Decentralized coordination for truck platooning

Zeng, Yikai; Wang, Meng; Rajan, Raj Thilak

DOI
10.1111/mice.12899

Publication date
2022

Document Version
Final published version

Published in
Computer-Aided Civil and Infrastructure Engineering

DOI: 10.1111/mice.12899

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Decentralized coordination for truck platooning

Yikai Zeng[1] | **Meng Wang**[2] | **Raj Thilak Rajan**[1]

[1]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands

[2]Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, Technische Universitaẗ Dresden 01069, Dresden, Germany

Correspondence
Meng Wang, Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, Technische Universität Dresden
Email: meng.wang@tu-dresden.de

Present address
Meng Wang, Stevinweg 1, 2628 CN Delft, The Netherlands.

Abstract

Coordination for truck platooning refers to the active formation of a group of heavy-duty vehicles traveling at close spacing to reduce the overall truck operations costs. Conventionally, this coordination is achieved by optimizing various truck-related parameters, such as schedules, velocities, and routes, based on an objective function that minimizes a certain cost, for example, fuel usage. However, prevalent algorithms for the coordination problem are typically integer-constrained, which are not only hard to solve but are not readily scalable to increasing fleet sizes and networks. In this paper, to overcome these limitations, we propose a centralized formulation to optimize the truck parameters and solve a multidimensional objective cost function including fuel, operation time costs and preferential penalty. Furthermore, to improve the scalability of our proposed approach, we propose a decentralized algorithm for the platoon coordination problem involving multiple fleets and objectives. We perform both theoretical and numerical studies to evaluate the performance of our decentralized algorithm against the centralized solution. Our analysis indicates that the computation time of the proposed decentralized algorithms is invariant to the increasing fleet size, at the cost of a small relative gap to the optimum cost given by the centralized method. We discuss these results and present future directions for research.

1 | INTRODUCTION

Heavy-duty vehicles (HDVs) emit around 5% of the total carbon emissions in the world; therefore, there is an ever-increasing demand for enhancing the fuel efficiency of HDVs (Schroten et al., 2012). In addition to ecological effects, one-third of the overall operation cost of an HDV is fuel consumption (Schittler, 2003). Therefore, it is beneficial both environmentally and economically to develop systems to increase the fuel efficiency in HDVs operations. Truck platooning is one of such systems that has received substantial attention due to its promising benefits and business model (Larson et al., 2013; Varaiya, 1993).

Truck platooning refers to controlling a group of HDVs at a close intervehicle distance (Varaiya, 1993). In a platoon, the overall air drag is decreased, thus improving the average fuel efficiency of the platoon. There are several approaches for forming platoons. If there are a sufficient number of trucks traveling in the network, it may be possible to form a platoon by encountering peer trucks, known as the opportunistic platooning (Bhoopalam et al., 2018). However, given the low percentage of truck traffic in the overall traffic composition and the gradual introduction of truck automation and intelligent transportation systems (Jiang & Adeli, 2004a, 2004b; Wei et al., 2022), the probability of forming a platoon in an ad hoc way is very low in

practice. Therefore, it is necessary to coordinate different trucks to actively form more platoons (Larson et al., 2013).

The goal of the truck platooning coordination problem is to maximize the platooning benefits by adjusting the trucks' itinerary plans. Coordination is generally achieved by optimizing the schedule, velocity, and routes to increase the platooning chances. However, the mathematical program built for the coordination problem is typically integer-constrained, making it generally hard to solve for and scale to large fleet sizes and networks. This paper deals with computational algorithms for truck platooning coordination. To justify the contribution of the paper, we begin with a review of the current methods and state-of-the-art computational algorithms on truck platooning coordination in the next section.

2 | LITERATURE REVIEW: TRUCK PLATOONING COORDINATION

Truck platooning coordination can be classified into various categories. In this paper, we mainly consider cooperative versus noncooperative and centralized versus decentralized in this study.

2.1 | Cooperative versus noncooperative

In a cooperative truck platooning coordination scenario, all the trucks seek to minimize a global cost function. For example, it is common that in one logistics company, cooperative trucks share a set of delivery assignments. The trucks are cooperative as they belong to the same owner and are thus interested in optimizing the overall cost of the fleet. It is therefore acceptable for individual trucks to travel in a plan with a higher individual cost as long as the global fleet cost is optimized. In contrast, noncooperative truck platooning participants are concerned only with individual cost functions. In this scenario, an individual truck owner is unwilling to accept a plan that increases his or her cost, albeit the overall cost reduction of all trucks.

2.2 | Centralized versus decentralized

In a centralized scenario, a centralized node gathers demands for each truck and solves the coordination problem. For example, in a logistics company, trucks' travel plans are assigned by the dispatch center. Alternatively, a decentralized setup usually refers to the topology where the agents in the network send information to a fusion node. However, unlike the centralized node, the fusion node does not make decisions for each agent but serves the purpose such as relaying messages and broadcasting key parameters after computing based on inputs from multiple agents.

An overview of the truck platooning coordination literature is given in Table 1. We focus our review on the computational algorithms for truck platooning and for a more comprehensive survey of truck platooning coordination, we refer readers to Bhoopalam et al. (2018).

2.3 | Cooperative centralized coordination

Cooperative centralized platooning planning was first proposed by Larson et al. (2013), in which a fast heuristic was built to solve the problem for large fleets. The research suggests that the fuel-saving potential increases as the number of trucks increases in the road network. They also find that trucks are more likely to deviate from their individually optimized routes if detouring only leads to insignificant travel distance changes. The existence of such favorable alternative routes depends on the topology of the road network.

Larson et al. (2014) proposed a method to handle the large complexity of the platooning coordination algorithm by setting local control nodes throughout the graph. However, since the decisions are made at these control nodes, the method is still centralized by the classification method of this paper. This was one of the first attempts to reduce complexity by splitting the graph into different parts, guiding vehicles to gather in certain local centers to form platoons.

An extended study on the computational complexity in coordination heuristics in the truck platooning problem was carried out (Larsson et al., 2015). It was proven that the platooning coordination problem is in general nondeterministic polynomial-time NP-hard, even if all trucks share the same starting point and departure time. Both the same starting-point problem and general different starting-point problem are formulated as integer programs. This study proposed a solver for an exact solution; however, the computation time for the solver was infeasible for realistic networks and fleet sizes. Hence, three heuristic solvers were proposed, which offer a suboptimal solution but achieve the solution in a more practical time.

An alternative approach for approximating the optimum to solve the problem under a fuel-consumption-based objective function with a genetic algorithm is proposed in Nourmohammadzadeh & Hartmann (2016). The simulation is based on a simplified German intercity network with 10–50 trucks. The work incorporates features of the latest arrival time restriction and travel speed options (Nourmohammadzadeh & Hartmann, 2018). The model is

TABLE 1 Truck platooning coordination review

Authors (year)	Scenarios	Topology	Objective		Decision		
			Fuel	Others	Schedule	Routes	Speed
Larson et al. (2013)	Cooperative	Centralized	✓		✓	✓	✓
Larson et al. (2014)	Cooperative	Centralized	✓		✓		✓
Larsson et al. (2015)	Cooperative	Centralized	✓		✓	✓	
Nourmohammadzadeh and Hartmann (2016)	Cooperative	Centralized	✓		✓	✓	
Zhang et al. (2017)	Cooperative	Centralized	✓	✓	✓		
Sokolov et al. (2017)	Cooperative	Centralized	✓		✓		
Nourmohammadzadeh and Hartmann (2018)	Cooperative	Centralized	✓		✓	✓	✓
Boysen et al. (2018)	Cooperative	Centralized	✓	✓	✓		
Johansson et al. (2018)	Noncooperative	Centralized	✓	✓	✓		
Johansson and Mårtensson (2019)	Noncooperative	Centralized	✓	✓	✓		
Abdolmaleki et al. (2021)	Cooperative	Centralized	✓		✓	✓	✓
This study	Cooperative	Centralized	✓	✓	✓	✓	✓

Note: The column "objective" describes the composition of the cost function. Most studies focus only on fuel, while other aspects involve operation time cost and preference penalty, decisions for the optimization problem, including schedule, routes, and speed. The schedule parameters include the departure time and temporary parking point for each truck.

solved using particle swarm optimization due to the high computational complexity.

Recently, researchers formulated the platooning coordination problem as a mixed integer nonlinear program problem (Abdolmaleki et al., 2021), where the decision variables included the itinerary and the velocity of the individual trucks. This research characterized the cost function as an energy-based piecewise concave function and presented an exact solution. Furthermore, a fast dynamic-programming heuristic was applied to real-world test cases with significant improvement in computational time as compared with previous studies.

With practical approximated heuristics, researchers have performed a comparison study between scheduled planning and real-time opportunistic platooning to explore the potential in cooperative platooning planning (Sokolov et al., 2017). An unweighted Hanna grid in the POLARIS platform was used as a benchmark for comparison with a relatively large-scale test set (Auld et al., 2016). Furthermore, using a specific road network with only one path, the profitability of truck platooning was investigated in Boysen et al. (2018), which suggested that with an inefficient process to form a platoon, the benefits of fuel saving are limited.

All the aforementioned studies in this category assume that all trucks are identical and share a common global cost function, that is, they are cooperative. A centralized service provider is also necessary, which must handle a complex and computationally expensive centralized optimization problem.

2.4 | Noncooperative centralized coordination

Taking the competitive nature of the transportation industry into account, a noncooperative case where all trucks start at the same point with different preferred departure times and are coordinated by a centralized service provider has been studied (Johansson et al., 2018). Johansson et al. (2018) prove that when trucks are identical and noncooperative, the coordination problem is a congestion game, a classic problem in which a pure strategic Nash equilibrium can be searched by best response dynamics. The numerical simulation suggests that in a noncooperative setup, the overall benefits are less, compared to a cooperative case, showing that the self-interested property does not fully utilize the potential in truck platooning, although still improving the fuel efficiency, compared to nonplatooning scenarios (Johansson et al., 2018).

Noncooperative trucks are interested in knowing their exact position in a platoon since no air-drag reduction is available for the leading truck. The mechanisms of forming a platoon impact the profitability of each truck. A comparative study on the interaction mechanisms between leading and tailing trucks is carried out in Johansson and Mårtensson (2019), which suggests that the overall fuel savings are subject to the leading truck selection mechanism.

The models in the aforementioned studies assume all trucks start at the same node and the only adjustable decision is the departure time (Johansson & Mårtensson, 2019; Johansson et al., 2018), which may not be practically

TABLE 2 A review of state-of-the-art solutions for truck platoon coordination and their simulation setup

Author (year)	Solution	Network	Max fleet size under Exact solution	Max fleet size under Approximate solution
Larson et al. (2013)	Exact and Local search heuristic	German autobahn network	–	8000
Larsson et al. (2015)	Exact and Local search heuristic	German autobahn network	10	200
Nourmohammadzadeh and Hartmann (2016)	Exact and genetic algorithm	Simplified German intercity map	20	50
Sokolov et al. (2017)	Exact	10×10 grid	50	–
Zhang et al. (2017)	Exact	Identical path	2	–
Nourmohammadzadeh and Hartmann (2018)	Exact and particle swarm optimization	Chicago road network	–	1000
This study	Exact and decentralized heuristic	3×3, 4×4, 6×6, and 10×10 grid	50	50

feasible. Moreover, trucks belonging to the same company may compete with each other under these noncooperative assumptions, resulting in a suboptimal result for the company. Last but not the least, since trucks are noncooperative, a centralized node is required for coordination. A trustworthy service provider must exist to enable noncooperative centralized coordination, which may require extra service costs with the risk of losing privacy.

2.5 | Knowledge gap and proposed contribution

The existing literature has limitations in three major areas.

1. Restrictive assumptions, objective function, and decision variables: In previous studies, trucks are typically assumed to be identical for a favorable structure of the formulation, which may not be practical for a large fleet. Most studies focus only on fuel cost, which is not sufficiently representative for multidimensional interests of logistic operations. Furthermore, most studies have focused on departure time coordination, but few consider truck speed as a decision variable.
2. Infeasible exact solution method and lack of deterministic performance guarantees: With integer variables, the previously proposed models are generally computationally expensive to solve. Researchers have attempted to reduce the computational time using various approximation and stochastic methods. However, most of them are unable to offer deterministic performance guarantees of their solutions. An overview of these issues is summarized in Table 2.
3. Decision making limited to the centralized controller: Automated trucks today are equipped with considerable computational capability. To the best of our knowledge, there are no existing studies utilizing this resource in the coordination problem.

To address the limitations mentioned above, this paper develops a decentralized computational model for the multicriteria heterogeneous truck platooning coordination problem. The main contribution is as follows: A decentralized algorithm for solving the multifleet truck platooning coordination problem.

We also propose a mathematical model for cooperative truck platooning, in which our objective cost function consists of not only fuel usage but also travel time and schedule preference, and the truck routes, speed, and departure time are decision variables to be optimized. The proposed decentralized algorithm is compared to a centralized algorithm, and its performance is evaluated with varying problem sizes.

The rest of the paper is organized in the following structure. In Section 3, a mathematical model of the coordination problem is built with a generic cost function and the compatibility of having various types of trucks in the fleet of interest. Section 4 introduces the relaxation methods to reformulate the model, followed by numerical examples for the performance analysis. Section 5 introduces a decentralized method for solving the model and demonstrates the result of a numerical simulation for comparison with the centralized solver. Section 6 discusses the conclusion and future work. The general symbols and operators in this paper are explained in Table 3.

TABLE 3 Mathematical symbols and operators

a, A	Upper or lower case in plain text denotes scalars
$a(\cdot)$	Functions
$\mathbf{A}$	Bold symbol upper case denotes matrices
$\mathbf{a}$	Bold symbol lower case denotes column vectors
$\mathcal{A}$	Calligraphic letters denote a graph or a set
$\boldsymbol{\mathcal{A}}$	Bold-symbol calligraphic letters denote an optimization problem
$\mathbb{R}$	Blackboard style denotes the set of real number
$\mathbb{R}^{m\times n}$	Real-valued matrix with m rows and n columns
$\mathbb{R}^{n}$	n-Dimensional real-valued one-column vectors
$\mathbb{Z}^{n}$	n-Dimensional integer-valued one-column vectors
$[0,1]^{n}$	Set for n-dimensional one-column vectors with elements being 0 or 1
$1_n, 0_n$	All one or zero one-column vectors of length n
$\mathbf{I}_{n\times n}$	Identity matrices of size$n \times n$
$(\cdot)^{T}$	Transpose
$\times$	Cartesian product for nonscalars
$\leq, \geq$	Components-wise inequalities for nonscalars

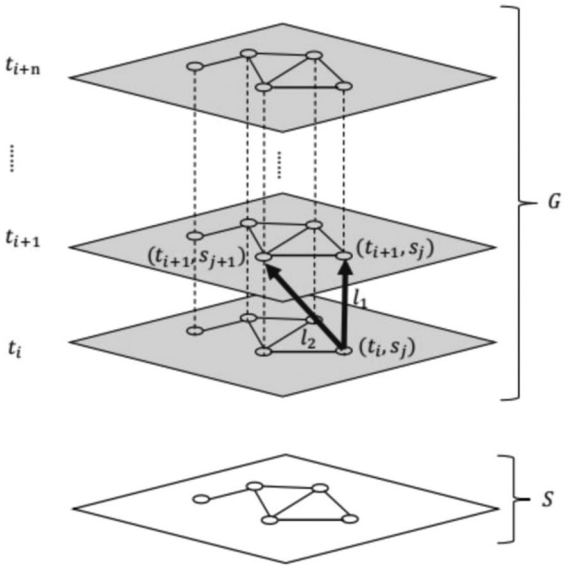

FIGURE 1 The figure shows the spatiotemporal representation of the road network, where $\mathcal{S}$ shows the physical nodes and $\mathcal{G}$ shows a collection of all possible graphs. Here, l_1 represents a vehicle that waits at s_i from time t_i to t_{i+1}. l_2 represents leaving s_j at t_i and arriving at s_{j+1} at t_{i+1}

3 | PLATOON COORDINATION PROBLEM FORMULATION

In this section, we formulate the truck platooning coordination problem. A discrete spatiotemporal road network is constructed, enabling the representation of truck traveling plans with binary decision vectors.

3.1 | Spatiotemporal road network

The network model, also known as STEN, is frequently seen in transportation network modeling as introduced by Tong et al. (2015). We adopt the notation for road network from Abdolmaleki et al. (2021), with minor modifications. The physical road network is given with $\mathcal{S}$ as the set of all physical nodes, where $s_i \in \mathcal{S}$ is the ith ordered physical node. A physical node may be a parking lot, a city, or a point on the road. The study time period is divided into a sequence of successive intervals, which are represented by the set $\mathcal{T}$, and we denote $t_i \in \mathcal{T}$ as the ith ordered time interval. The spatiotemporal road network $\mathcal{G}$ is constructed with nodes defined as (t_i, s_i), which represents a truck that may be at s_i at time interval t_i. The set of all nodes in $\mathcal{G}$ is defined as $(\mathcal{G})$, and the set of all links in $\mathcal{G}$ is denoted by $\mathcal{E}(\mathcal{G})$. An example of $\mathcal{G}$ is shown in Figure 1.

A link $l \in \mathcal{E}(\mathcal{G})$ connects two nodes; hence, $l := (t_i, s_i, t_j, s_j)$ means a truck leaves s_i at time t_i to arrive s_j at time t_j if either there is a physical link between s_i and s_j or $s_i = s_j$. Due to the physical limitation that a truck cannot travel back in time, t_i is always smaller than t_j, that is, $\mathcal{G}$ is a directed graph. The total number of links in $\mathcal{E}(\mathcal{G})$ is denoted by L. All the information about $\mathcal{G}$ is assumed to be known by all trucks.

3.2 | Individual truck path and fleet plan

We denote $\mathcal{K}$ as the set of trucks planning to operate on $\mathcal{G}$ and K as the total number of trucks in $\mathcal{K}$. A binary decision variable is denoted by $x_{k,l}$, which indicates whether truck k transverses on route l as

$$x_{k,l} = \begin{cases} 1 & \text{Truck } k \text{ traverses through link } l \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We stack all decision variables of truck k in the order of links into a vector $\bar{\mathbf{x}}_k \in \mathbb{R}^L$, which represents the path of truck k,

$$\bar{\mathbf{x}}_k = \left[x_{k,1}, x_{k,2}, \ldots, x_{k,L}\right]^T \qquad \forall l \in \mathcal{E}(\mathcal{G}) \tag{2}$$

Each truck $k \in \mathcal{K}$ has a local trip schedule specifying the origin $o_k \in \mathcal{S}$, the destination $d_k \in \mathcal{S}$, the earliest departure time $t_k^{ED} \in \mathcal{T}$, the preferred arrival time $t_k^{PA} \in \mathcal{T}$ and the latest arrival time $t_k^{LA} \in \mathcal{T}$.

Clearly, with physical limitations, a considerable number of links in $\mathcal{G}$ are infeasible for truck k to transverse

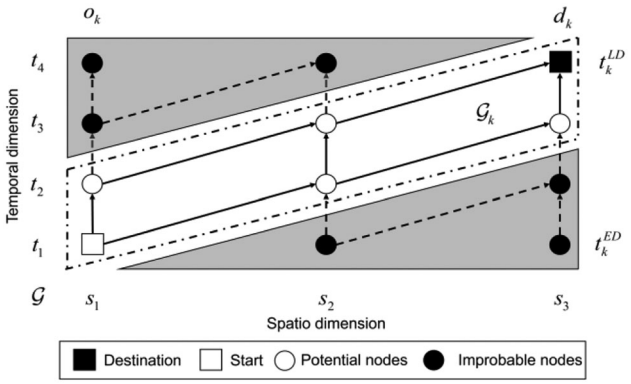

FIGURE 2 The example graph is a spatiotemporal extension of a one-dimensional geographical map from $\boldsymbol{s}_1$ to $\boldsymbol{s}_3$, which is on the horizontal axis. The vertical axis represents the time dimensions at each interval

within operating time limitations and limited speed options. For the sake of computational efficiency, a refining process is applied to $\mathcal{G}$ proposed by Masoud and Jayakrishnan (2017). Let $\mathcal{G}_k$ be the subgraph of $\mathcal{G}$ after the refining process, in which two kinds of nodes and their connected links are removed as shown in the simple example given in Figure 2. Nodes in the lower right gray area are infeasible because a truck cannot arrive at the destination within physical speed limits before t_k^{LA} once it travels to one of these nodes. Second, nodes in the upper left gray area are too far from the origin and are not reachable due to departure time t_k^{ED} and speed constraints.

Let L_k be the number of links in $\mathcal{G}_k$. The decision vector is refined as

$$\mathbf{x}_k = \left[x_{k,1}, x_{k,2}, \ldots, x_{k,L}\right]^T \qquad \forall l \in \mathcal{E}\left(\mathcal{G}_k\right) \tag{3}$$

A fleet is a subset of trucks. For the cth fleet, $\mathcal{K}_c$ is a set of K_c trucks and belongs to fleet c, where $\mathcal{K}_c \subseteq \mathcal{K}$. Consequently, let $\mathbf{a}_c$ denote the path of all trucks in fleet c

$$\mathbf{a}_c = \left[\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_{K_c}^T\right]^T \qquad \forall k \in \mathcal{K}_c \tag{4}$$

The spatiotemporal nature of graph $\mathcal{G}$ entails that the decision vector captures departure time, route, and link speed adjustments.

3.3 | Cost function

In this subsection, the cost of a truck is formulated as a function of the travel plan. Inspired by Zhang et al. (2017), we consider three terms in the cost function, namely, (a) fuel cost, (b) travel time cost, and (c) the schedule preference penalty. However, we propose a more general model with a tunable cost for each truck in the fleet.

3.4 | Fuel cost

Truck platooning saves fuel for the following trucks but not the platoon leader. A mechanism to balance the savings is assumed to be applied, such that all trucks in the platoon are assumed to have the same average air drag reduction factor during the platooning trip. Let φ_l denote the average air drag reduction factor as a function of the number of trucks N_l on link $l \in \mathcal{E}(\mathcal{G})$, which is given by

$$\varphi_l\ (N_l) = \varphi_o\ - \frac{\varphi_o}{N_l} \tag{5}$$

where φ_o is the air reduction factor for all platooning trucks and is assumed to be constant and equal for all trucks during the platooning.

To model the fuel cost on link $l = (t_i, s_i, t_j, s_j)$, let L_l be the distance between physical nodes s_i and s_j and $T_l = t_j - t_i$ denote the travel time from s_i to s_j. The fuel cost $\tilde{f}_{k,l}$ on link l for truck k is given by (6), which is composed of the fuel consumption, air drag, and other friction (Franceschetti et al., 2013).

$$\tilde{f}_{k,l}(\varphi_l) = \frac{w_f \xi}{\kappa}\left(\mu_k E_k V_k T_l + 0.5\frac{c_{d,k}\rho A_k(1-\varphi_l)}{\varepsilon_k \varpi_k}\frac{L_l^3}{T_l^2} + \frac{M_k g(\sin\theta_l + c_{r,k}\cos\theta_l)}{\varepsilon_k \varpi_k} L_l\right) \tag{6}$$

with the physical constants defined as w_f, price of oil (€/kg); ξ, fuel-to-mass ratio; κ, heating value of the fuel (J/kg); ρ, density of air (kg/m^3); g, gravitational constant (m/s^2); and vehicle-dependent $(\cdot)_k$ or link-dependent $(\cdot)_l$ parameters are defined as θ_l, road gradient (°); E_k, engine speed (rev/s); L_l, distance between the physical nodes (m); T_l, engine working time spent on the link (s); μ_k, engine friction factor; V_k, engine displacement (m^3); $c_{d,k}$, coefficient of aerodynamic drag; A_k, front area of the truck (m^2); $c_{r,k}$, coefficient of rolling resistance; M_k, total vehicle weight (kg); ε_k, drive train efficiency; ϖ_k, an efficiency parameter for the engine.

The only unknown variable in (6) is φ_l for a given link l and truck k; therefore, (6) may be rewritten as

$$\tilde{f}_{\mathrm{k,l}}\ (\varphi_l) = C_{k,l,f_l}\,\varphi_l + C_{k,l,f_2} \tag{7}$$

where

$$C_{k,l,f_1} = -0.5\frac{\xi}{\kappa}\frac{c_{d,k}\rho A_k L_l^3}{\varepsilon_k \varpi_k T_l^2} \tag{8}$$

$$C_{k,l,f_2} = \frac{\xi}{\kappa}\left(\mu_k E_k V_k T_l + 0.5\frac{c_{d,k}\rho A_k}{\varepsilon_k \varpi_k}\frac{L_l^3}{T_l^2} + \frac{M_k g(\sin\theta_l + c_{r,k}\cos\theta_l)}{\varepsilon_k \varpi_k} L_l\right) \quad (9)$$

Notably, in the case that for $l = (t_i, s_i, t_j, s_j)$, where $s_i = s_j$, the above equation indicates that the vehicle is waiting at a certain physical node. We assume that the vehicle is shut down and that there is no fuel cost. Substituting (5) into (7), we have

$$\tilde{f}_{k,l}(\varphi_l) = \check{f}_{k,l}(N_l) = \begin{cases} \frac{a_{k,l}}{N_l} + b_{k,l} & N_l \geq 1 \\ 0 & N_l = 0 \end{cases} \quad (10)$$

where N_l is the number of trucks traversing on l. In (10), we convert the function φ_l to the cost $\tilde{f}_{k,l}$ to a function from N_l to the cost, $\check{f}_{k,l}(N_l)$.

$$a_{k,l} = \varphi_o C_{k,l,f_1} \quad (11)$$

$$b_{k,l} = C_{k,l,f_2} - \varphi_o C_{k,l,f_1} \quad (12)$$

Let us denote $\eta_l(\cdot) : \mathbf{Z}^{L_c} \to \mathbb{R}$ for every link l, which is a linear transformation from the overall planning to the number of vehicles on a certain link:

$$N_l = \eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) \quad (13)$$

where $\mathbf{a}_c$ is the decision vector for fleet c, while $\mathbf{a}_{-c}$ is the decision vector for all fleets other than c. Finally, the fuel cost of truck k on link l is given by

$$f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \begin{cases} \frac{a_{k,l}}{\eta_l(\mathbf{a}_c, \mathbf{a}_{-c})} + b_{k,l} & \eta_l(\mathbf{a}_c) \geq 1 \\ 0 & \eta_l(\mathbf{a}_c) = 0 \end{cases} \quad (14)$$

3.5 | Travel time cost

The real-world stakeholders of freight transportation are concerned with various multidimensional costs in addition to the fuel cost (Significance Quantitative Research, V. U. University Amsterdam & John Bates Services, 2013). For example, the process of forming platoons may increase time-related costs, including travel time cost and schedule preference penalty. Furthermore, the value of cargoes could also impact the travel time cost, which is assumed to be identical in all trucks and hence discarded in the formulation. The *travel time cost* is given by

$$\tilde{g}_k(\mathbf{x}_k) = w_{t,k}\mathbf{1}_{L_k}^T \mathbf{x}_k \quad (15)$$

where $w_{t,k}$ is a weight for the travel time cost for the truck. The definition of $\mathbf{x}_k$ makes each nonzero element in $\mathbf{x}_k$ represent a certain time duration in the truck's operation. Thus, by adding all elements in $\mathbf{x}_k$ and assigning a proper weight $w_{t,k}$, (15) delivers the value of the time cost of truck k.

3.6 | Schedule preference penalty

The *schedule preference penalty* represents the preferred arrival time, which for truck k is given by

$$\tilde{h}_k(\mathbf{x}_k) = \max\left\{w_{k,l}\left(\mathbf{1}_{L_k}^T \mathbf{x}_k + t_k^{ED} - t_k^{PA}\right), - w_{k,e}(\mathbf{1}_{L_k}^T \mathbf{x}_k + t_k^{ED} - t_k^{PA})\right\} \quad (16)$$

where $w_{k,l,}$ positive weight for later than schedule preference penalty; $w_{k,e}$, positive weight for earlier than schedule preference penalty; t_k^{ED}, earliest departure time; t_k^{PA}, preferred arrival time.

Note that the term $\mathbf{1}_{L_k}^T \mathbf{x}_k + t_k^{ED} - t_k^{PA}$ calculates the errors between the real arrival time and preferred arrival time in seconds. The weights for early and late arrival may be different since the stakeholders may value them differently. The maximizing manipulation in (16) is to combine two parts of the schedule preference penalty into a uniform formulation.

The total cost function $F_c : \mathbf{z}^{L_c} \to \mathbb{R}$ is a function of $\mathbf{a}_c$; therefore, (15) and (16) require reformulation. Based on (4), a selection matrix $\mathbf{S}_k$ of the proper size can be chosen such that the following holds:

$$\mathbf{x}_k = \mathbf{S}_k \mathbf{a}_c \quad (17)$$

where $\mathbf{x}_k$ is the decision vector for truck k, and $\mathbf{a}_c$ is the decision vector of fleet c, defined in (3) and (4), respectively. To convert (15) into a function of $\mathbf{a}_c$, we substitute (17) into (15)

$$\tilde{g}_k(\mathbf{x}_k) = g_k(\mathbf{a}_c) = w_{t,k}\mathbf{1}_{L_k}^T \mathbf{S}_k \mathbf{a}_c \quad (18)$$

and likewise for (16), we have

$$\tilde{h}_k(\mathbf{x}_k) = h_k(\mathbf{a}_c) = \max\left\{w_{k,l}\left(\mathbf{1}_{L_k}^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA}\right), - w_{k,e}\left(\mathbf{1}_{L_k}^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA}\right)\right\} \quad (19)$$

The cost function of truck k is given by summing up all pieces on all links,

$$f_k(\mathbf{a}_c) = \sum_{l \in \mathcal{E}(\mathcal{G}_k)} f_{k,l}(\mathbf{a}_c) + g_k(\mathbf{a}_c) + h_k(\mathbf{a}_c) \quad (20)$$

Consequently, the cost for the fleet is given by summing the cost of all related trucks

$$F_c(\mathbf{a}_c) = \sum_{k\in\mathcal{K}_c}\left[\sum_{l\in\mathcal{E}(\mathcal{G}_k)} f_{k,l}(\mathbf{a}_c) + g_k(\mathbf{a}_c) + h_k(\mathbf{a}_c)\right] \quad (21)$$

where $\mathbf{a}_c$, the planning of fleet c defined in (4); $f_{k,l}(\cdot)$, fuel cost of truck k on link l defined in (14); $g_k(\cdot)$, time cost of truck k defined in (18); $h_k(\cdot)$, preference penalty of truck k defined in (19).

3.7 | Constraints

Each fleet attempts to minimize its own cost as described in (21) while limited by two sets of constraints. The first set of constraints guarantees that the solution is a feasible path in $\mathcal{G}$ and that the truck obeys the latest arrival requirement. The constraint is formulated based on the multicommodity network flow problem (Even et al., 1975). The flow conservation constraints are written as

$$\sum_{(t_i,s_i):l=(t_i,s_i,t,s)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} - \sum_{(t_j,s_j):l=(t,s,t_j,s_j)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} = d^k_{(t,s)} \quad (22)$$

where

$$d^k_{(t,s)} = \begin{cases} -1 & (t,s) = (t^{ED}_k, o_k) \\ 1 & (t,s) = (t^{LA}_k, d_k) \\ 0 & \text{otherwise} \end{cases}$$

and $x_{k,l}$ is the binary decision variable as described in (1). We convert the formulation into a matrix form to simplify the notation,

$$\mathbf{A}_k = \left[\mathbf{e}^T_{(t_1,s_1)}, \dots, \mathbf{e}^T_{(t_i,s_i)}\right]^T, \quad \forall (t,s)\in\mathcal{G}_k \quad (23)$$

where $\mathbf{e}_{(t,s)}$ is a vector of length L_k, which is the number of links in $\mathcal{G}_k$ defined as

$$\mathbf{e}^T_{(t,s)}\mathbf{x}_k = \sum_{(t_i,s_i):l=(t_i,s_i,t,s)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} - \sum_{(t_j,s_j):l=(t,s,t_j,s_j)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} \quad (24)$$

Meanwhile, let $\mathbf{b}_k$ be a vector of an appropriate size, where

$$\mathbf{b}_k = [d^k_{(t_i,s_i)}, \dots, d^k_{(t_i,s_i)}]^T, \quad \forall (t,s)\in\mathcal{G}_k \quad (25)$$

In this way, (22) is rewritten as

$$\mathbf{A}_k\mathbf{S}_k\mathbf{a}_c = \mathbf{b}_k \quad \forall k\in\mathcal{K}_c \quad (26)$$

where $\mathbf{S}_k$ is a selection matrix that satisfies (17). By assigning (t^{ED}_k, d_k) as the spatiotemporal destination, the path is guaranteed to terminate at the node before a certain time period.

The other constraint simply follows the definition of binary decision variable $x_{k,l}$, which is defined in (1). Combining the abovementioned constraints, the best-response subproblem, denoted as $\boldsymbol{\mathcal{P}}$, is given in

$$\begin{aligned} \min_{\mathbf{a}_c} \quad & F_c(\mathbf{a}_c, \mathbf{a}_{-c}) \\ \text{subject to} \quad & \mathbf{A}_k\mathbf{S}_k\mathbf{a}_c = \mathbf{b}_k \quad \forall k\in\mathcal{K}_c \\ & \mathbf{a}_c \in [0,1]^{L_c} \end{aligned} \quad (\boldsymbol{\mathcal{P}})$$

where $\mathbf{a}_{-c}$ is considered constant and $\mathbf{a}_c$ is defined in (4). $F_c(\mathbf{a}_c, \mathbf{a}_{-c})$ is defined in (21). $L_c = \sum_{k\in\mathcal{K}_c} L_k$ is the sum of the quantity of all links in all $\mathcal{G}_k$.

In the following sections, we aim to solve this optimization problem using centralized and decentralized solutions.

4 | CENTRALIZED SOLUTION

Observe that the cost function in $\boldsymbol{\mathcal{P}}$ lacks structure since $f_{k,l}$ is discontinuous, although the remaining part of this function is linear. In this section, we discuss the details of applying relaxations to preserve the linear form of the problem and confirm the problem to be solved as a standard mixed integer linear program (MILP) problem.

4.1 | Relaxation for the fuel cost function

In this section, we aim to convert the original cost function into an integer linear program. The reformulation begins with fitting (10) with piecewise linear segments, which we restate here for the sake of completion:

$$\check{f}_{k,l}(N_l) = \begin{cases} \frac{a_{k,l}}{N_l} + b_{k,l} & N_l \geq 1 \\ 0 & N_l = 0 \end{cases}$$

where $a_{k,l}$ and $b_{k,l}$ are constants depending on the truck and the link as described in (11) and (12). Here, N_l is a nonnegative integer by definition; thus, this method is not an approximation since the values are exact at feasible points as shown in Figure 3.

The function is fitted with several linear segments. The jth linear segments of total J pieces is denoted as $(\alpha_{k,l,j}N_l + \beta_{k,l,j})$ for truck k on link l, in which

$$\alpha_{k,l,j} = \frac{a_{k,l}}{j+1} - \frac{a_{k,l}}{j} \quad (27)$$

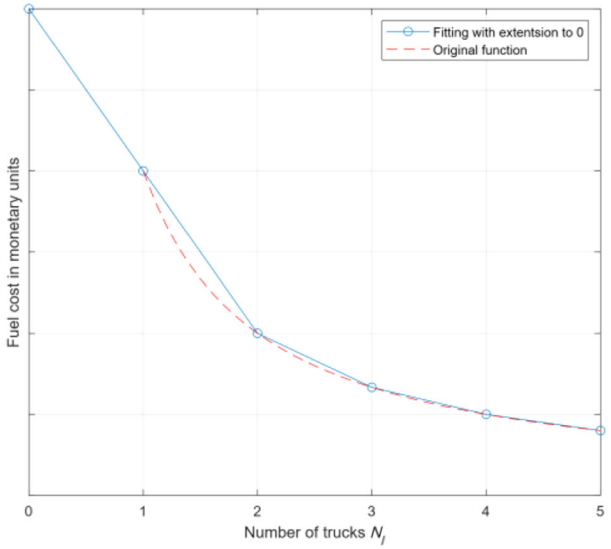

FIGURE 3 An example of applying piecewise fitting to $\check{f}(N_l)$

$$\beta_{k,l,j} = \frac{a_{k,l}}{j} + b_{k,l} - j\left(\frac{a_{k,l}}{j+1} - \frac{a_{k,l}}{j}\right) \tag{28}$$

where $j = 1, 2, \ldots, J$. Subsequently,

$$\hat{f}_{k,l}(\mathbf{a}_c) = \max\{\alpha_{k,l,j}\eta_l(\mathbf{a}_{-c}) + \beta_{k,l,j}\} \quad j = 1, 2, \ldots, J \tag{29}$$

where $\eta_l(\mathbf{a}_c, \mathbf{a}_{-c})$ (introduced earlier in Equation 13) is a linear function given by

$$\eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) = \sum_{k\in\mathcal{K}_c} x_{k,l} + N_{-c} \tag{30}$$

N_{-c} is the number of trucks belonging to other fleets, which we consider to be a constant. Observe that (29) is convex but makes a major change in the cost of $\eta_l\ (\mathbf{a}_c, \mathbf{a}_{-c}) = 0$. A new formulation is then proposed to eliminate this effect. Substituting (30) into (29), we have $\forall j = 1, 2, \ldots, J$,

$$\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \max\left\{\alpha_{k,l,j}\left(\left(\sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} + x_{k,l}\right) + N_{-c}\right) + \beta_{k,l,j}\right\} \tag{31}$$

Taking advantage of the binary nature of $\boldsymbol{x_{k,l}}$, we have

$$x_{k,l}\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \begin{cases} \hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) & x_{k,l} = 1 \\ 0 & x_{k,l} = 0 \end{cases} \tag{32}$$

We now introduce an auxiliary $u_{k,l}$ to relax the fuel cost, that is,

$$x_{k,l}\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) \le u_{k,l} \tag{33}$$

By further applying the pointwise maximum method, we have

$$\alpha_{k,l,j}x_{k,l}^2 + \alpha_{k,l,j}x_{k,l}\sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} + (\alpha_{k,l,j}N_{-c} + \beta_{k,l,j})x_{k,l} \le u_{k,l} \tag{34}$$

where $j = 1, \ldots, J$. Again, since $x_{k,l}$ is binary, $x_{k,l}^2 = x_{k,l}$. Let us apply another auxiliary variable $v_{k,l}$ to replace bilinear terms $x_{k,l}\sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l}$. The constraints are reformulated as

$$(\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c} + \beta_{k,l,j})x_{k,l} + \alpha_{k,l,j}v_{k,l} \le u_{k,l} \quad j = 1, \ldots, J \tag{35}$$

with

$$v_{k,l} \le (K_c - 1)x_{k,l} \tag{36}$$

$$v_{k,l} \le \sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \tag{37}$$

Note that (36) guarantees that there is no platooning effect on links where the truck does not traverse, meaning that if $x_{k,l} = 0$, $v_{k,l}$ is no greater than 0 and thus no saving is achieved in the term $\alpha_{k,i,j}v_{k,l}$. Otherwise, when $x_{k,l} = 1$, (36) does not bound $v_{k,l}$ since $\sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \le K_c - 1$ by definition. Equation (37) limits the platooning effects by bounding the maximum platooning partners.

4.2 | LP relaxation on the schedule preference penalty

The schedule preference penalty $h_k(\mathbf{a}_c)$, which is defined in (19), may be relaxed by introducing an auxiliary parameter $p_k \in \mathbb{R}$ as follows:

$$w_{k,l}\left(\mathbf{1}_{L_k}^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}\right) \le p_k \tag{38}$$

$$-w_{k,e}\left(\mathbf{1}_{L_k}^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}\right) \le p_k \tag{39}$$

where the variables are defined in (16).

4.3 | Overview of the relaxed problem

We now combine the original problem $\boldsymbol{\mathcal{P}}$ with the constraints (38), (39), (34), (36), and (37) to arrive at a relaxed

problem formulation as given in $\mathcal{R}$

$$\begin{aligned} \min_{\mathbf{a}_c} \quad & \sum_{k\in\mathcal{K}_c}\left[\sum_{l\in\mathcal{E}(\mathcal{G}_k)} u_{k,l} + g_k(\mathbf{a}_c) + p_k\right] \\ \text{subject to} \quad & (38),(39),(34),(36),(37) \end{aligned} \qquad (\mathcal{R})$$

The feasible solutions of $\mathcal{R}$ are at integer points, making the piecewise fitting equal to the original objective values at all feasible points. Relaxation methods introduce three types of auxiliary variables, which only serve as placeholders without changing feasible objective values. $\mathcal{R}$ is guaranteed to have the same optimum objective value as $\mathcal{P}$. It also matches the form of a standard MILP problem. The integer value constraints make the problem generally difficult to solve. A few examples in the following section show the trend of increasing computational time against increasing number of nodes and fleet size for solving the problem formulation $\mathcal{R}$.

4.4 | Numerical examples with centralized solution

In this section, we focus on simulating the computational time for solving the formulation $\mathcal{R}$ with a centralized solver. The computational time is defined as the time duration from the demand set input to the delivery of the travel plan set.

4.4.1 | Simulation setup

To simulate the highway network, we assume there are a set of nodes on a plane, which represent the set of nodes in $\mathcal{G}$. The nodes are connected by horizontal and vertical unweighted links to represent a connected transportation network. The graph is also known as the Hanan grid, which is adapted for similar simulations in previous studies (Abdolmaleki et al., 2021; Sokolov et al., 2017). The number of nodes used in the simulations is 9, 16, 36, and 100 to represent the enlargement of the graph size. The size of the fleet increases from 5 to 50 at a step size of 5. The assignments for trucks are randomly chosen from the graph. The latest arrival time is chosen in such a way that feasible paths exist. The preferred arrival time is also randomly selected during the time window that is possible to satisfy. There are two speed options, high and low. A truck may choose to spend one time interval or two on traveling through one link. We use simple integers for the simulation to test the performance of the centralized solving method. The piecewise fitting parameters are calculated based on the assumption that for a truck to travel through one edge, it takes 22 and 20 units of fuel for high and low speeds, respectively, and each unit of fuel weighs 1.47. The constants and assumptions used in this experiment are shown in Table 4. The numericals are solved using the CVX solver (Grant & Boyd, 2008, 2014), and the simulations on each fleet size have been repeated for 50 runs.

TABLE 4 Parameter settings for the numerical example simulation with centralized solution

Truck parameters			
Fleet size	K_c	**Range**	**Step size**
		[0,50]	**5**
Time cost weight	$w_{k,t}$	5	
Preference penalty–late	$w_{k,l}$	5	
Preference penalty–early	$w_{k,e}$	5	
Piecewise linear fitting parameters			
Speed		**High**	**Low**
Segment 1 constants	α	−1.617	−1.47
	β	33.957	30.87
Segment 2 constants	α	0	0
	β	30.723	27.93

4.4.2 | Performance analysis of the centralized solver

It is worth mentioning that the simple case in this simulation focuses on the increasing trend and the variation in the solving time. The absolute time in the result does not represent a real case. The results regarding the solving time in the small- and medium-size graph are shown in Figure 4, where the plots indicate time versus fleet size. In subgraphs Figure 4a,b for nodes 9 and 16, although the fleet sizes increase, the elapsed time is relatively short. In Figure 4d, we enlarge the regions of 0–800 s to show that the required computational time does not increase exponentially in this range in a majority of cases. However, depending on the feature of the demand from trucks, extreme cases may occur, leading to impractical computation time.

In a large-scale problem with 100 nodes in the graph, the extreme cases take significantly more time to compute; thus, the simulation terminates before raising the fleet size to 20. As shown in Figure 5a, there are extreme cases that cost different orders of magnitude of time for solving even with the same fleet size. The subplot is then enlarged, as shown in Figure 5b, suggesting that the solving time may vary significantly with increasing fleet size. Three histograms of fleet sizes 5 to 15 are given in Figure 5c–e, in which the y-axis represents the number

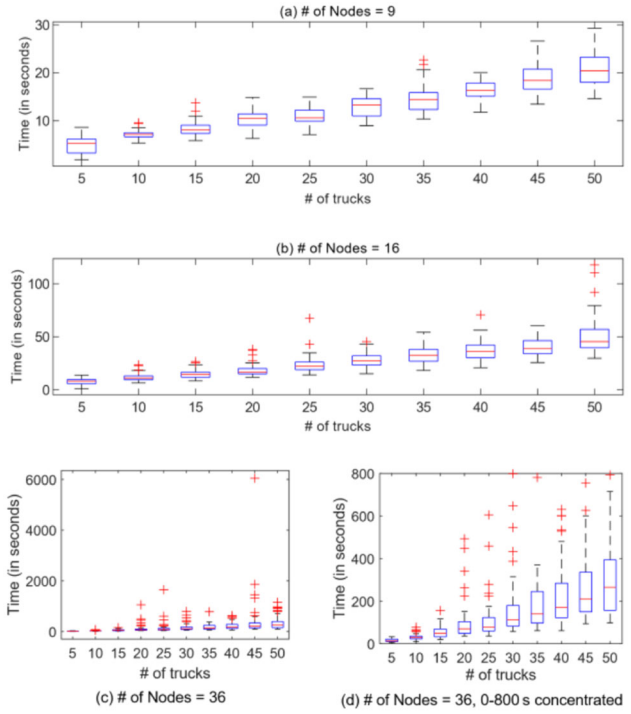

FIGURE 4 Computational time simulation result with centralized solver for varying numbers of trucks and nodes

of cases that fall into certain ranges of elapsed time. In all three scenarios, the majority of cases finish in a relatively short time, while there are about 10% of cases that are outliers. The result also suggests that the solving time is affected by the features of the demand, as well as the problem size. In fact, in a real-world problem, the graph and fleet numbers are likely to be fixed. A representative demand set may reveal a more practical solving time challenge in truck platooning coordination. On the other hand, the increasing computation time for increasing nodes and fleet size is not acceptable for various stakeholders—one of the challenges that we address in the following sections

5 | DECENTRALIZED SOLUTION

In this section, we implement the decentralized method to the coordination problem formulated in $\mathcal{R}$. We introduce the principle behind this method and present a simulation in comparison with the results of the centralized solver. We summarize this section with a discussion of the proposed algorithms.

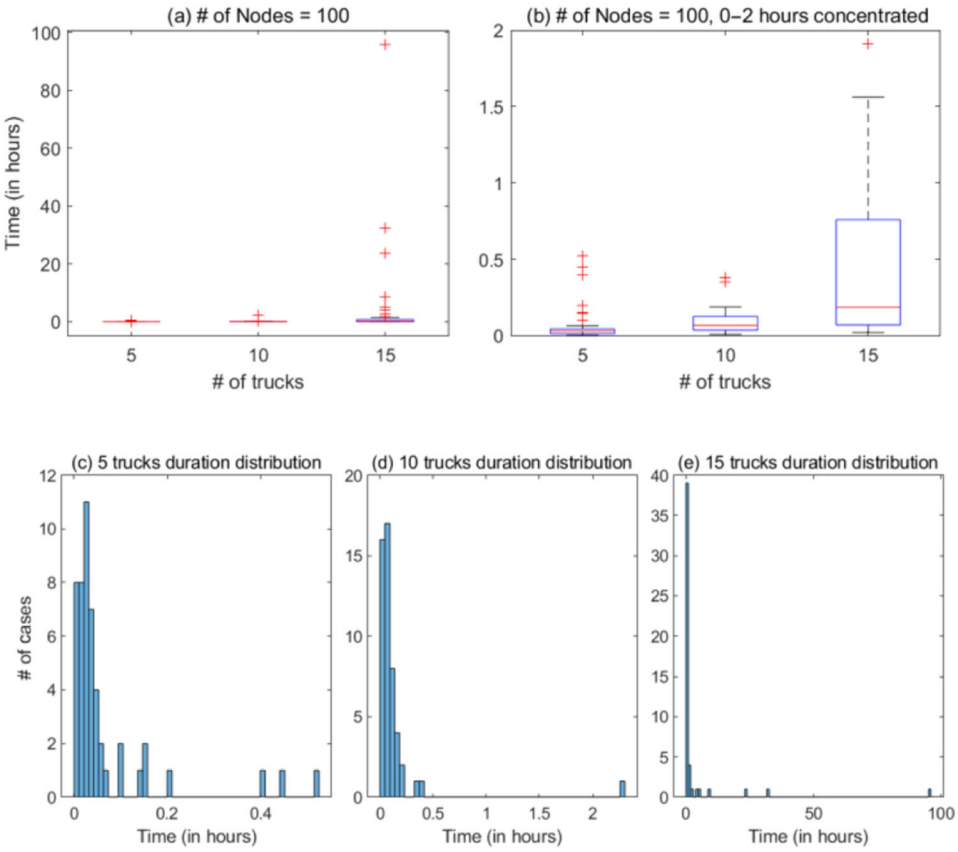

FIGURE 5 100-node graph computational time simulation result with centralized solver for varying numbers of trucks. (a) and (b) show the time complexity for 100 nodes, and (c)–(e) show the distribution of the solver operating time when the number of trucks = 5, 10, and 15

5.1 | Reformulation for the decentralized solution

The coordination problem $\mathcal{R}$ is restated as

$$\min_{\mathbf{a}_c} \sum_{k\in\mathcal{K}_c}\left[\sum_{l\in\mathcal{E}(\mathcal{G}_k)} u_{k,l} + g_k(\mathbf{a}_c) + p_k\right] \quad (\mathcal{R})$$

$$\text{subject to} \quad \mathbf{A}_k\mathbf{S}_k\mathbf{a}_c = \mathbf{b}_k \quad \forall k \in \mathcal{K}_c \quad (40)$$

$$\mathbf{a}_c \in [0,1]^{L_c} \quad \forall k \in \mathcal{K}_c \quad (41)$$

$$w_{k,l}(\mathbf{1}_{L_k}^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}) \le p_k \quad \forall k \in \mathcal{K}_c \quad (42)$$

$$-w_{k,e}(\mathbf{1}_{L_k}^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}) \le p_k \quad \forall k \in \mathcal{K}_c \quad (43)$$

$$(\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c} + \beta_{k,l,j})x_{k,l} + \alpha_{k,l,j}v_{k,l} \le u_{k,l} \quad \forall j, \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \quad (44)$$

$$v_{k,l} \le (K_c - 1)x_{k,l} \quad \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \quad (45)$$

$$v_{k,l} \le \sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \quad \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \quad (46)$$

where constraints (40) to (46) are based on (38), (39), (34), (36), and (37). By reformulating $\mathcal{R}$ into a matrix form, the formulation is compact and easier to denote. On the other hand, by placing the constants and parameters with physical significance within a matrix, we may temporally focus only on the mathematical structure. We will introduce the details of the reformulation in the rest of this section. We will introduce the details of the reformulation in the rest of this section. Let $\mathbf{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{(2L_k+1)}$ be a vector for each truck,

$$\mathbf{z}_k = \left[\mathbf{x}_k^T, p_k, \mathbf{u}_k^T, \mathbf{v}_k^T\right]^T \quad (47)$$

where

$$\mathbf{u}_k = [u_{k,1}, \dots, u_{k,l}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}_k) \quad (48)$$

$$\mathbf{v}_k = [v_{k,1}, \dots, v_{k,l}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}_k) \quad (49)$$

where $\mathbf{u}_k$ is the vector of fuel cost of truck k for all links l, and $\mathbf{v}_k$ represents the product of the decision variable $x_{k,l}$ and the number of platoon partners. This indicates the level of platooning benefits for truck k and thus equals 0 (since $x_{k,l} = 0$) when truck k is not present on link l.

Finally, $\mathbf{z}_k$ is considered as the extended decision vector of each truck.

For each truck k, we may have a cost vector $\mathbf{c}_k$ based on the cost function in $\mathcal{R}$,

$$\mathbf{c}_k^T\mathbf{z}_k = g_k(\mathbf{a}_c) + p_k + \sum_{l\in\mathcal{E}(\mathcal{G}_k)} u_{k,l} = w_{t,k}\mathbf{1}_{L_k}^T\mathbf{x}_k + p_k + \mathbf{1}_{L_k}^T\mathbf{u}_k \quad (50)$$

where $\mathbf{c}_k$,

$$\mathbf{c}_k = [w_{t,k}\mathbf{1}_{L_k}^T, 1, \mathbf{1}_{L_k}^T, \mathbf{0}_{L_k}^T]^T \quad (51)$$

Likewise, we reformulate the equality constraint in (40) as

$$\overline{\mathbf{A}}_k\mathbf{z}_k = \overline{\mathbf{b}}_k \quad (52)$$

where

$$\overline{\mathbf{A}}_k = [\mathbf{A}_k, \mathbf{0}_{L_k\times(2L_k+1)}] \quad (53)$$

$$\overline{\mathbf{b}}_k = \left[\mathbf{b}_k^T, \mathbf{0}_{(2L_k+1)}^T\right]^T \quad (54)$$

$$\underbrace{\begin{bmatrix} w_{k,l}\mathbf{1}_{L_k}^T & -1 & \mathbf{0}_{L_k}^T & \mathbf{0}_{L_k}^T \\ -w_{k,e}\mathbf{1}_{L_k}^T & -1 & \mathbf{0}_{L_k}^T & \mathbf{0}_{L_k}^T \\ \mathbf{C}_{k,x} & \mathbf{0}_{JL_k} & \mathbf{C}_{k,u} & \mathbf{C}_{k,v} \\ (K_c-1)\mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & -\mathbf{I}_{L_k\times L_k} \\ -\mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} \\ \mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} \end{bmatrix}}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ p_k \\ \mathbf{u}_k \\ \mathbf{v}_k \end{bmatrix}}_{\mathbf{z}_k} \le \underbrace{\begin{bmatrix} -w_{k,l}(t_k^{ED} - t_k^{PA}) \\ w_{k,e}(t_k^{ED} - t_k^{PA}) \\ \mathbf{0}_{JL_k} \\ \mathbf{0}_{L_k} \\ \mathbf{0}_{L_k} \\ \mathbf{1}_{L_k} \end{bmatrix}}_{\mathbf{d}_k} \quad (55)$$

which indicates the flow conservation constraints, ensuring that $\mathbf{x}_k$ represents a feasible path. The inequality constraints (38), (39), (34), and (36) are rewritten in, where $[\mathbf{C}]_i$ denotes the ith row of matrix $\mathbf{C}$ and

$$[\mathbf{C}_{k,x}]_{J(l-1)+l}\mathbf{x}_k = (\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c} + \beta_{k,l,j})x_{k,l} \quad (56)$$

$$[\mathbf{C}_{k,v}]_{J(l-1)+l}\mathbf{v}_k = \alpha_{k,l,j}v_{k,l} \quad (57)$$

$$[\mathbf{C}_{k,u}]_{J(l-1)+l}\mathbf{u}_k = -u_{k,l} \quad (58)$$

Note that the last two rows in the matrix on the left side in () constrain the binary variable $x_{k,l}$ into the range of [0,1].

The complex formulation in () may be simplified

$$\mathbf{H}_k\mathbf{z}_k \le \mathbf{d}_k \quad (59)$$

The matrix $\mathbf{H}_k$ and $\mathbf{d}_k$ represents the feasible half space defined by inequality constraints The remaining constraints (37) may be written as

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \le \mathbf{0}_{L_c} \tag{60}$$

where $\mathbf{D}_k \in \mathbb{R}^{(3L_k+1)\times L_c}$ satisfies and $L_c = \sum_{k\in\mathcal{K}_c} L_k$

$$\sum_{k\in\mathcal{K}_c} [\mathbf{D}_k]_l \mathbf{z}_k = \upsilon_{k,l} - \sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \tag{61}$$

Notably, (59) and (52) are local for a truck; let $\mathcal{Z}_k$ be a set

$$\mathcal{Z}_k = \left\{ \mathbf{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{2L_k+1} | \mathbf{H}_k \mathbf{z}_k \le \mathbf{d}_k, \overline{\mathbf{A}}_k \mathbf{z}_k = \overline{\mathbf{b}}_k \right\} \tag{62}$$

To summarize, $\boldsymbol{\mathcal{R}}$ is reformulated as

$$\min_{\mathbf{z}_1,\dots,\mathbf{z}_{K_c}} \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k \tag{$\boldsymbol{\mathcal{R}}_d$}$$

$$\text{subject to} \quad \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \le \mathbf{0}_{L_c} \tag{$\boldsymbol{\mathcal{R}}_{d.1}$}$$

$$\mathbf{z}_k \in \mathcal{Z}_k \quad \forall k \in \mathcal{K}_c \tag{$\boldsymbol{\mathcal{R}}_{d.2}$}$$

$\boldsymbol{\mathcal{R}_d}$ is an MILP problem with coupled inequality constraints among trucks. It is worth mentioning that constraint ($\boldsymbol{\mathcal{R}}_{\mathbf{d}}$.1) is connected to every truck. However, the rest of the constraints are local. To decompose the problem, we adopt the decentralized method with minor modification in the notations, which is proposed by Falsone et al. (2019).

5.2 | Dual decomposition

The dual problem of $\boldsymbol{\mathcal{R}_d}$ is formulated as

$$\max_{\boldsymbol{\lambda}} \min_{\mathbf{z}_1,\dots,\mathbf{z}_{K_c}} \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k + \boldsymbol{\lambda}^T \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k$$

$$\text{subject to} \quad \boldsymbol{\lambda} \ge \mathbf{0}_{L_c} \tag{$\boldsymbol{\mathcal{D}}$}$$

$$\mathbf{z}_k \in \mathcal{Z}_k \quad \forall k \in \mathcal{K}_c$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{L_c}$ is the Lagrange multiplier for the coupled inequality constraints.

$$L(\mathbf{z}_1, \dots, \mathbf{z}_k, \boldsymbol{\lambda}) = \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k + \boldsymbol{\lambda}^T \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \tag{63}$$

$$= \sum_{k\in\mathcal{K}_c} (\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k) \mathbf{z}_k \tag{64}$$

where the Lagrangian is decomposable. Hence, for a given $\boldsymbol{\lambda}$, solving for $\mathbf{z}_k$ is equivalent to

$$\mathbf{z}_k(\boldsymbol{\lambda}) \in \arg\min_{\mathbf{z}_k\in\mathcal{Z}_k} \quad (\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k) \mathbf{z}_k \tag{65}$$

where $\boldsymbol{\lambda}$ is now the global variable that should be exchanged in the communication network. This approach, also known as the projected dual subgradient method (Boyd et al., 2003), aims to solve the following equations iteratively:

$$\mathbf{z}_{k,i+1} = \arg\min_{\mathbf{z}_k\in\mathcal{Z}_k} \quad (\mathbf{c}_k^T + \boldsymbol{\lambda}_i^T \mathbf{D}_k) \mathbf{z}_k \tag{66}$$

$$\boldsymbol{\lambda}_{i+1} = \left[\boldsymbol{\lambda}_i + \alpha(i) \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_{k,i} \right]_+ \tag{67}$$

where with an abuse of notation, we denote $\{\mathbf{z}_{k,i}, \boldsymbol{\lambda}_i\}$ as the values of corresponding $\{\mathbf{z}_k, \boldsymbol{\lambda}\}$ at the ith iteration. The operation $[\cdot]_+$ is the projection to a nonnegative orthant, and $\alpha(i)$ is the step size at iteration i. With a centralized fusion center for the $\boldsymbol{\lambda}$ update, the problem is separated into smaller problems for each truck.

However, the dual decomposition method does not guarantee a feasible solution because of the discrete variables. Counter examples are given in the Appendix of Falsone et al. (2019). This approach, however, provides an optimal solution of a relaxed problem

$$\min_{\mathbf{z}_1,\dots,\mathbf{z}_{K_c}} \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k$$

$$\text{subjectto} \quad \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \le \mathbf{0}_{L_c}$$

$$\mathbf{z}_k \in \text{conv}(\mathcal{Z}_k) \quad \forall k \in \mathcal{K}_c \tag{$\boldsymbol{\mathcal{R}}_{\text{LP}}$}$$

where $\text{conv}(\mathcal{Z}_k)$ is the convex hull of the local set $\mathcal{Z}_k$. To offer more detailed insights into the duality gap between $\boldsymbol{\mathcal{R}_d}$ and $\boldsymbol{\mathcal{D}}$, we restate a theorem from Vujanic et al. (2016) in the context of our problem.

Theorem 1. *(Bound on duality gap, Theorem 2.3 in Vujanic et al., 2016). There exists an* $\overleftarrow{\mathbf{z}}_k \in \mathcal{Z}_k$ *such that* $\boldsymbol{D}_k \overleftarrow{\mathbf{z}}_k \le \boldsymbol{D}_k \boldsymbol{z}_k, \forall \boldsymbol{z}_k \in conv(\mathcal{Z}_k)$*, then*

$$J^*_{\mathcal{R}_d} - J^*_{\mathcal{D}} \le K_c \max_{k\in\mathcal{K}_c} \gamma_k, \tag{68}$$

where

$$\gamma_k = \max_{\mathbf{z}_k\in\mathcal{Z}_k} \mathbf{c}_k^T \mathbf{z}_k - \min_{\mathbf{z}k\in\mathcal{Z}_k} \mathbf{c}_k^T \mathbf{z}_k \tag{69}$$

The theorem suggests the existence of a feasible solution within the bound given in (68). However, there is no algorithmic way to produce the solution. To address the issues

for possible infeasible solutions, a method is proposed by tightening the primal problem to ensure feasibility for the dual decomposition solutions (Vujanic et al., 2016). We present the related parts in the next subsection.

5.3 | Primal problem modification

The method proposed in Vujanic et al. (2016) is based on tightening the feasible region limited by (60) with an appropriate contraction. The contraction is to tighten the inequality constraints to guarantee feasibility, although it may increase the suboptimal level. The tightened version of the relaxed problem $\boldsymbol{\mathcal{R}}_{\mathbf{LP}}$ is given by

$$\begin{aligned} \min_{\mathbf{z}_1,\dots,\mathbf{z}_{K_c}} \quad & \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k \\ \text{subject to} \quad & \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \le -\boldsymbol{\rho} \qquad \boldsymbol{\mathcal{R}}_{\mathbf{LP},\boldsymbol{\rho}} \\ & \mathbf{z}_k \in \mathrm{conv}(\mathcal{Z}_k) \quad \forall k \in \mathcal{K}_c \end{aligned}$$

where $\boldsymbol{\rho} \in \mathbb{R}^{L_c}$ is nonnegative and the tightening vector. Consequently, the dual problem of $\boldsymbol{\mathcal{R}}_{\mathbf{LP},\boldsymbol{\rho}}$ is given by

$$\max_{\boldsymbol{\lambda}\ge\mathbf{0}} \quad \boldsymbol{\lambda}^T\boldsymbol{\rho} + \sum_{k\in\mathcal{K}_c} \min_{\mathbf{z}_k\in\mathcal{Z}_k} (\mathbf{c}_k^T + \boldsymbol{\lambda}^T\mathbf{D}_k)\mathbf{z}_k \boldsymbol{D}_\rho$$

Notably, the contraction problem does not affect the subproblem in (66). Let $\tilde{\rho}$ be a proper selection of ρ

$$[\tilde{\boldsymbol{\rho}}]_j = K_c \left\{ \max_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k - \min_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k \right\} \tag{70}$$

where $[\cdot]_j$ is the jth row of the corresponding matrix.

Now, let $\tilde{\boldsymbol{R}}_{\boldsymbol{Lp},\boldsymbol{\rho}}$ and $\tilde{\boldsymbol{D}}_{\boldsymbol{\rho}}$be the primal-dual pairs, where $\boldsymbol{\rho} = \tilde{\boldsymbol{\rho}}$. The contraction method with $\tilde{\rho}$ is different from the original formulation in Vujanic et al. (2016), in which the latter assumes the length of $\boldsymbol{\rho}$ to be considerably smaller with respect to the number of agents. In general, this assumption does not hold in the truck platooning problem as in formulation $\boldsymbol{\mathcal{R}}_{\boldsymbol{d}}$, as $L_c \ge K_c$, which implies that there are significantly more links in the spatiotemporal graph than the number of trucks. Now, we will prove that the feasibility of the original method in Vujanic et al. (2016) remains true with this change in the assumption in the truck platooning coordination problem.

We may assume that for an optimal solution set $\tilde{\boldsymbol{z}}^*_{LP} = [\tilde{\boldsymbol{z}}^*_{1,\boldsymbol{LP}}; \dots; \tilde{\boldsymbol{z}}^*_{K_c,\boldsymbol{LP}}]$ of $\boldsymbol{\mathcal{R}}_{\mathbf{LP},\boldsymbol{\rho}}$, a subset $\mathcal{K}_{c,1} \subseteq \mathcal{K}_c$ exists such that

$$\tilde{\mathbf{z}}^*_{k,LP} = \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) \quad \forall k \in \mathcal{K}_{c,1} \tag{71}$$

where $\tilde{\boldsymbol{\lambda}}^*$ is an optimal solution of $\boldsymbol{D}_\rho$ and $\boldsymbol{z_k}(\tilde{\boldsymbol{\lambda}}^*)$ is the corresponding solution given by (65). The subset $\mathcal{K}_{c,1}$ contains the trucks that have the same optimal plan with or without the contraction method implemented. By dividing the trucks into two sets, we have the following derivations:

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) = \sum_{k\in\mathcal{K}_{c,1}} \mathbf{D}_k \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) + \sum_{k\in\mathcal{K}_c\setminus\mathcal{K}_{c,1}} \mathbf{D}_k \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) \tag{72}$$

Notably, if $\tilde{z}^*_{LP}$ is unique, we have

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k \tilde{\mathbf{z}}^*_{k,LP} \le -\tilde{\boldsymbol{\rho}} \tag{73}$$

On the conditions that $\boldsymbol{\mathcal{R}}_{\mathbf{LP},\boldsymbol{\rho}}$ has multiple optimal solutions, each truck may reach a local solution, which is a part of an optima, but there is no guarantees that all trucks' iterate to the same optima, which possibly fails the inequality condition in (73). On the other hand, based on (65), $\boldsymbol{z_k}(\tilde{\boldsymbol{\lambda}}^*)$ is determined by $\tilde{\boldsymbol{\lambda}}^*$. If $\tilde{\boldsymbol{\lambda}}^*$ is not unique, this may result in an uncertainty in the selection of $\tilde{\boldsymbol{\lambda}}^*$of each truck. In summary, the following assumption is proposed:

Assumption 1. (Uniqueness, Assumption 2.4 in Vujanic et al., 2016; $\tilde{\boldsymbol{R}}_{\boldsymbol{Lp},\boldsymbol{\rho}}$ and $\tilde{\boldsymbol{D}}_{\boldsymbol{\rho}}$ have unique solutions $\tilde{\boldsymbol{z}}^*_{\boldsymbol{LP}}$ and $\tilde{\boldsymbol{\lambda}}^*$).

If Assumption 1 holds, we have (73) true and subsequently,

$$\begin{aligned} \sum_{k\in\mathcal{K}_{c,2}} & \left([\mathbf{D}_k]_j \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) - [\mathbf{D}_k]_j \tilde{\mathbf{z}}^*_{k,LP} \right) \\ & \le K_c \max_{k\in K_c} \left\{ \max_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k - \min_{\mathbf{z}_k\in\mathrm{conv}(\mathcal{Z}_k)} [\mathbf{D}_k]_j \mathbf{z}_k \right\} \end{aligned} \tag{74}$$

It is straightforward to see that it held since there are at most K_c trucks in $\mathcal{K}_c \setminus \mathcal{K}_{c,1}$. Furthermore, based on (61), it is clear that $[\mathbf{D}_k]_j \mathbf{z}_k$ is either $v_{k,l}$ or $-x_{k,l}$. The linearity of $[\mathbf{D}_k]_j \mathbf{z}_k$ guarantees that $\min_{\mathbf{z}_k\in\mathrm{conv}(\mathcal{Z}_k)} [\mathbf{D}_k]_j \mathbf{z}_k = \min_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k$ since the binary elements will be on the extreme edges, which for truck platooning is either −1 or 0 by definition. This supports that (70) is a reasonable choice. Following (72),

$$\begin{aligned} \sum_{k\in\mathcal{K}_c} \mathbf{D}_k \tilde{\mathbf{z}}^*_{k,LP} + \sum\nolimits_{k\in\mathcal{K}_c\setminus\mathcal{K}_{c,1}} & \left(\mathbf{D}_k \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) - \mathbf{D}_k \tilde{\mathbf{z}}^*_{k,LP} \right) \\ & \le -\boldsymbol{\rho} + \boldsymbol{\rho} = \mathbf{0}_{L_c} \end{aligned} \tag{75}$$

To summarize, the following theorem is given.

Theorem 2. *(Feasibility guarantees, Theorem 3.1 in Vujanic et al., 2016). If Assumption 1 is satisfied,* $z^*(\tilde{\boldsymbol{\lambda}}^*) = [\boldsymbol{z}^*_1(\tilde{\boldsymbol{\lambda}}^*); \dots; \boldsymbol{z}^*_{K_c}(\tilde{\boldsymbol{\lambda}}^*)]$ *is feasible for* $\boldsymbol{\mathcal{R}}_{\boldsymbol{d}}$.

However, it worth mentioning that in the truck platooning problem, the chosen $\tilde{\boldsymbol{\rho}}$ is unnecessarily large, compromising on the suboptimal level. We may build a

tighter $\boldsymbol{\rho_o}$ based on (61),

$$\boldsymbol{\rho}_o = 2(K_c - 1) \cdot \mathbf{1}_{L_c} \tag{76}$$

Even with $\boldsymbol{\rho}_o$, the tightening is strict and always assumes the worst case, which may be optimized. A method has been proposed to combine the dual subgradient method to the adaptive tighten primal problem with improvement in the suboptimal level and asymptotic convergence guarantees (Falsone et al., 2019). We modified the algorithm to meet the properties of our problem.

5.4 | Decentralized truck platooning coordination

A method has been proposed to combine the dual subgradient method to the adaptive tighten primal problem with improvement in the suboptimal level and asymptotic convergence guarantees (Falsone et al., 2019). We modified the algorithm to meet the properties of our problem. Instead of fixing the value of $\boldsymbol{\rho}$, the algorithm is adaptive in contracting the primal problem to achieve a better suboptimal level. There are two main parts of preparing the procedure before the implementation of the algorithm. To satisfy Assumption 1, a perturbation in the cost vector $\mathbf{c}_k$ may be needed. However, since in the real world, the fuel on each link is likely to be unique, leading to unique solutions, the step may not be necessary.

One of the critical assumptions in previous works is that in Step 7, the feasible set must be bounded (Falsone et al., 2019). Since the objective function is modified as $(\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k)\mathbf{z}_k$, proper upper bounds for $u_{k,l}$ and lower bounds for $v_{k,l}$ should be assigned. The extra bounds are given by

$$v_{k,l} \geq 0, \quad u_{k,l} \leq \alpha_{k,l,1} + \beta_{k,l,1} \tag{77}$$

which is nothing but the physical limitation. $v_{k,l}$ is a place holder for $x_{k,l} \sum_{k' \in \mathcal{K}_c \setminus k} x_{k',l}$, ensuring the positive semidefinite nature. $u_{k,l}$ is the placeholder for the fuel cost, limited by the physical nature. The fuel cost is at most when the truck transverses through the link alone. Likewise, we formulate (77) as

$$\underbrace{[\mathbf{0}_{L_k}^T, 0, \mathbf{1}_{L_k}^T, -\mathbf{1}_{L_k}^T]}_{\mathbf{M}_k} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ p_k \\ \mathbf{u}_k \\ \mathbf{v}_k \end{bmatrix}}_{\mathbf{z}_k} \leq \underbrace{\begin{bmatrix} \mathbf{0}_{L_k}^T \\ 0 \\ \boldsymbol{\alpha}_{k,1} + \boldsymbol{\beta}_{k,1} \\ \mathbf{0}_{L_k}^T \end{bmatrix}}_{\mathbf{n}_k} \tag{78}$$

The extra constraints given by (77) are also linear and local; thus, for a simple notation in the algorithm, we define

$$\hat{\mathcal{Z}}_k = \mathcal{Z}_k \cap \{\boldsymbol{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{2L_k+1} | \boldsymbol{M}_k \boldsymbol{z}_k \leq \boldsymbol{n}_k\} \tag{79}$$

The algorithm is a variant of the dual subgradient method for the tighten primal problem. After initialization, the algorithm locally computes a subproblem in Step 7. The following steps from 9 to 12 are adaptive steps for finding a minimum $\boldsymbol{\rho}$ for a better suboptimal level. The difference is quite significant with the previous (Vujanic et al., 2016) the contraction is fixed. Since taking values from finite sets, the sequence of $\boldsymbol{\rho}(i)$ converges to some $\overleftarrow{\boldsymbol{\rho}}$. In a similar manner, with $\boldsymbol{\rho} = \overleftarrow{\boldsymbol{\rho}}$, the contraction is still sufficient to ensure feasibility. We recommend that the reader refer to Falsone et al. (2019) for the proof. With the latest update on, $\boldsymbol{\rho}$, the centralized node performs an update on $\boldsymbol{\lambda}$ in Step 14. $\alpha(i)$ is a nonsummable diminishing step size with a positive scalar a (Boyd et al., 2003). Since the algorithm is based on a decentralized setup, there may not be a clear stop criterion unless information is sent to the centralized node. It is also practical to set upper bounds for iterations.

Algorithm 1. Decentralized truck platooning coordination

1: $i = 0$ ▷% Rounds of iteration
2: $\lambda(i) = 0$
3: $\bar{s}(i) = -\infty, k = 1, \ldots, \mathbf{K_c}$
4: $\underline{s}(i) = +\infty, k = 1, \ldots, \boldsymbol{K_c}$
5: **repeat**
6: **for** $k = 1$ **to** $\boldsymbol{K_c}$ **do**
7: $z_k(i+1) \leftarrow \arg\min_{\boldsymbol{z_k} \in \hat{\mathcal{X}}_k} \{(\boldsymbol{c_k^T} + \boldsymbol{\lambda}^{TD_k})\boldsymbol{z_k}\}$
8: **end for**
9: $\bar{\mathbf{s}}_k(i+1) \leftarrow \max\{\bar{\mathbf{s}}_k(i), \boldsymbol{D_k z_k}(i+1)\}, k = 1, \ldots, K_c$
10: $\underline{\mathbf{s}}_k(i+1) \leftarrow \min\{\underline{\mathbf{s}}_k(i), \boldsymbol{D}_k \boldsymbol{z}_k(i+1)\}, k = 1, \ldots, K_c$
11: $\boldsymbol{\rho}_k(i+1) = \bar{\mathbf{s}}_k(i+1) - \underline{\mathbf{s}}_k(i+1), \quad k = 1, \ldots, K_c$
12: $\rho(i+1) = K_c \max\{\rho_1(i+1), \ldots, \rho_{K_c}(i+1)\}$
13: $\alpha(i) = a/\sqrt{(i+1)}, \quad a > 0$
14: $\lambda(i+1) = [\lambda(i) + \alpha(i)(\sum_{i=1}^{m} \mathrm{b} D_{kz_k} + \rho(i+1))]_+$
15: $i \leftarrow i+1$
16: **until** some stop criterion is met

To summarize the flow of Algorithm 1, at iteration i, each truck locally updates the tentative solution $\mathbf{z}_k(\boldsymbol{\lambda}(i-1))$ and sends the value of $\mathbf{D}_k \mathbf{z}_k(\boldsymbol{\lambda}(i-1))$ to the centralized node. Based on the latest updates from trucks, a centralized node updates $\boldsymbol{\rho}(i)$, leading to an ascent to $\boldsymbol{\lambda}(i)$. It is noted

that the process does not require synchronization (Falsone et al., 2019), although we build up the algorithm in a synchronous setup for a result comparable to the centralized result.

5.5 | Numerical examples with a decentralized solution

In this section, we compare the computation time needed for the decentralized solution versus the centralized solution for increasing fleet size in the network. In addition, we address the compromise in optimality as a result of decomposing the centralized problem and solving it in a distributed fashion. A simple decentralized method is applied as the benchmark for Algorithm 1, which is an opportunistic platooning on each truck's solo optimal plan.

The simulation is performed on the same setup as introduced in Section 4.4.1 but only for the medium-sized 36-node Hanan graph. At each iteration of Algorithm 1, Steps 6–8 are considered parallel. Thus, only the worst case is considered. Assuming all trucks send the result to the centralized node after the slowest agent finished the computation, a centralized node performed the computation from Steps 9–14. The computational time of the parallel is given by the following rules. For each iteration, all trucks performed the local computation, and the longest time spent was set as the time of this iteration. The total time is given by summing up the duration of each iteration. Latency in communication is assumed to be none.

5.6 | Results and discussions

The improvement in computational time is shown in Figure 6. The computation time of the centralized solver is significantly influenced by the demand for trucks, with relatively longer cases and possibly impractical for a large-scale problem as shown in Figure 6a. In contrast, the decentralized algorithm shows better consistency in solving problems with different demands and sizes, as shown in Figure 6b. The result of the decentralized method is enlarged to highlight the range from 30 to 90 s as shown in Figure 6c. Although Algorithm 1 takes more time in small fleet sizes, it shows the advantage of solving a large-scale problem. This suggests that in scenarios with smaller fleet sizes, the centralized method may achieve an optimal solution and terminate in a short period of time and hence may not need many iterations. It would be an interesting topic to design a size-dependent iteration limit in future work. Given the size of the graph, the problem is broken down into smaller problems with almost the same size, which are not significantly affected by the number of trucks since they are conducted on connected vehicles in parallel. The time increment is mainly the result of centralized node computation, which is not significant.

The computational time performance is promising even in the real-world size problem, as the fleet size increment contributes to the advantages in the decentralized method. It is likely that Algorithm 1 still outperforms the centralized solver in terms of time, even with communication latency.

In this work, we provided some insights to improve the computational time of an MILP problem, which is typically computationally time inefficient. In the decentralized method, when the problem is broken down to each truck at each iteration, the problem becomes a special kind of MILP, which may be considered the shortest simple path in a weighted and directed graph. In essence, this is a $\mathcal{P}$ problem, which can be solved efficiently and in parallel using classic off-shelf solvers, such as Dijkstra's Algorithm (Dijkstra, 1959).

On the other hand, the improvement in time efficiency is the result of a compromise in optimality. The result is given in Figure 7, showing the performance in benefiting the stakeholders, compared to the centralized solution and forming platoons without coordination beforehand. The graph in Figure 7a is the relative gap between the optimal cost given by the centralized method and the suboptimal objective value from Algorithm 1. On average, the relative gap to the optimum is kept below 1%, with no significant increment with more trucks in the fleet. To reveal the benefits of employing the decentralized algorithm, a comparison with opportunistic platooning, in which all trucks operate without coordination beforehand, forms platoons once encountered during the trip. This comparison is to rule out the benefits received through the inherent overlap. Generally, Algorithm 1 reduces the cost by about 0.2% on average as shown in Figure 7b. Although the reduction is insignificant, the lowest subplot indicates that the algorithm realizes about 20% of the potential as shown in Figure 7c.

With the result so far, we learn that the implementation of Algorithm 1 is likely to benefit the stakeholders of the fleet, as it has a significant improvement in computational time. However, the absolute and relative objective value reduction depends on the demand, including the truck assignment distribution among the geographical graph and the period.

6 | CONCLUSION AND FUTURE WORK

In this paper, we built a mathematical model for the cooperative truck platooning problem with different pieces of the cost function. A series of relaxation methods are

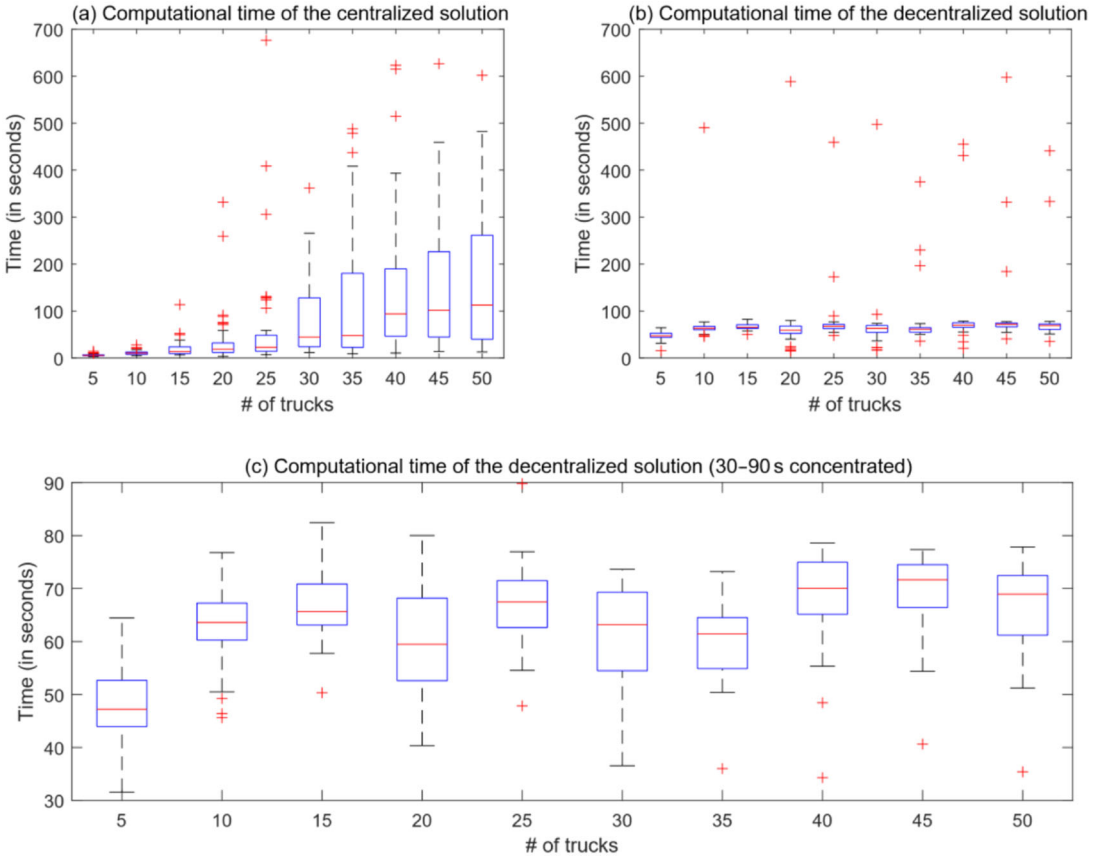

FIGURE 6 Computational time comparison between the centralized solution and the decentralized solution

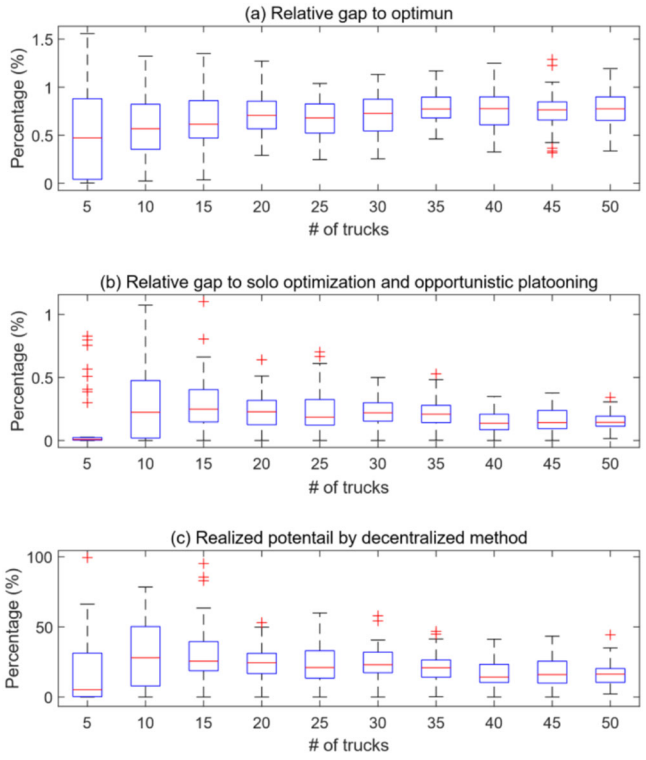

FIGURE 7 Suboptimal level and benefits of the decentralized solution

applied to handle the lack of structure of the problem $\mathcal{P}$, leading to the standard MILP form $\mathcal{R}$. Even with an efficient centralized algorithm, the problem remains difficult to solve with increasing scale. The increasing computational time variation is also a challenge for real companies. Therefore, a decentralized algorithm is proposed to be implemented, followed by a numerical analysis. The comparison result suggests that the proposed decentralized algorithm is robust even in large-scale problems. The elapsed time is relatively insensitive to the size of the fleet, making it more practical for real-world problems.

There are some interesting future directions to study on in the further research. By separating the large graph into smaller subgraphs or dividing the fleet into smaller groups, the problem is decomposed in another decentralized manner. From another perspective, the problem with smaller truck sets may not require much searching, and the iterations in the decentralized method may be inefficient, as all improvements are made in the first few iterations. A size-dependent stop criterion to improve the current decentralized algorithm is worth looking at.

It is also noticed that the computational time and the gap to optimal value are affected by the demand set and setting of weights, and hence a study involving real-world datasets should be explored. Real-world scenarios may involve

congestion effect and uncertainties in time prediction, which are not taken into consideration in this work. The trip coordination is performed before the departure, and a delay in arrival may cause the absence of an expected platooning partner, leading to a cost increment if the truck spent extra time or detoured for this platooning. Robust planning that minimizes the cost of the worst-case scenario or stochastic planning that ensures platooning coordination with a high probability threshold are possible ways to deal with travel time uncertainties explicitly. They are left for future studies. Moreover, the model is still limited to the trucks within the same fleet, and interactions among self-interested fleets are not introduced in this paper, which should be addressed in the next steps. The model also enables the possibility of preserving the privacy of participating companies since all decisions are made locally (Zeng, 2020).

REFERENCES

Abdolmaleki, M., Shahabi, M., Yin, Y., & Masoud, N. (2021). Itinerary planning for cooperative truck platooning. *Transportation Research Part B: Methodological*, *153*, 91–110. https://www.sciencedirect.com/science/article/pii/S0191261521001661

Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., & Zhang, K. (2016). Polaris: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. *Transportation Research Part C: Emerging Technologies*, *64*, 101–116. https://www.sciencedirect.com/science/article/pii/S0968090x15002703

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

Boyd, S., Xiao, L., & Mutapcic, A. (2003). Subgradient methods. Lecture notes of EE392o. *Stanford University, Autumn Quarter*, *2003*. https://web.stanford.edu/class/ee392o/subgrad_method.pdf

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

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*, 269–271.

Even, S., Itai, A., & Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. 16th Annual Symposium on Foundations of Computer Science (SFCS 1975), Berkeley, CA (pp. 184–193).

Falsone, A., Margellos, K., & Prandini, M. (2019). A decentralized approach to multi-agent MILPS: Finite-time feasibility and performance guarantees. *Automatica*, *103*, 141–150.

Franceschetti, A., Honhon, D., Van Woensel, T., Bektaş, T., & Laporte, G. (2013). The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, *56*, 265–293. http://www.sciencedirect.com/science/article/pii/S0191261513001446

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, & H. Kimura (Eds.), *Recent advances in learning and control* (pp. 95–110). Springer-Verlag Limited. http://stanford.edu/~boyd/graph_dcp.html

Grant, M., & Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx

Jiang, X., & Adeli, H. (2004a). Clustering-neural network models for freeway work zone capacity estimation. *International Journal of Neural Systems*, *14*(03), 147–163.

Jiang, X., & Adeli, H. (2004b). Object-oriented model for freeway work zone capacity and queue delay estimation. *Computer-Aided Civil and Infrastructure Engineering*, *19*(2), 144–156.

Johansson, A., & Mårtensson, J. (2019). Game theoretic models for profit sharing in multi-fleet platoons. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand (pp. 3019–3024).

Johansson, A., Nekouei, E., Johansson, K. H., & Mårtensson, J. (2018). Multi-fleet platoon matching: A game-theoretic approach. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI (pp. 2980–2985).

Larson, J., Kammer, C., Liang, K. -Y., & Johansson, K. H. (2013). Coordinated route optimization for heavy-duty vehicle platoons. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands (pp. 1196–1202).

Larson, J., Liang, K. -Y., & Johansson, K. H. (2014). A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, *16*(1), 419–429.

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

Masoud, N., & Jayakrishnan, R. (2017). A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem. *Transportation Research Part B: Methodological*, *99*, 1–29.

Nourmohammadzadeh, A., & Hartmann, S. (2016). The fuel-efficient platooning of heavy duty vehicles by mathematical programming and genetic algorithm. *International Conference on Theory and Practice of Natural Computing*, Sendai, Japan (pp. 46–57).

Nourmohammadzadeh, A., & Hartmann, S. (2018). Fuel efficient truck platooning with time restrictions and multiple speeds solved by a particle swarm optimisation. *International Conference on Theory and Practice Of Natural Computing*, Dublin, Ireland (pp. 188–200).

Schittler, M. (2003). *State-of-the-art and emerging truck engine technologies for optimized performance, emissions and life cycle costs*. (Tech. Rep.). *DaimlerChrysler AG (US)*.

Schroten, A., Warringa, G., & Bles, M. (2012). *Marginal abatement cost curves for heavy duty vehicles*. Background Report, CE Delft, Delft.

Significance Quantitative Research, V. U. University Amsterdam, & John Bates Services. (2013). *Values of time and reliability in passenger and freight transport in the Netherlands*. The Hague, The Netherlands, Ministry of Infrastructure and the Environment.

Sokolov, V., Larson, J., Munson, T., Auld, J., & Karbowski, D. (2017). Maximization of platoon formation through centralized routing and departure time coordination. *Transportation Research Record*, *2667*(1), 10–16.

Tong, L., Zhou, X., & Miller, H. J. (2015). Transportation network design for maximizing spaceime accessibility. *Transportation Research Part B: Methodological*, *81*, 555–576. https://www.sciencedirect.com/science/article/pii/S191261515001711X

Varaiya, P. (1993). Smart cars on smart roads: Problems of control. *IEEE Transactions on Automatic Control*, *38*(2), 195–207.

Vujanic, R., Esfahani, P. M., Goulart, P. J., Mariéthoz, S., & Morari, M. (2016). A decomposition method for large scale MILPS, with performance guarantees and a power system application. *Automatica*, *67*, 144–156.

Wei, S., Feng, S., Ke, J., & Yang, H. (2022). Calibration and validation of matching functions for ride-sourcing markets. *Communications in Transportation Research*, *2*, 100058. https://www.sciencedirect.com/science/article/pii/S2772424722000087

Zeng, Y. (2020). *Distributed coordination for multi-fleet truck platooning*. [Master's thesis, TU Delft Electrical Engineering].

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

How to cite this article: Zeng, Y., Wang, M., & Rajan, R. T. (2022). Decentralized coordination for truck platooning. *Computer-Aided Civil and Infrastructure Engineering*, 1–19. https://doi.org/10.1111/mice.12899