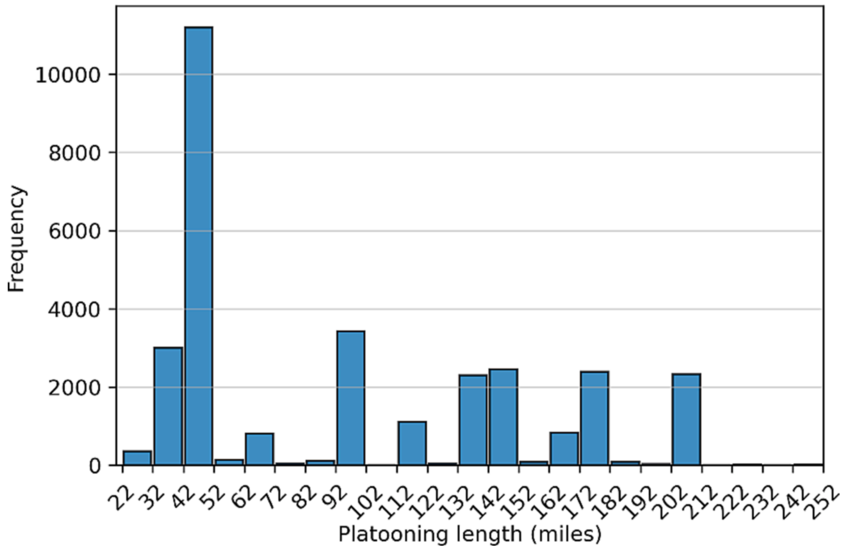


Fig. 12. Platooning length distribution among platooning trucks under the base scenario.

$$x_{u,v} = x_{v,u}, \quad \forall (u,v): \ v \in L_u \text{ and } u \in L_v \tag{11}$$

$$x_{u,v} \in \{0,1\}, \quad \forall (u,v) \in \mathbb{L} \tag{12}$$

where the decision variables $x_{u,v}$ are the same as in ILP (4)-(8). The sum of utility gains of trucks u and v when forming a platoon with each other is denoted by $w_{u,v}$. Recalling Eq. (1), $w_{u,v} = \Delta U_{u(v)} + \Delta U_{v(u)}$. Constraints (10)-(12) are the same as (5)-(6) and (8) respectively. As solving ILP (9)-(12) for one problem instance takes about 3–4 h, here the comparison is made for 20 instances. Fig. 10 shows the resulting system utility gains under MS-TPP and system utility gain maximization. We can see that the difference is quite small. On average across the 20 instances, the system utility gains under MS-TPP and system utility gain optimization are \$2,566 and \$2,653 respectively, or 3.4 % difference (with MS-TPP as the base).

Figs. 11-13 give more details about the MS-TPP outcome using results from all 50 instances. A histogram of departure delay of platooning trucks is plotted in Fig. 11. We observe that overall, the departure delay is small in the order of minutes. A significant portion of the platooning trucks actually have zero departure delay. This is because for a platoon, only the truck that arrives earlier at the start of the platooning length will incur a departure delay, while the other truck will not. For the platooning trucks with small non-zero departure delay, the small delay also makes intuitive sense given the stability requirement: if departure delay were too large, it

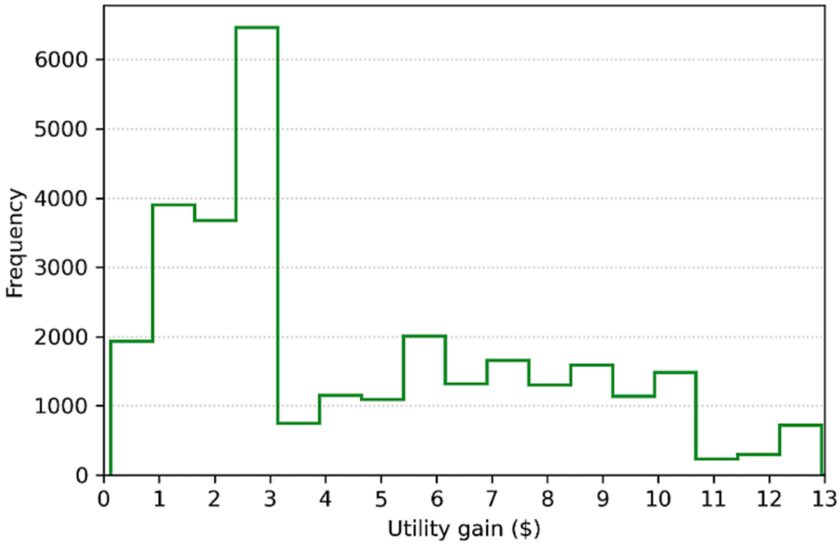


Fig. 13. Utility gain distribution among platooning trucks under the base scenario.

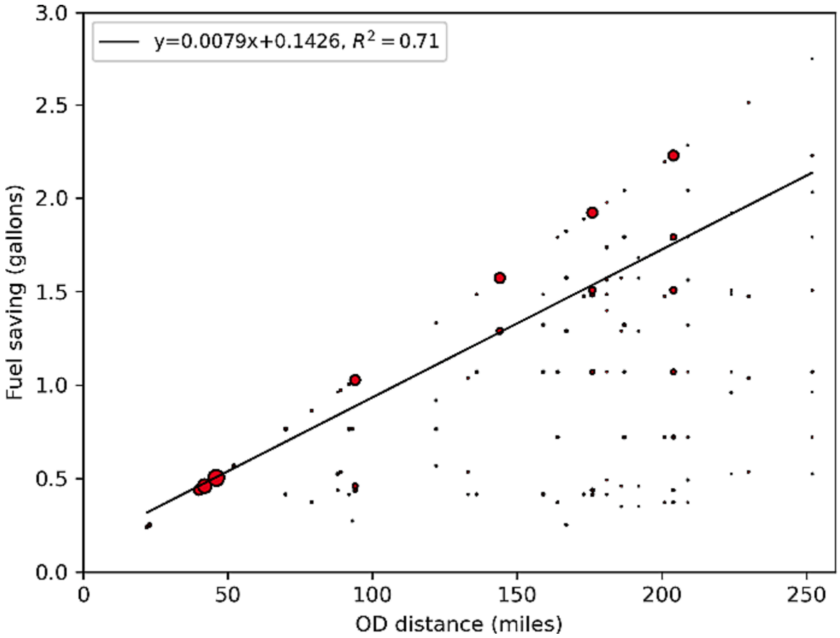


Fig. 14. Scatterplot of platooning truck fuel saving vs. OD distance and the associated regression line under the base scenario.

would mean that trucks made lots of schedule deviation efforts to form a platoon. Consequently, trucks in such platoons would likely be less stable and break away to platoon with another truck that would require less schedule deviation effort.

Fig. 12 shows a histogram of the platooning length of platooning trucks. In principle, the distribution of platooning lengths is affected by the OD distance distribution (as shown in Fig. 6) and truck traffic on these ODs. We can see that a large number of platoons travel for a distance of around 42–52 miles. The concentration of platooning lengths in this short distance range is attributed to the heavy truck traffic volume between Chicago and cities in the surrounding area. On the other hand, over 2,000 platoons are observed in several other distance ranges (92–102, 132–152, 172–182, and 202–212 miles), which may be associated with the non-trivial truck traffic between cities in the Chicago metropolitan region and farther-away cities such as Springfield, Champaign, and Peoria.

For the utility gain of platooning trucks, Fig. 13 shows that most platooning trucks will receive a utility gain between $0 and $3, while the platooning trucks with a utility gain ranging between $4 and $11 are more or less evenly distributed. The fairly small utility gains are the result of a relatively small network considered (northern part of Illinois, which itself is a medium-size state in the US) and a relatively small number of trucks (1,000). It is worth recalling that the actual number of trucks traveling on the network is larger than

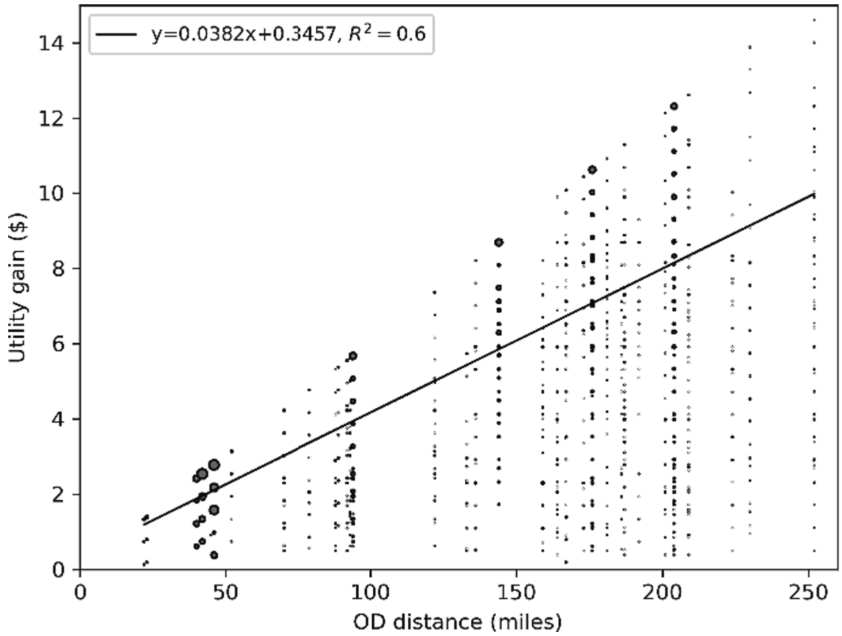


Fig. 15. Scatterplot of platooning truck utility gain vs. OD distance and the associated regression line under the base scenario.

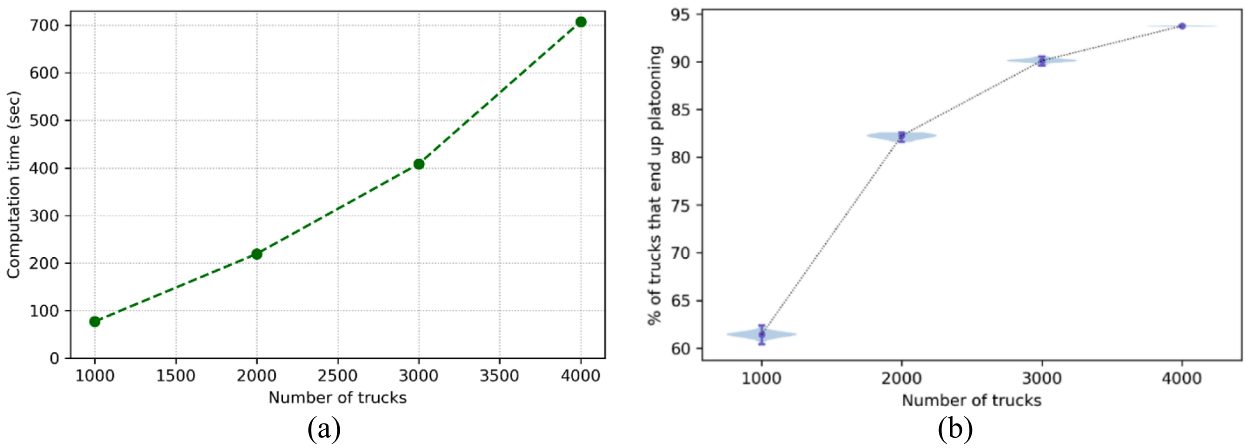


Fig. 16. Impact of the number of trucks on computation time (a) and percentage of trucks platooning (b).

the ones considered here because of truck traffic of other ODs beyond the network. The implications of a larger truck traffic volume will be investigated in Section 7.3.

Lastly, we investigate the relationship of fuel saving and utility gain of platooning trucks as a function of OD distance. Figs. 14-15 present the scatterplots of platooning trucks over the 50 instances, in which larger dots correspond to a greater number of data points taking the same value, which can happen when the platooning length of a number of trucks is the same. For example, trucks of the same OD can form and travel in platoon throughout the entire trip, resulting in the same fuel saving for these trucks.[1] In each of these platoons, at least one truck will have zero departure delay. Thus, these trucks with zero departure delay will have the same utility gain. Following this thought, it is not surprising to see that the scatterplot of Fig. 14 is upper-bounded by dots which form a straight line if connected. The straight line will cross the origin as fuel saving is proportional to the platooning length, which is the OD distance for these trucks. In the utility gain plot of Fig. 15, the upper-bounded dots correspond to a subset of the trucks whose departure delay is zero. Thus, utility gain of these trucks is only fuel saving benefit, which is proportional to OD distance.

In Figs. 14-15, we also run regression based on the scatterplots. The regression results, displayed on top of each figure, show that on

[1] Recalling section 3, we consider that fuel savings of a platoon are evenly shared between the two trucks in the platoon.

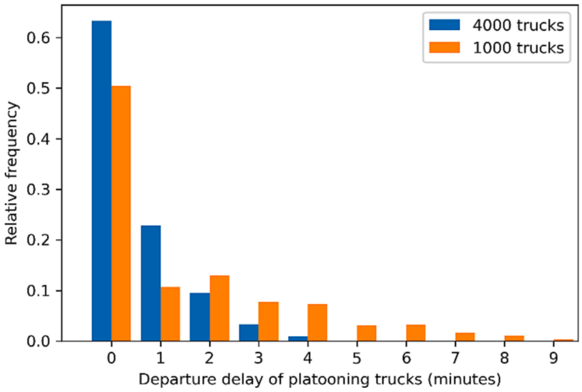


Fig. 17. Departure delay distributions among platooning trucks under the base and the 4,000-truck scenarios.

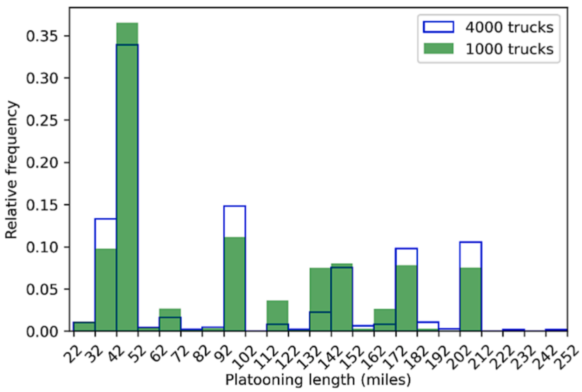


Fig. 18. Platooning length distributions among platooning trucks under the base and the 4,000-truck scenarios.

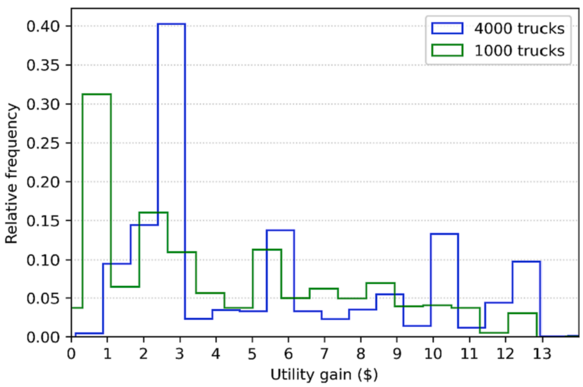


Fig. 19. Utility gain distributions among platooning trucks under the base and the 4,000-truck scenarios.

average, a 10-mile OD distance increase will lead to 0.079 gallon of fuel saving and $0.382 utility gain for a platooning truck in the studied network. The 0.079 gallon of fuel saving is less than fuel saving when a truck platoons all the way along the 10-mile distance, which would be $\frac{10 \text{ miles}}{6.5 \text{ miles/gallon}} \times 7.1\% = 0.11$ gallon. This is not surprising as on average a truck will not platoon all the way in its journey. Similarly, due to departure delay cost, the marginal utility gain is less than the marginal fuel saving benefit, which is 0.079 gallon $\times \frac{\$5.5}{\text{gallon}} = \$0.43$.

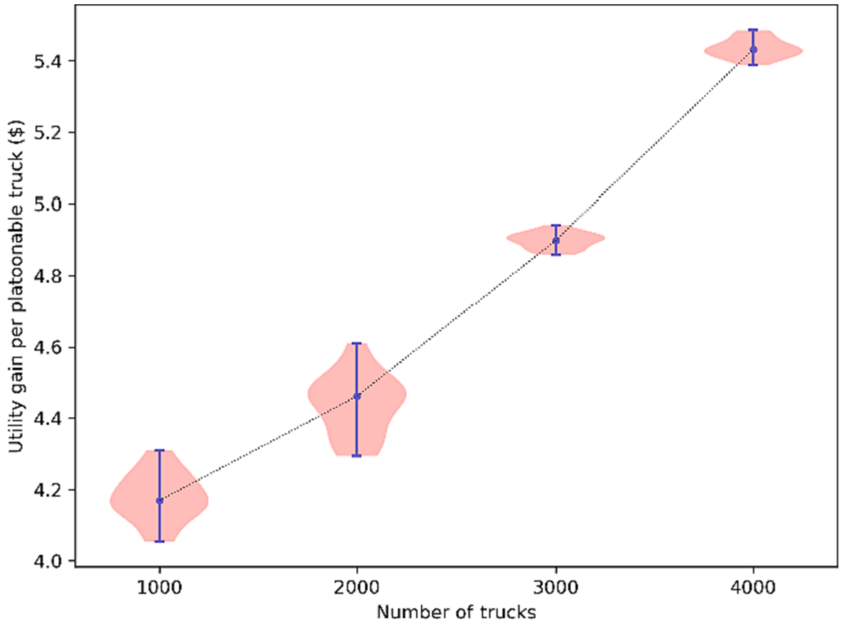


Fig. 20. Utility gain per platooning truck under different scenarios.

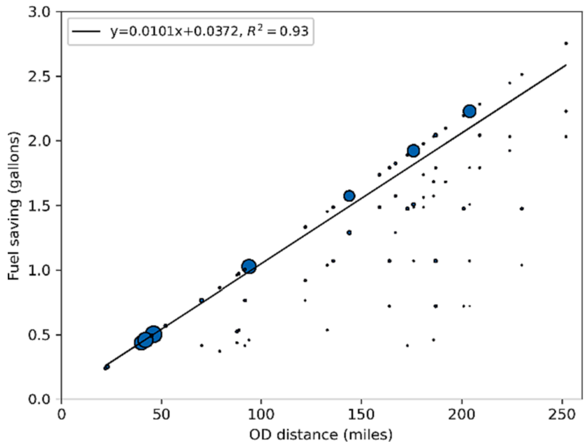


Fig. 21. Scatterplot of platooning truck fuel saving vs. OD distance and the associated regression line under the 4,000-truck scenario.

7.3. Results from alternative scenarios with more trucks

To further investigate scalability of MS-TPP and the two-phase algorithmic approach, in this subsection we consider three alternative scenarios with the total number of trucks at 2,000, 3,000, and 4,000. For each scenario, we randomly generate and then solve 20 instances. The remaining setup is the same as for the base scenario. As mentioned earlier, considering a larger number of trucks is relevant given that the Chicago metropolitan region is a national freight hub with inbound/outbound/through truck traffic beyond the OD traffic considered in the base scenario. Although the origin and/or the destination of the additional trucks may not be in the network, we consider that when traversing a route in the network, a truck has an expected leaving time from the start of the route, which may be viewed as its preferred departure time. In addition, to gain a positive utility these trucks are willing to delay the leaving time from the start of the route in order to form and travel in a platoon in the network.

Fig. 16(a) shows a slightly more-than-linear increasing trend of the average computation time over the 20 instances for each of the four scenarios, as we increase the number of trucks from 1,000 to 4,000. When 4,000 trucks are considered, the average number of truck pairs in the preference lists at the start of the algorithm is 10,761. Solving an instance of such problem size takes about 700 s (or less than 12 min) on average. Given that MS-TPP is an operation planning problem and 4,000 trucks correspond to a large problem size, spending 700 s to solve a problem of this size will not be unacceptable. The empirical computation time increase as a function of the number of trucks presented here suggests that the actual computation effort is likely to be less than the theoretical bound derived in

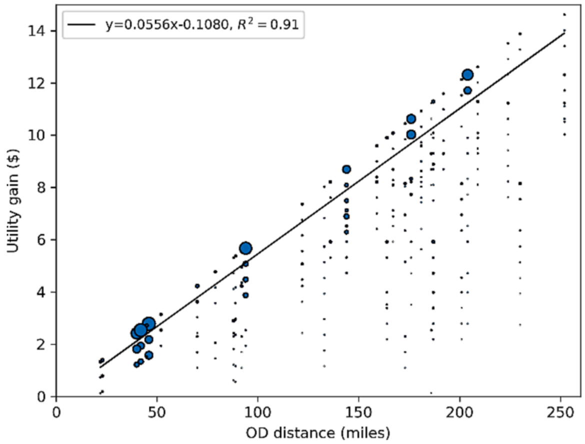


Fig. 22. Scatterplot of platooning truck utility gain vs. OD distance and the associated regression line under the 4,000-truck scenario.

Section 5.3. Fig. 16(b) displays the distribution of the percentage of trucks that end up platooning over the 20 instances for each of the four scenarios. It is found that a greater portion of trucks will platoon as more trucks are considered. This is not surprising, as more trucks present in the network provide a truck with more platooning opportunities. Another observation is that as the number of trucks increases, stochasticity across problem instances diminishes. In other words, as each truck is given more platooning opportunities, the system will become more certain about the percentage of trucks that end up in platooning.

As a result of more platooning opportunities, considering a larger number of trucks brings greater benefits to platooning trucks in the forms of shorter departure delay, longer platooning length, and more utility gain. Figs. 17-20 plot histograms of departure delay, platooning length, and utility gain of platooning trucks for the base (1,000 trucks) and the 4,000-truck scenarios. It should be noted that the histograms for the base scenario here are not exactly the same as in Section 7.2, because here we use 20 randomly generated instances instead of 50 in Section 7.2. Fig. 17 suggests that when 4,000 trucks are in the system, a higher portion of trucks will be able to find a close-by truck – both spatially and temporally – to form a platoon with zero or one minute of departure delay. In contrast, the portion of platooning trucks experiencing two or more minutes of departure delay is reduced.

With more trucks as potential platooning partners, a truck is likely to be able to platoon for a longer distance. This is shown in Fig. 18 with multiple higher bars at longer platooning lengths. Combining the reduced departure delay and greater fuel saving benefits due to platooning for longer distances, the average utility gain per platooning truck is expected to increase. Fig. 19 appears to confirm this speculation, with a larger portion of trucks enjoying higher utility gains (higher blue bars at large utility gain values). A further comparison of the utility gain per platooning truck across the four scenarios is illustrated in violin plots of Fig. 20. The figure clearly shows that as the number of trucks increases, the average utility gain per platooning truck also increases, from $4.17 per platooning truck with 1,000 trucks to $5.43 with 4,000 trucks, or about 30 % increase. In addition, there seems to be a trend of reduced variation across different problem instances as more trucks are considered. These results suggest that if different areas were to be selected for MS-TPP implementation, areas with denser truck traffic should be given higher priority.

Figs. 21-22 present the scatterplots of platooning truck fuel saving and utility gain as a function of OD distance for the 4,000-truck scenario. All data from the 20 instances are shown on the figures. Similar to Figs. 14-15, larger dots in these two figures correspond to a greater number of data points from the 20 instances that take the same value. Most of the larger dots are located along the upper boundary associated with trucks that travel in platoon throughout their trips (in Fig. 22, those trucks will have zero departure delay). In Fig. 21, the regression result indicates a larger marginal effect of OD distance on fuel saving for the 4,000-truck case than for the 1,000-truck case. In fact, based on the regression equations fuel saving will always be larger for the 4,000-truck case across almost all OD distances (except for very small OD distances). This is not surprising, as more trucks allow for greater and better platooning opportunities leading to more fuel saving. The goodness-of-fit measured in R^2 is better for the 4,000-case as well, at 0.93 as opposed to 0.71. This can be attributed to a greater portion of the formed platoons consisting of two trucks of the same OD, in which case platooning truck fuel saving is proportional to OD distance. This, together with the reduced departure delay, also leads to a substantially increased R^2 (from 0.6 in Figs. 15 to 0.91 in Fig. 22) when regressing utility gain on OD distance. The regression line slope is increased from 0.0382 to 0.0556, which is slightly smaller than the fuel saving benefit when traveling an additional mile in platooning ($\frac{1\ \text{mile}}{6.5\ \text{miles/gallon}} \times 7.1\% \times \frac{\$5.5}{\text{gallon}} = \$0.06$). As not all trucks platoon throughout the entire OD distance and departure delay still occurs to some platooning trucks, the negative intercept reflects these effects on the utility gain.

8. Summary, discussions of limitations, and future research

In this research, we propose a platform-based platooning system to maximize participation of two-truck platooning considering stability of the formed platoons which arises from truck preferences for platooning partners. The preferences depend on the benefits of fuel saving and schedule adjustment to coordinate the time for platoon formation. We consider truck preferences such that all

Table A1
Link lengths of the network (in miles).

Node	Node	Length	Node	Node	Length
Rockford	Rochelle	25	Decatur	Champaign	49
Rockford	Elgin	52	Elgin	Aurora	22
Rochelle	Lasalle	43	Elgin	Mayfair	32
Rochelle	Aurora	45	Aurora	Joliet	23
Peoria	Lincoln	45	Aurora	Naperville	11
Peoria	Bloomington	38	Joliet	Chicago	40
Lincoln	Springfield	34	Joliet	Homewood	26
Lincoln	Bloomington	32	Champaign	Homewood	118
Springfield	Decatur	40	Naperville	Chicago	33
Lasalle	Bloomington	67	Waukegan	Mayfair	38
Lasalle	Joliet	56	Mayfair	Chicago	10
Bloomington	Decatur	46	Chicago	Homewood	26
Bloomington	Joliet	98	Bloomington	Champaign	50

platooning trucks receive a positive utility gain compared to traveling alone (thus trucks have an incentive to participate). Moreover, the formed platoons are stable in the sense of core stability that no two trucks in two platoons would break away from their current platoons and platoon with each other. Tackling this operation planning problem, the proposed system involves a platform interacting with individual trucks in a way that reduces truck communication and computation burdens, mitigates truck privacy concerns of sharing sensitive proprietary information with other trucks, and overcomes the issue of trucks misreporting private information. The central methodological investigation of the interactive process is on how to solve the MS-TPP problem, for which a two-phase algorithmic approach is proposed. The idea of this approach is to progressively reduce the lengths of truck preference lists, by eliminating truck pairs that do not affect the MS-TPP solution presence and separating trucks in odd rotations from the rest of the truck population.

The algorithmic approach is theoretically and empirically examined, the latter through a comparison with an integer programming approach and extensive numerical experiments. We find much reduced computation time using the algorithmic approach compared to solving the ILP model, and substantially greater platooning participation of trucks under MS-TPP than using a greedy approach. The algorithmic approach is computationally scalable for solving large problem instances. The advantages of MS-TPP in terms of the percentage of trucks in platooning and the average utility gain become more prominent as we deal with a larger system with more trucks. Overall, the proposed MS-TPP problem as embedded in the platform-based platooning system and the algorithmic approach demonstrate a potential to engage trucks in the platooning practice to receive energy and economic benefits.

This study presents a start of a new perspective of maximizing truck participation in platooning. As such, the study does have some limitations. Below we point out three of them and offer our preliminary thoughts for future exploration. First, while there are evidences of the relevance of two-truck platooning to practice (as argued in section 1), it will be interesting to consider forming platoons of larger (but still realistic) sizes, say up to three trucks in a platoon. To accommodate such larger platoons, how to extend/adapt the current algorithmic approach will be an interesting research direction. One possible way is to convert the new problem into a two-truck platooning problem, by introducing virtual trucks (e.g., trucks A, B, and C platooning may be considered as a virtual truck D composed of trucks A and B platooning with truck C). To do so, characterizing the relation between a virtual truck and its component physical trucks would be required. For example, virtual truck D would be on truck C's list but not on A's or B's. The utility of virtual truck D should be specified as some function of and align with A and B's utilities. Moreover, some new algorithmic operations would be needed, to ensure that a truck does not end up in multiple formed platoons on one road segment due to the existence of a virtual truck and its component physical trucks.

Second, while the current work focuses on finding one MS-TPP solution, further efforts can be made to seek multiple solutions if they exist. One way to do so could be alternating truck orders in the proposal-deletion process and in rotation identification. In fact, we have tried 50 experiments for a problem instance of 1,000 trucks. In these experiments, we randomly alternate the order of trucks to be chosen while identifying rotations. We find that these 50 runs all yield the same number of truck platoons (608), as expected, but the solutions can differ in the truck composition of the formed platoons. Thus, the platform may need to choose one out of the many solutions in actual implementation. To do so, additional criteria would need to be conceived and perhaps integrated in the algorithmic approach. For instance, if it is desired to have platooning trucks come from a wide geographic coverage rather than from specific regions, we could adopt metrics that measure each truck's OD distinction from the population average. One such metric can be $\sqrt{\left(d_u^O\right)^2 + \left(d_u^D\right)^2}$, where d_u^O (d_u^D) is the distance from the truck's origin (destination) to the average origin (destination) of the truck population. The metric will be used to prioritize truck selection while performing the algorithmic approach, such as in the proposal-deletion process of phase 1 and in rotation identification of phase 2.

Third, in our paper each truck is assumed to have one route. To increase the likelihood of platoon formation, trucks may have some flexibility in routing. Thus, we could consider that each truck has multiple possible routes. Doing so would require augmenting the truck preference lists. Each element of the preference list of a truck u will be a combination of a truck v, and the routes of trucks u and v on which platooning occurs. This is unlike the preference list in this paper where each element is a truck. Correspondingly, adaptations would be needed in rotation identification, as in a rotation u and v will still be trucks. For example, if a truck v holds both the first and the second positions in truck u's preference list (i.e., u's two most preferred platooning possibilities are both with v, though under different routing options), only one of the two positions should be kept while constructing a rotation. This is because otherwise, u's first

Table A2
Weight of each OD.

	Chicago	Aurora	Joliet	Rockford	Springfield	Peoria	Elgin	Champaign	Waukegan
Chicago	0.530	0.039	0.029	0.029	0.023	0.022	0.022	0.017	0.017
Aurora	0.039	0.003	0.002	0.002	0.002	0.002	0.002	0.001	0.001
Joliet	0.029	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.001
Rockford	0.029	0.002	0.002	0.002	0.001	0.001	0.001	0.001	0.001
Springfield	0.023	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Peoria	0.022	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Elgin	0.022	0.002	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Champaign	0.017	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Waukegan	0.017	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Table A3
OD demand in the network (in number of trucks).

	Chicago	Aurora	Joliet	Rockford	Springfield	Peoria	Elgin	Champaign	Waukegan
Chicago	0	85	63	63	49	48	48	37	37
Aurora	85	0	5	5	4	3	3	3	3
Joliet	63	5	0	3	3	3	3	2	2
Rockford	63	5	3	0	3	3	3	2	2
Springfield	49	4	3	3	0	2	2	2	2
Peoria	48	3	3	3	2	0	2	1	1
Elgin	48	3	3	3	2	2	0	1	1
Champaign	37	3	2	2	2	1	1	0	1
Waukegan	37	3	2	2	2	1	1	1	0

and second most preferred trucks would be identical. Our preliminary investigation shows that if this happens, keeping only v that corresponds to the platooning possibility of the second position will achieve the maximum number of platoons while preserving stability. Further efforts may follow this line of thought to fully understand how the algorithmic approach would adapt to accommodate platooning with multiple routing options.

CRediT authorship contribution statement

Limon Barua: Conceptualization, Data curation, Methodology, Investigation, Visualization, Formal analysis, Validation, Writing - original draft, Writing - review & editing. **Bo Zou:** Conceptualization, Methodology, Investigation, Funding acquisition, Supervision, Formal analysis, Project administration, Resources, Writing - original draft, Writing - review & editing. **Pooria Choobchian:** Methodology, Validation, Writing - review & editing.

Declaration of Competing Interest

This research was funded in part by the US National Science Foundation (NSF) under Grant Number CMMI-2221418. The financial support of NSF is gratefully acknowledged. We would like to sincerely thank the anonymous referees for their constructive comments which helped us improve the presentation and content of the paper.

Appendix A:. Information about the northern Illinois road network

Table A1, Tables A2 and A3.

References

Abdolmaleki, M., Shahabi, M., Yin, Y., Masoud, N., 2021. Itinerary planning for cooperative truck platooning. Transp. Res. B Methodol. 153, 91–110.
Abeledo, H.G., Rothblum, U.G., 1994. Stable matchings and linear inequalities. Discret. Appl. Math. 54 (1), 1–27.
Federal Motor Carrier Safety Administration, 2022. A&I analysis & information online: custom report - registration data. U.S. Department of Transportation. Available at: https://ai.fmcsa.dot.gov/RegistrationStatistics/CustomReports#RegData (Retrieved on 03/07/2022).
Albiński, S., Crainic, T.G., Minner, S., 2020. The day-before truck platooning planning problem and the value of autonomous driving. Technical Report, CIRRELT.
Al-Qadi, I.L., Okte, E., Ramakrishnan, A., Zhou, Q., Sayeh, W., 2021. Truck-platoonable pavement sections in Illinois' network. Technical Report, Illinois Center for Transportation.
European Automobile Manufacturers Association (2017). EU roadmap for truck platooning. Brussels, Belgium. Available at: https://www.acea.auto/publication/eu-roadmap-for-truck-platooning/ (Retrieved on 02/16/2022).
Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., Tsugawa, S., 2012. Overview of platooning systems. In: Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria.

Bhoopalam, A.K., Agatz, N., Zuidwijk, R., 2018. Planning of truck platoons: A literature review and directions for future research. Transp. Res. B Methodol. 107, 212–228.
Bian, Z., Liu, X., 2019. Mechanism design for first-mile ridesharing based on personalized requirements part I: Theoretical analysis in generalized scenarios. Transp. Res. B Methodol. 120, 147–171.
Bishop, R., Bevly, D., Humphreys, L., Boyd, S., Murray, D., 2017. Evaluation and testing of driver-assistive truck platooning: phase 2 final results. Transp. Res. Rec. 2615 (1), 11–18.
Bonnet, C., Fritz, H., 2000. Fuel consumption reduction in a platoon: experimental results with two electronically coupled trucks at close spacing. SAE Technical Paper, 2000-01-3056, https://doi.org/10.4271/2000-01-3056.
Bouchery, Y., Hezarkhani, B., Stauffer, G., 2022. Coalition formation and cost sharing for truck platooning. Transp. Res. B Methodol. 165, 15–34.
Boysen, N., Briskorn, D., Schwerdfeger, S., 2018. The identical-path truck platooning problem. Transp. Res. B Methodol. 109, 26–39.
Browand, F., McArthur, J., Radovich, C., 2004. Fuel saving achieved in the field test of two tandem trucks. Technical Report UCB-ITSPRR-2004-20. California PATH, Institute of Transportation Studies, University of California, Berkeley.
Chen, S., Wang, H., Meng, Q., 2021. Autonomous truck scheduling for container transshipment between two seaport terminals considering platooning and speed optimization. Transp. Res. B Methodol. 154, 289–315.
Crane, C., Bridge, D. J., & Bishop, R. (2018). Driver assistive truck platooning: considerations for Florida state agencies. Technical Report. Florida Department of Transportation. Available at: https://www.fdot.gov/docs/default-source/legislative/documents/datp.pdf (Retrieved on 02/13/2022).
Echenique, F., Pereyra, J.S., 2016. Strategic complementarities and unraveling in matching markets. Theor. Econ. 11 (1), 1–39.
FAF (2019). Freight analysis framework: version 4.5. Available at: https://faf.ornl.gov/fafweb/ (Retrieved on 05/24/2022).
Feder, T., 1992. A new fixed point approach for stable networks and stable marriages. J. Comput. Syst. Sci. 45 (2), 233–284.
Hajdukova, J., 2006. Coalition formation games: A survey. International Game Theory Review 8 (04), 613–641.
Hassan, H., Dessouky, S., Talebpour, A., Rahim, M.A., 2020. Investigating the Impacts of Truck Platooning on Transportation Infrastructure in the South-Central Region. Transportation Consortium of South-Central States, Available at: https://digitalcommons.lsu.edu/cgi/viewcontent.cgi?article=1084&context=transet_pubs.
Hochschule Fresenius, DB Schenker and MAN Truck & Bus SE (2019), "EDDI: electronic drawbar-digital innovation", Technical Report. Available at: https://www.deutschebahn.com/resource/blob/4136372/d08df4c3b97b7f8794f91e47e86b71a3/Platooning_EDDI_Project-report_10052019-data.pdf (Retrieved 03/16/2022).
Irving, R.W., 1985. An efficient algorithm for the "stable roommates" problem. J. Algorithms 6 (4), 577–595.
Janssen, G.R., Zwijnenberg, J., Blankers, I.J., de Kruijff, J.S., 2015. Truck platooning: Driving the future of transportation. Technical Report, TNO whitepaper.
Johansson, A., Mårtensson, J., Sun, X., Yin, Y., 2021. Real-time cross-fleet pareto-improving truck platoon coordination. In: In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 996–1003.
Kamranian, Z., 2019. Load evaluation of the hay river bridge under different platoons of connected trucks. University of Calgary, Calgary. Master's Thesis,.
Knuth, D.E., Morris Jr, J.H., Pratt, V.R., 1977. Fast pattern matching in strings. SIAM J. Comput. 6 (2), 323–350.
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., Munson, T., & Sokolov, V. (2016). Coordinated platoon routing in a metropolitan network. In 2016 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing, pp. 73-82. Society for Industrial and Applied Mathematics.
Larsson, E., Sennton, G., Larson, J., 2015. The vehicle platooning problem: computational complexity and heuristics. Transportation Research Part c: Emerging Technologies 60, 258–277.
Simon Law (2019). Platooning-legislation-50-state-survey. Available at: https://simonlawpc.com/wp-content/uploads/2019/12/platooning-legislation-50-state-survey.pdf (Retrieved on 04/19/2022).
Liang, K.Y., Mårtensson, J., Johansson, K.H., 2014. In: Fuel-Saving Potentials of Platooning Evaluated through Sparse Heavy-Duty Vehicle Position Data. IEEE, pp. 1061–1068.
Liang, R., Wang, J., Huang, M., Jiang, Z.Z., 2020. Truthful auctions for e-market logistics services procurement with quantity discounts. Transp. Res. B Methodol. 133, 165–180.
Ling, T., Cao, R., Deng, L., He, W., Wu, X., Zhong, W., 2022. Dynamic impact of automated truck platooning on highway bridges. Eng. Struct. 262, 114326.
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.
Luo, F., Larson, J., Munson, T., 2018. Coordinated platooning with multiple speeds. Transportation Research Part c: Emerging Technologies 90, 213–225.
Maryland Department of Transportation (2021). Platooning in Maryland flyer. Available at: https://mva.maryland.gov/Documents/Platooning-in-Maryland-Flyer.pdf (Retrieved on 05/16/2022).
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. SAE Technical Paper. https://doi.org/10.4271/2018-01-1181.
North American Council for Freight Efficiency (2016). Truck platooning confidence report: two-truck platooning, Technical Report, pp 1-72. Available at: https://nacfe.org/wp-content/uploads/2018/02/TE-Platooning-CR-FINAL-_0.pdf (Retrieved on 03/07/2022).
Noruzoliaee, M., Zou, B., Zhou, Y.J., 2021. Truck platooning in the US national road network: A system-level modeling approach. Transportation Research Part e: Logistics and Transportation Review 145, 102200.
Nourmohammadzadeh, A., Hartmann, S., 2016. The fuel-efficient platooning of heavy duty vehicles by mathematical programming and genetic algorithm. In: International Conference on Theory and Practice of Natural Computing. Springer, Cham, pp. 46–57.
Peloton (2019). Peloton technology: Overview on driver-assistive truck platooning. In 2019 Illinois Transportation & Highway Engineering Conference. Available at: https://www.theconf.com/files/2021/06/Truck-Platooning-and-Automation.pdf (Retrieved on 04/01/2022).
Peloton (2020a). 2020 peloton technology. Available at: https://peloton-tech.com/ (Retrieved on 05/03/2022).
Peloton (2020b). Network operations center. Available at: https://peloton-tech.com/?albdesign_popup_cpt=network-operations-center (Retrieved on 04/05/2023).
Peloton (2021). The platooning experience. Available at: http://peloton-tech.com/how-it-works/ (Retrieved on 07/19/2022).
Schoettle, B., Sivak, M., Tunnell, M., 2016. A survey of fuel economy and fuel usage by heavy-duty truck fleets. Retrieved on 06/19/2022 Technical Report. the University of Michigan, Sustainable Worldwide Transportation. Available at. https://truckingresearch.org/wp-content/uploads/2016/10/2016.ATRI-UMTRI.FuelEconomyReport.Final_.pdf.
Scholl, J., Boysen, N., Scholl, A., 2023. E-platooning: optimizing platoon formation for long-haul transportation with electric commercial vehicles. Eur. J. Oper. Res. 304 (2), 525–542.
Smartt, B (2020). The Peloton Cloud: Key to Platooning Safety, Efficiency & Control. Available online at: http://peloton-tech.com/the-peloton-cloud-key-to-platooning-safety-efficiency-control/ (Reterived on 03/03/2023).
Sokolov, V., Larson, J., Munson, T., Auld, J., Karbowski, D., 2017. Maximization of platoon formation through centralized routing and departure time coordination. Transp. Res. Rec. 2667 (1), 10–16.
Sun, X., Yin, Y., 2019. Behaviorally stable vehicle platooning for energy savings. Transportation Research Part c: Emerging Technologies 99, 37–52.
Sun, X., Yin, Y., 2021a. Decentralized game-theoretical approaches for behaviorally-stable and efficient vehicle platooning. Transp. Res. B Methodol. 153, 45–69.
Sun, X., Yin, Y., 2021b. An auction mechanism for platoon leader determination in single-brand cooperative vehicle platooning. Econ. Transp. 28, 100233.
Sun, X., Wu, H., Abdolmaleki, M., Yin, Y., Zou, B., 2021. Investigating the potential of truck platooning for energy savings: empirical study of the US national highway freight network. Transp. Res. Rec. 2675 (12), 784–796.
Tan, J.J., 1990. A maximum stable matching for the roommates problem. BIT Numer. Math. 30 (4), 631–640.

Technavio (2018). Top companies in the global truck platooning systems market. Available at: https://blog.technavio.com/blog/top-companies-global-truck-platooning-systems-market (Retrieved on 04/01/2022).
Teo, C.P., Sethuraman, J., 2000. On a cutting plane heuristic for the stable roommates problem and its applications. Eur. J. Oper. Res. 123 (1), 195–205.
Trimble Transportation (2023). It takes a platform to platoon. Available at: https://transportation.trimble.com/resources/blogs/it-takes-a-platform-to-platoon (Retrieved on 04/05/2023).
Tsugawa, S., Jeschke, S., Shladover, S.E., 2016. A review of truck platooning projects for energy savings. IEEE Trans. Intell. Veh. 1 (1), 68–77.
van de Hoef, S., 2016. Fuel-efficient centralized coordination of truck platooning. KTH Royal Institute of Technology. Doctoral dissertation.
Wang, M., van Maarseveen, S., Happee, R., Tool, O., van Arem, B., 2019. Benefits and risks of truck platooning on freeway operations near entrance ramp. Transp. Res. Rec. 2673 (8), 588–602.
World population review, 2021. Available at: https://worldpopulationreview.com/states/cities/illinois (Retrieved on 06/13/2022).
Xu, W., Cui, T., Chen, M., 2023. Optimizing two-truck platooning with deadlines. IEEE Trans. Intell. Transp. Syst. 24 (1), 694–705.