Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
http://ens.ewi.tudelft.nl/

CAS-2020-4836332

M.Sc. Thesis

Distributed Coordination for Multi-fleet Truck Platooning

Yikai Zeng B.Sc.

Abstract

Truck platooning refers to coordinating a group of heavy-duty vehicles at a close inter-vehicle distance to reduce overall fuel consumption. This coordination between trucks is traditionally achieved by adjusting the schedule, velocity and routines to increase the platooning chances, and thus improve the overall fuel efficiency. However, the data model built for the coordination problem is typically integer-constrained, making it generally hard to solve. On the other hand, the interaction among self-interested fleets which are operated by different companies is not well-studied. This thesis aims to build a distributed framework for multi-fleet truck platooning coordination to enable the coordination without a third-party service provider. The interaction among fleets is considered a non-cooperative finite game, for which we propose the best response search method, which essentially requires to solve a cooperative truck platooning optimization problem iteratively. We refer to the optimization problem as a best-response problem, which is formulated as a mixed-integer linear problem with relaxation skills. To achieve a feasible time complexity for the best-response subproblem, we propose a decentralized algorithm, distributing the computational load to connected automated vehicles within the fleet. The proposed method is examined under a real-world featured demand set to compare the performance in optimality and time complexity with previous studies. The result suggests that the decentralized algorithm delivers the optimal objective value in this case, while the best-response search does not deliver extra benefits as the dominating time costs in the cost functions eliminate the potential for improvement.

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Distributed Coordination for Multi-fleet Truck Platooning

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Yikai Zeng B.Sc.
born in Chengdu, China

This work was performed in:

Circuits and Systems Group
Department of Microelectronics & Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics & Computer Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Distributed Coordination for Multi-fleet Truck Platooning"** by **Yikai Zeng B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 22 September 2020

Chairman: ______________________
dr.ir. R.C. Hendriks

Advisors: ______________________
dr. R.T. Rajan

____
dr.ir. M. Wang

Committee Members: ______________________

____

Abstract

Truck platooning refers to coordinating a group of heavy-duty vehicles at a close inter-vehicle distance to reduce overall fuel consumption. This coordination between trucks is traditionally achieved by adjusting the schedule, velocity and routines to increase the platooning chances, and thus improve the overall fuel efficiency. However, the data model built for the coordination problem is typically integer-constrained, making it generally hard to solve. On the other hand, the interaction among self-interested fleets which are operated by different companies is not well-studied. This thesis aims to build a distributed framework for multi-fleet truck platooning coordination to enable the coordination without a third-party service provider. The interaction among fleets is considered a non-cooperative finite game, for which we propose the best response search method, which essentially requires to solve a cooperative truck platooning optimization problem iteratively. We refer to the optimization problem as a best-response problem, which is formulated as a mixed-integer linear problem with relaxation skills. To achieve a feasible time complexity for the best-response subproblem, we propose a decentralized algorithm, distributing the computational load to connected automated vehicles within the fleet. The proposed method is examined under a real-world featured demand set to compare the performance in optimality and time complexity with previous studies. The result suggests that the decentralized algorithm delivers the optimal objective value in this case, while the best-response search does not deliver extra benefits as the dominating time costs in the cost functions eliminate the potential for improvement.

Acknowledgments

At the very outset of this thesis, I would like to express my gratitude to those who helped me through the project.

Foremost, I would like to thank the chairperson and my master coordinator, dr.ir. R.C. Hendriks for giving me the opportunity to chose this topic for my graduation thesis. I would like to appreciate both my daily supervisory, dr. R.T. Rajan and dr. M. Wang for your patient guidance and insightful feedback throughout the progress. It is impossible to explore this new topic without dr. M. Wang revealing the concerns and hot spots in the truck platooning. It is of equal importance that dr. R.T. Rajan leading me overcome different obstacles along the way.

There are some researchers I would like to mention for their kind help along the way. I would like to thank Q. Li for the kind help and guidance. I would like to thank prof.dr. Y. Yin and M. Abdolmaleki for their inspiring work and help on the modelling problem and open attitude. Also I would like to thank dr.ir. M.A. de Bok, L.Xiao and prof.dr.ir. L.A. Tavasszy from Civil Engineering, TU Delft for their kind support.

I would also give my sincere appreciation for the supporting staff of CAS, EEMCS. Thank you very much for the the streamlined graduation process.

Last but not least, I would like to express my great appreciation for my family and friends for supporting me in these years.

Yikai Zeng B.Sc.
Delft, The Netherlands
22 September 2020

Contents

List of Figures

List of Tables

Introduction 1

1.1 Background

Heavy-duty vehicles (HDVs) emit around 5% of the total carbon emissions in the world, and therefore there is an ever increasing demand for fuel efficiency of HDVs. [5]. In addition to ecological effects, one-third of the overall operation cost of a HDV is fuel consumption, which is expensive [6]. Therefore, it is beneficial both environmentally and economically to develop mechanisms to increase the fuel efficiency in HDVs operations.

Truck platooning refers to controlling a group of heavy-duty vehicles at a close inter-vehicle distance [7]. The overall air drag of the platoon is decreased, improving the average fuel efficiency of the platoon. Multiple studies reveal the potential in fuel-saving in different scenarios. Road tests reveal that a 13% of energy saving at a 10 m gap between trucks and a 18% saving at a 4.7 m gap [8]. A more recent research suggests that the fuel saving may reach 14.2% at maximum [9]. An extensive overview of the state of the art in saving potential due to platooning is reported by Zhang et al [10].

Benefits of truck platooning is only viable when platoons are formed. There are several approaches for forming platoons. If there is a sufficient number of trucks operating, it may be possible to form a platoon with encountering peer trucks, known as the opportunistic platooning [11]. However, given the low percentage of truck traffic in overall traffic composition and the gradual introduction of truck automation, the probability of forming a platoon in an ad hoc way is very low. Therefore, it is necessary to coordinate different trucks to form more platoons by synchronizing their departure times, routes, and velocities, which are parameters of trucks' traveling plans. The goal of truck platooning coordination problem is to maximize the platooning benefits by adjusting trucks' traveling plans. This thesis aims to maximizing the benefits of freight transportation stake holders.

1.2 Literature Review: Truck Platooning Coordination

In this section, we introduce a series of studies aiming at the coordination problem. An overview table of the literature review is given in Table 1.1. We refer readers for more details on the state-of-the-art of truck platooning coordination to [11]. Truck platooning coordination can be classified under various categories. The major classification is introduced as followed.

Cooperative v.s. Non-cooperative: Cooperative truck platooning coordination is the scenario in which all trucks are seeking to minimize a global cost function. For example, it is common in logistics companies that cooperative trucks share a set of

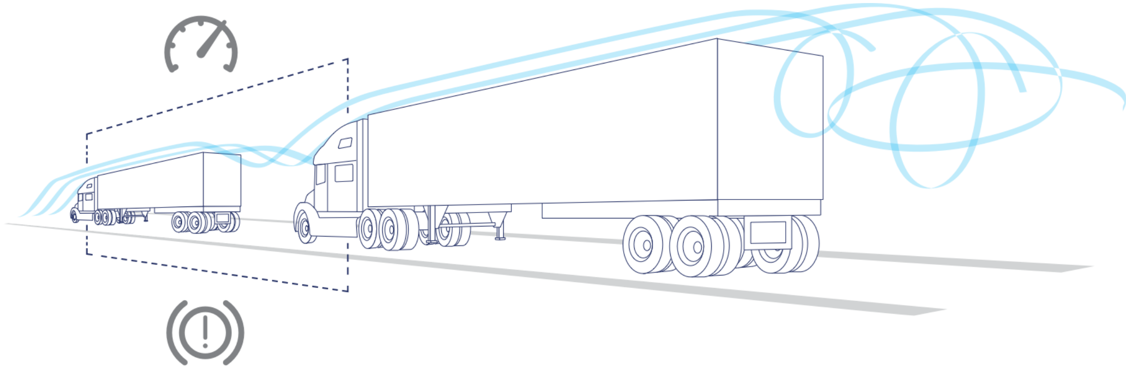

Figure 1.1: A truck platooning illustration [1]

delivery assignments. The trucks are cooperative as they belong to the same owner, and are thus only 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, non-cooperative truck platooning participants concern only on individual cost functions. An individual truck owner is unwilling to obey the plan if it increases his or her plan, even with more overall cost reduction.

Centralized v.s. Distributed: The existence of a centralized node, which gathers demands from each truck and solves the planning problem, is the feature of a centralized truck platooning coordination. In a logistics company, trucks' travel plans are assigned by the dispatch center, which is a typical centralized set-up.

On the contrary, in a distributed set-up, trucks exchange information with connected peers only. It is worth mentioning that the decisions are made by the truck itself. If the trucks belong to different companies, there will not be an authority coordinating them. However, with wireless technologies, the trucks in a distributed set-up may exchange information to coordinate themselves.

Scheduled v.s. Real-time: The operation of the freight transportation is defined as the period at which the first truck starts its trip until the last truck finishes its trip. In a scheduled planning case, all the truck plans are announced and fixed prior to the trip. Scheduled methods are typical in the coordination within the same company, where assignments are set before the operation.

Real-time applications allow trucks to announce or even change plans during the operation. Unforeseen changes during the operation may change the cost of the current plan, as a result, a real-time application may make adjustments based on the latest status.

At this stage, the most common way is the scheduled platooning method with a centralized service provider with all agents operating cooperatively. In comparison, there is less research on other classes of coordination. Now we present a more detailed review on some major categories of truck platooning coordination.

Authors(Year)	Objective		Decision			Scenarios	Dynamics		Topology
	Fuel	Others	Schedule	Routes	Speed		Scheduled	Real-time	
Larson et al.(2013) [12]	✓		✓	✓	✓	Cooperative	✓	✓	Centralized
Larson et al.(2014) [13]	✓		✓		✓	Cooperative		✓	Distributed
Larsson et al.(2015) [14]	✓		✓	✓		Cooperative	✓		Centralized
Nourmohammadzadeh and Hartmann(2016) [15]	✓		✓	✓		Cooperative	✓		Centralized
Zhang et al. (2017) [16]	✓	✓	✓			Cooperative	✓		Centralized
Sokolov et al.(2017) [17]	✓		✓			Cooperative	✓		Centralized
Nourmohammadzadeh and Hartmann(2018) [18]	✓		✓	✓	✓	Cooperative	✓		Centralized
Boysen et al.(2018) [19]	✓	✓	✓			Cooperative	✓		Centralized
Johansson et al.(2018) [20]	✓	✓	✓			Non-cooperative	✓		Centralized
Johansson et al.(2019) [21]	✓	✓	✓			Non-cooperative	✓		Centralized
Abdolmaleki et al.(2019) [22]	✓		✓	✓	✓	Cooperative	✓		Centralized

Table 1.1: Truck Platooning Coordination Review. 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 refer to the adjustable parameters tuning for tuning the optimization problem, including schedule, routes and speed. The schedules parameter includes the departure time and temporary parking point for each truck.

1.2.1 Scheduled Cooperative Centralized Coordination

Scheduled cooperative centralized platooning planning was first proposed by [12], in which a fast heuristic is built to solve the problem for large fleets. In this study, the simulation suggests that the fuel savings potential increases as the number of trucks increases in the road network. The results from this study suggest that detouring may save more energy depending on the nature of the graph. Particularly, the existence of paths with similar length benefits to the likelihood of greater savings.

A further study on the computational complexity in coordination heuristics in truck platooning problem is carried out [14]. It is also proven that the platooning coordination problem is 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 in the above studies. The geographical map is represented by a graph, in which each link is associated with a binary decision variable, indicating the presence of the corresponding link in the planning of a truck. The study proposed three heuristic solvers after showing the infeasibility in time complexity for an exact solution solver, which involves a best pair heuristic, a hub heuristic and a local search method. These methods does not guarantee the optimal solution but offering an acceptable sub-optimal solution to achieve a feasible solving elapsed time.

An alternative approach for approximating the optimum is proposed to solve the problem with a fuel-consumption-based objective function with a genetic algorithm [15]. The simulation is set up on a simplified German intercity network with 10-50 trucks. The work is improved with the features that the latest arrival time restriction and speed options are introduced in the model [18]. The model is solved by particle swarm optimization due to the high computational complexity.

Recently, researchers formulated the platooning coordination problem as a mixed-integer non-linear program (MINLP) problem [22]. The planning of trucks are presented by binary decision variables while auxiliary continuous variables are introduced for relaxation methods. The research characterized the cost function as an energy-based piece-wise concave function. After an analysis of the exact solution algorithm, a fast dynamic-programming heuristic is applied to real-world size test cases with significant computational time complexity improvement comparing to previous studies.

With practical approximation heuristics, researchers perform a comparison study between scheduled planning and real-time opportunistic platooning to explore the potential in cooperative platooning planning [17]. An unweighted Hanna grid in the POLARIS platform is the benchmark for comparison with a relatively large scale test set.

Meanwhile, a study about identical path truck platooning was done in [19]. The goal is to show if truck platooning is indeed practical as there is an inefficient process for trucks to emerge a platoon. The conclusion suggests that truck platooning is appreciated if taking other costs into account, such as drivers' wages because it only delivers insignificant savings in fuel.

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

1.2.2 Other Classes of Truck Platooning Coordination

Scheduled Non-cooperative Centralized Coordination: Taking the competitive nature of transportation into account, a non-cooperative case where all trucks start at the same point with different preferred departure time, coordinated by a centralized service provider has been studied in [20]. It is proven when trucks are identical and non-cooperative, the coordination problem is a congestion game. Congestion games are a classic problem which a pure strategic Nash equilibrium can be searched by best response dynamics. The numerical simulation suggests that in a non-cooperative set-up, the overall benefits are less compared to a cooperative case, showing the self-interested property will eliminate the potential in truck platooning, though still improving the fuel efficiency comparing to non-platooning scenarios.

Self-interest trucks would be interested in knowing their exact position in a platoon since no air-drag reduction is available for the leading truck. To address the issues and extend the work in [20], a comparative study on the interaction mechanisms between leading and tailing trucks are carried out in [21]. So far, the studies mentioned in this subsection are the works addressing the self-interested nature in the real world focusing on the truck platooning coordination [20, 21]. However, the model in these studies assume all trucks start at the same node and the only adjustable decision is the departure time. The numerical results may not be representative for potential studies. On the other hand, trucks belonging to the same company may compete with each other, resulting a suboptimal result for the company. Last but not the least, since trucks are non-cooperative, a centralized node is required for coordination. A trusted service provider must exist to enable non-cooperative centralized coordination, which may require extra fees to purchase such service with risks of losing privacy.

Distributed Coordination: The coordination problem based on distributed set-up is not widely carried out. A distributed framework for coordination with the features that multiple coordination controllers reside in different positions of the road network, processing the information from approaching HDVs [13]. In general, the problem is converted into coordinate a subset of trucks to form platoons. The horizon for each local centers is also restricted into a subgraph.

In a distributed set-up, fleets may have to reveal information of interest to each other, which may be a major barrier for real implementation. A secure and private communication framework for two fleet owners is studied in [23], which may be generalized for a distributed coordination mechanism as claimed. The fleet owner may submit encrypted queries to ask for information from peers. Appropriate answers are accessible without knowing the queries. The numerical simulation is based on a distributed set-up. However, the details about optimization process of each fleet owner is not discussed in the work.

1.2.3 Summary

Previous work mainly focused on the scheduled cooperative centralized platooning coordination. Nearly all studies assume there is a trusted service provider [12, 14, 15, 17, 19, 22]. However, destinations, departure times and other information of interest must be announced to the service provider before the operation. Since logistic companies are competitors, it is unlikely that they share a common cost function. Another category of studies model the trucks as non-cooperative peers with simple equilibrium searching solution [20, 21]. The disadvantages with the method are that all trucks must be identical and the cost reduction is less than cooperative scenarios. Distributed planning is also relatively neglected, in which a trusted third party is not required. A communication platform with encryption among fleet owners is designed [23], while further work is not present yet.

On the other hand, all studies mentioned are based on integer programming models, which make the problem generally hard to solve. A review on the algorithm performance is given in Table 1.2. The table compares the solution methods, simulation networks and the fleet size. It is revealed the the exact solution methods in different studies fail to handle a large scale problem, when the complexity of graph or the number of trucks increases. Different approximations are proposed to solve the problem, but limited to centralized method. The existing literature has limitations in three major areas.

Authors(Year)	Solution	Network	Max fleet size	
			Exact	Alternative
Larson et al.(2013) [12]	Exact and Local search heuristic	German autobahn network		8000
Larrson et al.(2015) [14]	Exact and Local search heuristic	German autobahn network	10	200
Nourmohammadzadeh and Hartmann(2016) [15]	Exact and genetic algorithm	Simplified German intercity map	20	50
Sokolov et al.(2017) [17]	Exact	10×10 grid	50	
Zhang et al.(2017) [16]	Exact	Identical path	2	
Nourmohammadzadeh and Hartmann(2018) [18]	Exact and particle swarm optimization	Chicago road network		1000
Abdolmaleki et al.(2019) [22]	Exact and dynamic programming heuristic	Germany highway network	150	1000

Table 1.2: Platooning Planning Review Table - Algorithm

- **Unrepresentative cost functions and limited adjustable decisions**: Most studies focus only on fuel-savings, which is not sufficiently representative. Furthermore, trucks are assumed to be identical for a favorable structure of the formulation in some of studies, which may not be practical for a large fleet. Few works introduce the speed options as an adjustable parameter.

- **Infeasible exact solution method and deterministic performance guarantees**: With integer variables, models proposed so far are generally computationally expensive to solve. Researchers attempt to reduce the time complexity in a variety of approximating methods and stochastic methods. However, most of them are unable to offer a deterministic performance guarantees in the sub-optimal level.

- **Limited truck interaction modeling**: While non-cooperative fleets only concern their own cost, trucks within the same fleet cooperate to minimize the owner's cost. For a truck, there are both cooperative and non-cooperative peers. The hybrid relationship is not yet adequately modeled.

1.3 Goals

The goal of this thesis is to develop a best-representative model for truck platooning coordination problem, aiming at the limitations mentioned in Section 1.2.3. The main goals are,

- A framework for solving truck platooning coordination as a non-cooperative coordination problem for self-interested fleets.
- A mathematical model for cooperative truck platooning, in which cost functions consist of different cost other than simply fuel cost and while routes, speed, types of trucks and schedule for trucks are decision variables to optimize.
- A decentralized algorithm for solving the multi-fleet truck platooning coordination problem.
- A simulation with real-world featured demand input to test the performance of the decentralized algorithm.

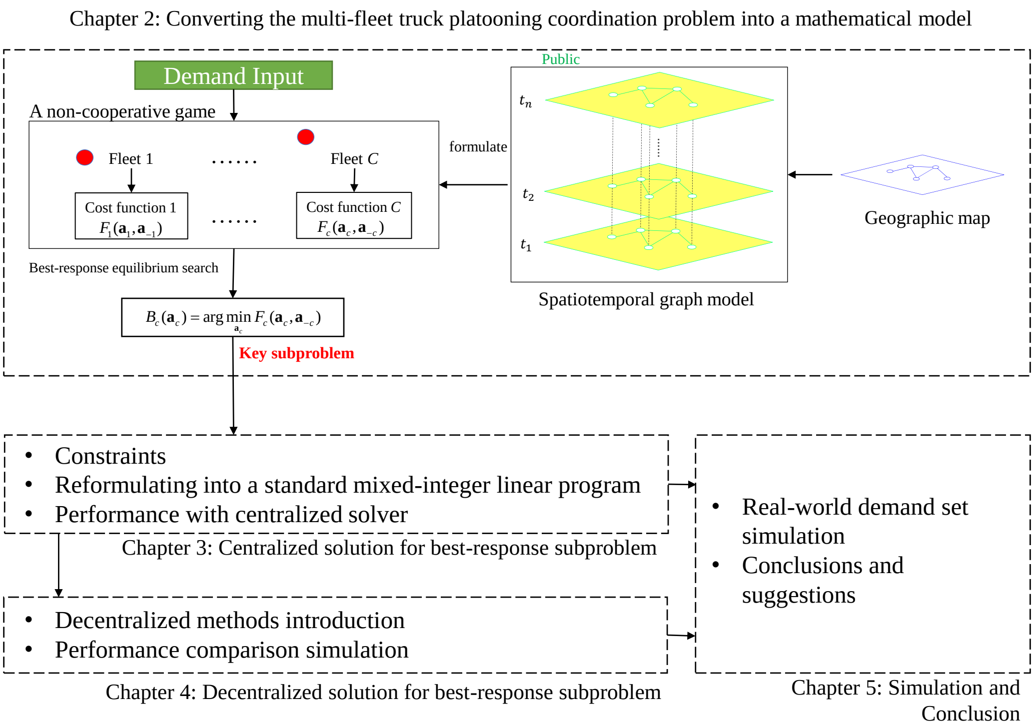

Figure 1.2: A brief schematic figure for thesis outline

1.4 Outline

The rest of the paper is organized in the structure shown in Figure 1.2. Chapter 2 presents our approach for modeling the relationship among trucks before the formulation of the cost function. The geographical map and the demand of all trucks are converted into a mathematical presentation for formulating the cost function for each fleet in a non-cooperative game. Based on the proposed game, the best-response search's critical challenge is to solve the optimization problem as in the block efficiently. In Chapter 3, we reformulate the problem to preserve a linear structure of the problem along with test numerical examples with a centralized exact solution solver. The formulation process includes regularizing constraints and applying relaxation methods for a favorable formation. Chapter 4 introduces the details on the decentralized algorithm and numerical examples, examining the performance of the proposed algorithm. Chapter 5 introduces the real-world input set for the large scale test and concludes the thesis.

Problem Formulation 2

In this chapter, we formulate the truck platooning coordination problem. A discrete spatio-temporal road network is constructed, enabling the representation of truck traveling plans with binary decision vectors. Consequently, the plan of a fleet consists of all trucks' plans. In this discrete presentation, the interaction among all fleets is shown to be a non-cooperative finite game. We introduce an algorithm for a steady outcome of the game and the details of each player's cost function.

2.1 Preliminaries

The general symbols and operators in this thesis are explained as followed.

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
$\mathbf{1}_n, \mathbf{0}_n$:	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 non-scalars
$\leq, \geq$:	components-wise inequalities for non-scalars
$\in$:	belong to

2.2 Spatio-temporal Road Network

We adopt the notation in [22] with minor modifications. The physical road network is given with $\mathcal{S}$ as the set of all physical nodes $\mathcal{S} = \{1, 2, \cdots, S\}$, where $s_i \in \mathcal{S}$ is the i*th* ordered physical node. A physical node may be a parking lot, a city, or a point on the road. The time dimension is discretized into unitary intervals of length δ. Let $\mathcal{T}$

denotes a set of the time dimension, where $\mathcal{T} = \{1, 2, \cdots, T\}$ and let $t_i \in \mathcal{T}$ be the ith ordered time interval. The spatio-temporal 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 $\mathcal{N}(\mathcal{G})$. Following this definition, the set of all links in $\mathcal{G}$ is denoted by $\mathcal{E}(\mathcal{G})$. An example of $\mathcal{G}$ is shown in Figure 2.1. A link $l \in \mathcal{E}(\mathcal{G})$

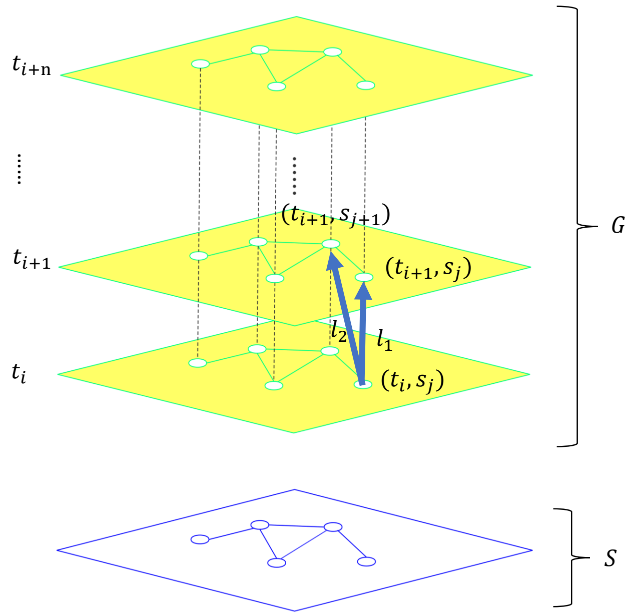

Figure 2.1: In the example $\mathcal{G}$, l_1 represents a vehicle waits at s_j from time t_i to t_{i+1}. l_2 represents leaving s_j at t_i and arriving at s_{j+1} at t_{i+1}

between two nodes, so that $l = (t_i, s_i, t_j, s_j)$ means a truck leaves s_i at time t_i to arrive s_j at time t_j if there is a physical link between s_i and s_j or $s_i = s_j$. Due to the physical limitation that a truck can not travel back in time, t_i is always smaller than t_j. Let L be the amount of links in $\mathcal{E}(\mathcal{G})$. All information about $\mathcal{G}$ is assumed to be known by all trucks.

2.3 Individual Truck Path and Fleet Plan

We assume there are $K \in \mathcal{K}$ trucks planning to operate on $\mathcal{G}$. Let $\mathcal{C} = \{1, \cdots, C\}$ be the set of all related fleets. A binary decision variable is given by $x_{k,l}$, which indicates whether k transverses on route l as

$$x_{k,l} = \begin{cases} 1 & \text{Truck } k \text{ traverses through link } l \\ 0 & \text{Otherwise} \end{cases} \tag{2.1}$$

Stacking up all decision variables of truck k in a link order, a vector $\bar{\mathbf{x}}_k \in \mathbb{R}^L$ is used to represent the path of truck k,

$$\bar{\mathbf{x}}_k = [x_{k,1}, x_{k,2}, \cdots, x_{k,L}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}) \tag{2.2}$$

Clearly, with physical limitations, a considerable amount of links in $\mathcal{G}$ is infeasible for truck k to transverse with operating time limitation and limited speed options. For the sake of efficiency, a refining process is applied to $\mathcal{G}$ as proposed in [24]. For all trucks $k \in \mathcal{K}$, each truck 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 $t_k^{LA} \in \mathcal{T}$. Let $\mathcal{G}_k$ be the sub-graph 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 is given in Figure 2.2. Nodes in red area is infeasible because a truck is impossible to arrive the destination in time once it travels to one of these nodes. Besides, nodes in the green area represents those that are too far from the origin to reach because of the speed limitation. The red Let L_k be the number of links in $\mathcal{G}_k$. The decision vector is refined as

$$\mathbf{x}_k = [x_{k,1}, x_{k,2}, \cdots, x_{k,L}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}_k). \tag{2.3}$$

A fleet is a subset of trucks. For the cth fleet, $\mathcal{K}_c$ is a set of K_c trucks and belongs to

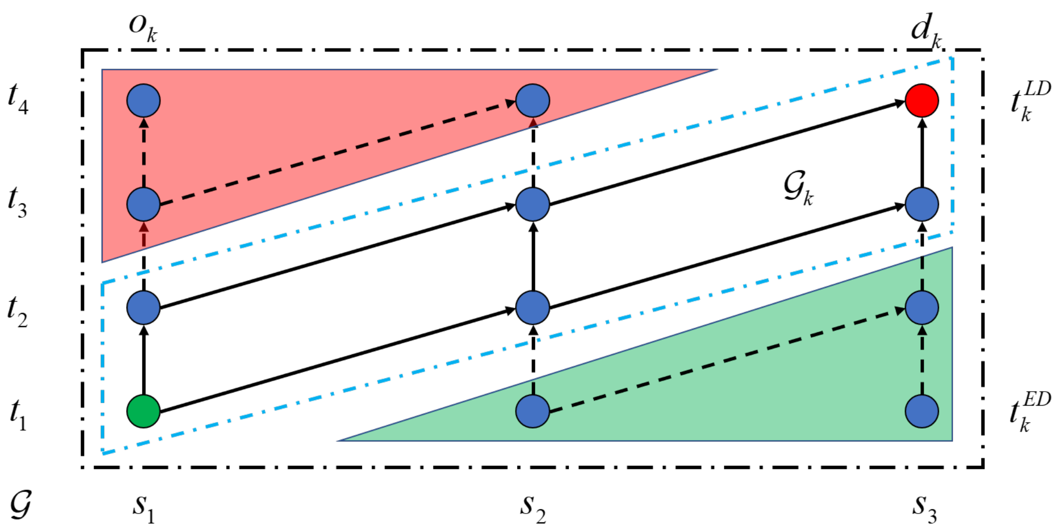

Figure 2.2: The example graph is a spatio-temporal extension of a one-dimension geographical map from $s1$ to $s3$, which is on the horizontal axis. The vertical axis represents the time dimensions at each interval. Two kinds of infeasible nodes in $\mathcal{G}$ are removed. The upper left red triangle area refers to the nodes that are not connected to the spatio-temporal destination (red). The green areas are removed likewise. The remaining $\mathcal{G}_k$ is in the blue rectangle.

fleet c, where $\mathcal{K}_c \subseteq \mathcal{K}$. Consequently, let $\mathbf{a}_c \in \mathbb{Z}^{L_c}$ denote the path of all trucks in fleet c

$$\mathbf{a}_c = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_{K_c}^T]^T \quad \forall k \in \mathcal{K}_c \tag{2.4}$$

2.4 A Non-cooperative Game Model among Fleets

To address the competitive behavior among fleets, we consider the interaction among all fleets $c \in \mathcal{C}$ a non-cooperative game Γ. As shown in Figure 1.2, based on the spatio-temporal model, the cost of a fleet may be formulated, which is to be introduced in Section 2.4.2. It is obvious that the cost of a fleet is not only determined by its plan, but also the plan of other fleets if they agree to merge to save costs. Although the fleets are competitive and have self-interest, they may merge to form platoons for multilateral benefits, which is also known as multi-fleet truck platooning [20]. The interaction in the multi-fleet truck platooning possess the typical structure of a non-cooperative game, in which a set of players aiming to maximize its benefits by changing strategies while the costs are affected by the strategy selection of all other players [25]. The structure is commonly seen in the area of game theory so that it is of interest to apply game theoretic approaches in the problem of truck platooning coordination as in the previous work [20, 21].

To rigorously define and denote such game of multi-fleet truck platooning coordination, we consider each fleet c has an action set $\mathcal{A}_c$ which contains all the feasible $\mathbf{a}_c$. $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_C$ is defined as the action profile of the game, where $\times$ is the Cartesian product. The other piece in the game is known as the pay-off function or the utility function, which represents the value the corresponding player aims to maximize. Consequently, the pay-off function in the multi-fleet truck platooning problem is simply the negative cost function. Let the cost function of the cth fleet be $F_c : \mathcal{A} \to \mathbb{R}$. The details of the cost function are introduced in Section 2.4.2. The ideal outcome of the game is that the game reaches a steady state, which is known as Nash equilibrium. Let $\mathbf{a}_c^*$ be the selection of plan for fleet c at the equilibrium and

$$\mathbf{a}_{-c} = \mathbf{a}_1 \times \cdots \times \mathbf{a}_{c-1} \times \mathbf{a}_{c+1} \times \cdots \times \mathbf{a}_C, \tag{2.5}$$

which is the plan selection of all fleets other than c. Likewise, let $\mathbf{a}_{-c}^*$ be the selection of all other fleets at the equilibrium. The equilibrium is a selection from $\mathcal{A}$ that

$$-F_c(\mathbf{a}_c^*, \mathbf{a}_{-c}^*) \geq -F_c(\mathbf{a}_c, \mathbf{a}_{-c}^*) \quad \forall c \in \mathcal{C} \tag{2.6}$$

where $\mathbf{a}_c^*$ and $\mathbf{a}_{-c}^*$ denote the equilibrium action taken by fleet c and other fleets. With such a selection of plans, no fleet may benefit more by alternating its plan. It is thus guaranteed that all cost-oriented fleets will keep the selection.

However, there are some critical properties of the game worth mentioning. In the assumption of this thesis, the fleet always seeks to maximize its benefits, implying that once current $\mathbf{a_{-c}}$ is known to c, the $\mathbf{a}_c$ is determined. The certainty in the strategy makes the game a pure strategic game and the equilibrium is named as pure strategic Nash equilibrium (PSNE). The existence of PSNE is not guaranteed and generally difficult to search in a multi-player game. On the other hand, since the selection of $\mathbf{a}_c$ is integer-constrained, it is unlikely to prove the existence with the fixed-point theorem with high non-convexity.

Last but not least, the proposed solution should be decentralized or distributed as there is assumed to be no service provider. The majority of the current searching algorithm requires that the complete pay-off matrix is available to all players, making them undesirable in this application [26]. In the next subsection, we would offer more details in our proposed method in searching for the PSNE, which is the best-response searching algorithm.

2.4.1 Best-response Searching Algorithm

In a best-response search, every player makes the most favorable decision given other players' strategies, which is to minimize the cost.

$$B_c(\mathbf{a}_{-c}) = \arg\min_{\mathbf{a}_c} F_c(\mathbf{a}_c, \mathbf{a}_{-c}) \tag{2.7}$$

$B_c(\cdot)$ is also known as the best-response function of fleet c. Fleets sequentially update the strategies with the best-response function assuming other peers' strategies are fixed. The process is shown in Figure 2.3. At the starting point of the algorithm, all fleets perform an optimization without considering other fleets and the results are assumed to be known to all fleets. For the first fleet, under the assumption that the plan of other fleets $\mathbf{a}_{-c}$ is fixed, the new plan $\mathbf{a}_c$ is given by the best-response function as shown in the blocks in Figure 2.3. If there is no PSNE in the game, or the iteration gets stuck

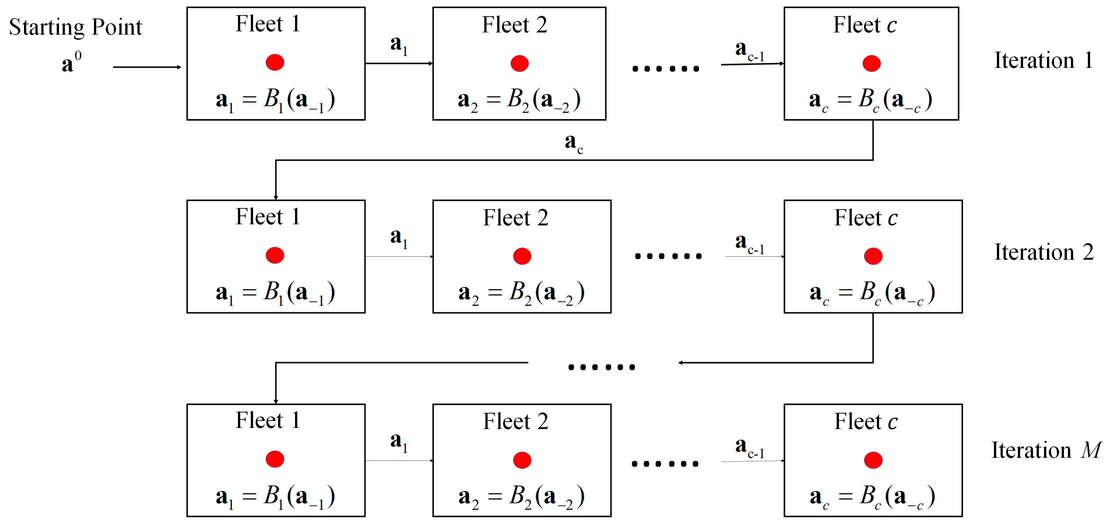

Figure 2.3: Proposed Best-response Search Algorithm. At iteration M, in which no fleet is able to improve the strategy, the desired outcome is assumed to be reached.

in a loop, the proposed approach will not reach a steady state. We proposed to set an upper bound on the rounds of iterations. If all fleets fail to agree on PSNE, all fleets shall return to the solo optimum. As a result, every fleet neglects the risk of the deficit.

The critical subproblem now is the cost function, we will then introduce the detailed formulation in the next subsection.

2.4.2 Cost function of the Fleet

The cost of a fleet is consequently sum of costs of each truck. The cost of a truck's trip is composed of fuel cost, travel time cost and schedule preference penalty [16].

2.4.2.1 Fuel Cost

Truck platooning saves only on fuel cost since it reduces overall air drag for the platoon. Since the platoon's leading truck has no savings, to simplify the problem, a mechanism to balance the savings is assumed to be applied, such that all vehicles taking the leading position sequentially. We assume all trucks in the platoon has the same average air drag reduction factor φ during the platooning trip. Let φ_l be described as a function of the amount of trucks N_l on the same link $l \in \mathcal{E}(\mathcal{G})$, which is given by

$$\varphi_l = \varphi_o - \frac{\varphi_o}{N_l} \tag{2.8}$$

where φ_o is the air reduction factor for all tailing trucks, which is assumed to be constant and equal for all trucks during the platooning. φ_l is the average air reduction factor on the link l.

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 $T_l = t_j - t_i$. The fuel cost $\tilde{f}_{k,l}$ on certain link l for truck k is given by (2.9)

$$\begin{aligned} &\tilde{f}_{k,l}(\varphi_l) = \\ &\frac{w_f \xi}{\kappa}\left(\mu_k E_k V_k T_l + 0.5 \frac{c_{d,k} \rho A_k (1-\varphi_l)}{1000 \varepsilon_k \varpi_k} \frac{L_l^3}{T_l^2} + \frac{M_k g (\sin\theta_l + c_{r,k}\cos\theta_l)}{1000\varepsilon_k \varpi_k} L_l\right) \end{aligned} \tag{2.9}$$

where physical constants are

w_f	Price of oil
ξ	Fuel-to-mass ratio
κ	Heating value of the fuel
ρ	Density of air
g	Gravitational constant

and vehicle-dependent $(\cdot)_k$ or link-dependent $(\cdot)_l$ parameters

θ_l	Road gradient
E_k	Engine speed
L_l	Distance between the physical nodes
T_l	Engine working time spent on the link
μ_k	Engine friction factor
V_k	Engine displacement
$c_{d,k}$	Coefficient of aerodynamic drag
A_k	Front area of the truck
$c_{r,k}$	The coefficient of rolling resistance
M_k	Total vehicle weight
ε_k	Drive train efficiency
ϖ_k	An efficiency parameter for the engine

The only unknown parameter in 2.9 is φ_l, for a given link l and truck k, and therefore (2.9) may be rewritten as

$$\tilde{f}_{k,l}(\varphi_l) = C_{k,l,f_1}\varphi_l + C_{k,l,f_2} \tag{2.10}$$

where

$$C_{k,l,f_1} = -0.5\frac{\xi}{\kappa}\frac{c_{d,k}\rho A_k L_l^3}{1000\varepsilon_k\varpi_k T_l^2} \tag{2.11}$$

$$C_{k,l,f_2} = \frac{\xi}{\kappa}\left(\mu_k E_k V_k T_l + 0.5\frac{c_{d,k}\rho A_k}{1000\varepsilon_k\varpi_k}\frac{L_l^3}{T_l^2} + \frac{M_k g(\sin\theta_l + c_{r,k}\cos\theta_l)}{1000\varepsilon_k\varpi_k}L_l\right) \tag{2.12}$$

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 certain physical node. We assume the vehicle is shutdown and there is no fuel cost. Subsequently, we define $\mathcal{W}(\mathcal{G})$ as a set of all links l, for which $L_l = 0, T_l = 0$. Consequently,

$$\tilde{f}_{k,l}(\varphi_l) = 0 \quad \forall l \in \mathcal{W}(\mathcal{G}) \tag{2.13}$$

Substituting (2.8) into (2.10), and since there is no fuel cost if there is no truck,

$$\tilde{f}_{k,l}(\varphi_l) = \check{f}_{k,l}(N_l) = \begin{cases} \frac{a_{k,l}}{N_l} + b_{k,l} & N_l \geq 1 \\ 0 & N_l = 0 \end{cases} \tag{2.14}$$

where N_l is the amount of trucks traversing on l. In (2.14), we convert the a function from φ_l to the cost,$\tilde{f}_{k,l}(\varphi_l)$, to a function from N_l to the cost, $\check{f}_{k,l}(N_l)$.

$$a_{k,l} = \varphi_o C_{k,l,f_1} \tag{2.15}$$

$$b_{k,l} = C_{k,l,f_2} - \varphi_o C_{k,l,f_1} \tag{2.16}$$

An example of the function $\tilde{f}_{k,l}(N_l)$ is shown in Figure 2.4. Let us denote $\eta_l(\cdot) : \mathcal{A} \to \mathbb{R}$, for every link l,

$$N_l = \eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) \tag{2.17}$$

Consequently, $f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c})$ is given by

$$f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \begin{cases} \frac{a_{k,l}}{\eta_l(\mathbf{a}_c, \mathbf{a}_{-c})} + b_{k,l} & \eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) \geq 1 \\ 0 & \eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) = 0 \end{cases} \tag{2.18}$$

(2.17) is for now the place holder for the method to get the amount of trucks from both the own fleet and other fleets. (2.18) is the cost function of truck k traversing on link l affected by both $\mathbf{a}_c$ and $\mathbf{a}_{-c}$.

2.4.2.2 Travel Time Cost and Schedule Preference Penalty

The real-world stake-holder of freight transportation concerns other cost as well as the fuel. The process of forming platoons may increase time-related costs, including travel time cost and schedule preference penalty. With these two pieces in the cost function, the model is more representative. The first piece is the travel time cost including wages for drivers. It is assumed that every unit of time has equal costs, thus the travel time cost of truck k is given by,

$$\tilde{g}_k(\mathbf{x}_k) = w_{t,k}\mathbf{t}_k^T\mathbf{x}_k \tag{2.19}$$

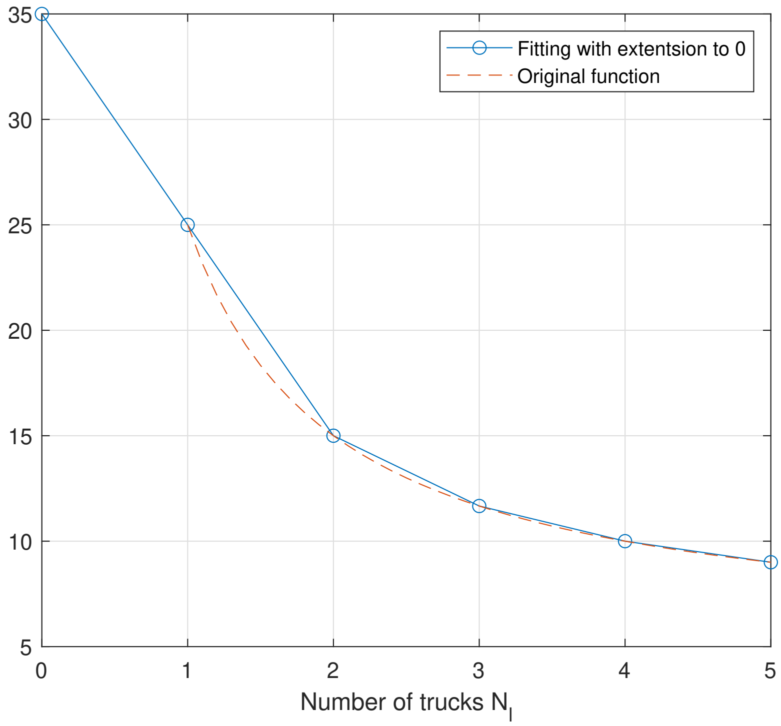

Figure 2.4: An example of $\tilde{f}_{k,l(N_l)}$ and a piece-wise fitting example

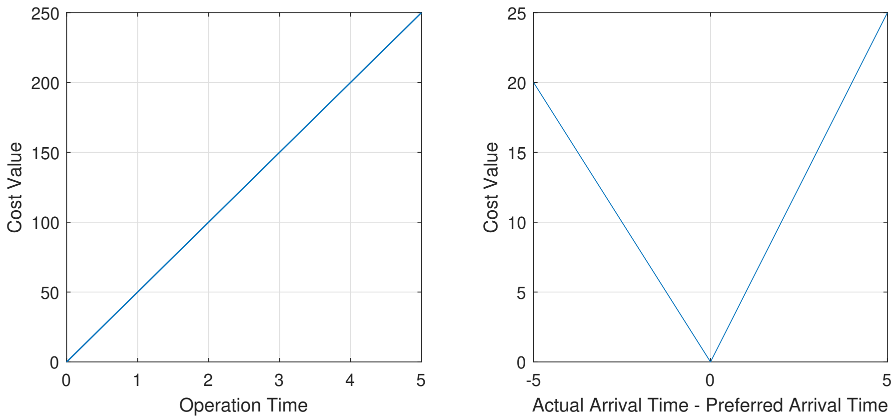

Figure 2.5: Example travel time cost (left) and schedule preference penalty (right)

where $w_{t,k}$ is a weight for travel time cost for truck k. $\mathbf{t}_k \in \mathbb{R}^{L_k}$ is a selection vector which selects decision variables that present the links that present the travelling time of the truck k. Let $t_{k,i}$ be the ith element in $\mathbf{t}_k$, which corresponding to link l and truck,

$$t_{k,i} = \begin{cases} 0 & l = (t_n, d_k, t_m, d_k) \\ 1 & \text{Otherwise} \end{cases} \tag{2.20}$$

where $d_k \in \mathcal{S}$ is the geographical destination of truck k. An example of travel time cost is given in Figure 2.5. In other respects, the schedule preference penalty represents the preferred arrival time of the truck owner. The industry of freight transportation prefers to eliminate the uncertainty in the arrival time. It is given for truck k by

$$\tilde{h}_k(\mathbf{x}_k) = \max\{w_{k,l}(\mathbf{t}_k^T\mathbf{x}_k + t_k^{ED} - t_k^{PA}), -w_{k,e}(\mathbf{t}_k^T\mathbf{x}_k + t_k^{ED} - t_k^{PA})\} \tag{2.21}$$

where, for truck k,

$$\begin{array}{ll} w_{k,l} & \text{Positive weight for later than schedule preference penalty} \\ w_{k,e} & \text{Positive weight for earlier than schedule preference penalty} \\ t_k^{ED} & \text{Earliest departure time} \\ t_k^{PA} & \text{Preferred arrival time} \end{array} \tag{2.22}$$

The weights for early and late arrival may vary since the stake-holder may value them differently. An example of schedule preference example is also given in Figure 2.5. The cost function $F_c : \mathcal{A} \to \mathbb{R}$ is a function of $\mathcal{A}$, therefore (2.19) and (2.21) requires reformulation. Based on (2.4), a selection matrix $\mathbf{S}_k$ of the proper size that

$$\mathbf{x}_k = \mathbf{S}_k\mathbf{a}_c \tag{2.23}$$

where $\mathbf{x}_k$ is the decision vector for truck k and $\mathbf{a}_c$ is the decision vector of fleet c, defined in (2.3) and (2.4). To convert (2.19) into a function of $\mathbf{a}_c$, we substitute (2.23) into (2.19)

$$\tilde{g}_k(\mathbf{x}_k) = g_k(\mathbf{a}_c) = w_{t,k}\mathbf{t}_k^T\mathbf{S}_k\mathbf{a}_c \tag{2.24}$$

and likewise for (2.21)

$$\tilde{h}_k(\mathbf{x}_k) = h_k(\mathbf{a}_c) = \max\{w_{k,l}(\mathbf{t}_k^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}), -w_{k,e}(\mathbf{t}_k^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA})\} \tag{2.25}$$

2.4.2.3 Overview

The goal of Section 2.4.2 is to formulate $F_c : \mathcal{A} \to \mathbb{R}$, we now combine the content in the above-mentioned subsections. The cost function of a truck is given by summing up all pieces on all links, the cost function of an individual truck is given by

$$f_k(\mathbf{a}_c, \mathbf{a}_{-c}) = \sum_{l \in \mathcal{E}(\mathcal{G}_k)} f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) + g_k(\mathbf{a}_c) + h_k(\mathbf{a}_c) \tag{2.26}$$

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

$$F_c(\mathbf{a}_c, \mathbf{a}_{-c}) = \sum_{k \in \mathcal{K}_c} \left[\sum_{l \in \mathcal{E}(\mathcal{G}_k)} f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) + g_k(\mathbf{a}_c) + h_k(\mathbf{a}_c) \right] \tag{2.27}$$

where

$$\begin{array}{rl} \mathbf{a}_c & \text{The planning of fleet } c \text{ defined in (2.4)} \\ \mathbf{a}_{-c} & \text{The planning of all other fleets other than } c \text{ defined in (2.5)} \\ f_{k,l}(\cdot) & \text{Fuel cost of truck } k \text{ on link } l \text{ defined in (2.18)} \\ g_k(\cdot) & \text{Time cost of truck } k \text{ defined in (2.24)} \\ h_k(\cdot) & \text{Preference penalty of truck } k \text{ defined in (2.25)} \end{array}$$

The proposed method is to iteratively compute the best-response function $B_c(\mathbf{a}_{-c})$, which is

$$B_c(\mathbf{a}_{-c}) = \arg\min_{\mathbf{a}_c} \quad F_c(\mathbf{a}_c, \mathbf{a}_{-c}) \tag{2.28}$$

The goal is to find an optimal $\mathbf{a}_c$ with fixed $\mathbf{a}_{-c}$ to minimize the cost of the fleet c. However, both $f_{k,l}(\cdot)$ and $h_k(\cdot)$ are not continuous and thus not differentiable. In terms of linearity, $f_{k,l}(\cdot)$ is not linear while $g_k(\cdots)$ and $h_k(\cdots)$ possess favorable linear forms. The problem lacks a structure, preventing it from being readily solved by any off-the-shelf solvers.

In comparison with previous studies, the influence of other fleets are taken into consideration in the formulation. In addition, we also consider the decision variables involve velocity, routes and schedule. Cost consists of fuel, time and schedule preference, which are not taken into consideration in the previous truck platooning coordination studies. In the following chapters, we discuss various methods to reformulate and efficiently solve the subproblem.

Centralized Method for Best-response Subproblem

3

In this chapter, the subproblem of finding the best response of a fleet is explained and reformulated. There are two sets of constraints which are applied to the original problem. The first one is an equality constraint set, guaranteeing a feasible path for the truck. The other constraint is based on the binary feature defined for decision variables. The non-linearity and discontinuity in the cost function are challenges, which are handled with relaxation methods. An overview of the relaxed problem is given at the end of this chapter.

3.1 Original Problem

Each fleet attempts to minimize its own cost as described in (2.27) while limited by two sets of constraints. The first set of constraints guarantees that the solution is a feasible path in $\mathcal{G}$ and the truck obeys the latest arrival demand. The constraint is formulated based on the multi-commodity network flow problem [27]. The flow conservation constrains are written as

$$\sum_{(t_i,s_i):l=(t_i,s_i,t,s)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} - \sum_{(t_j,s_j):l=(t,s,t_j,s_j)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} = d^k_{(t,s)} \tag{3.1}$$

where

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

and $x_{k,l}$ is the binary decision variable as described in (2.1),

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

We convert the formulation into a matrix form to simplify the notation,

$$\mathbf{A}_k = \left[\mathbf{e}^T_{(t_1,s_1)}, \cdots, \mathbf{e}^T_{(t_i,s_i)}\right]^T, \quad \forall (t,s) \in \mathcal{G}_k \tag{3.2}$$

where $\mathbf{e}_{(t,s)}$ is a vector of length L_k, which is the number of links in $\mathcal{G}_k$. It is selected in a way that

$$\mathbf{e}^T_{(t,s)}\mathbf{x}_k = \sum_{(t_i,s_i):l=(t_i,s_i,t,s)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} - \sum_{(t_j,s_j):l=(t,s,t_j,s_j)\in\mathcal{E}(\mathcal{G}_k)} x_{k,l} \tag{3.3}$$

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

$$\mathbf{b}_k = [d^k_{(t_i,s_i)}, \cdots, d^k_{(t_i,s_i)}]^T, \quad \forall (t,s) \in \mathcal{G}_k \tag{3.4}$$

In this way, (3.1) is rewritten as

$$\mathbf{A}_k \mathbf{S}_k \mathbf{a}_c = \mathbf{b}_k \quad \forall k \in \mathcal{K}_c \tag{3.5}$$

where $\mathbf{S}_k$ is a selection matrix that satisfy (2.23). By assigning (t_k^{ED}, d_k) as the spatio-temporal destination, the path is guaranteed to terminate at the node before a certain time period. The other constraint is simply followed the definition of binary decision variable $x_{k,l}$, which is defined in (2.1). Combining the above-mentioned constraints, the best-response subproblem, denoted as $\boldsymbol{\mathcal{P}}$, is given in

$$\begin{aligned}
\min_{\mathbf{a}_c} \quad & F_c(\mathbf{a}_c, \mathbf{a}_{-c}) = \sum_{k \in \mathcal{K}_c} \left[\sum_{l \in \mathcal{E}(\mathcal{G})} f_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) + g_k(\mathbf{a}_c) + h_k(\mathbf{a}_c) \right] \\
& \text{subject to} \quad \mathbf{A}_k \mathbf{S}_k \mathbf{a}_c = \mathbf{b}_k \quad \forall k \in \mathcal{K}_c \\
& \qquad\qquad\quad\;\; \mathbf{a}_c \in [0, 1]^{L_c}
\end{aligned} \tag{$\boldsymbol{\mathcal{P}}$}$$

where $\mathbf{a}_{-c}$ is considered as constant

$$\mathbf{a}_c = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_{K_c}^T]^T \quad \forall k \in \mathcal{K}_c$$

as in the (2.4). $L_c = \sum_{k \in \mathcal{K}_c} L_k$, which is the sum of all links in all $\mathcal{G}_k$.

3.2 Reformulation to Preserve Standard MILP Structure

The cost function in $\boldsymbol{\mathcal{P}}$ lacks structure as $f_{k,l}$ is discontinuous though the remaining part is linear. In this section, we now discuss the details on applying relaxations to preserve the linear form of the problem.

3.2.1 LP Relaxation on the Schedule Preference Penalty

$h_k(\mathbf{a}_c)$ falls into a typical form for LP relaxation. As in (2.21),

$$h_k(\mathbf{a}_c) = \max\{w_{k,l}(\mathbf{t}_k^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA}), -w_{k,e}(\mathbf{t}_k^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA})\}$$

which is the schedule preference penalty of truck k. An auxiliary parameter $p_k \in \mathbb{R}$ is introduced as in

$$w_{k,l}(\mathbf{t}_k^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA}) \leq p_k \tag{3.6}$$

$$-w_{k,e}(\mathbf{t}_k^T \mathbf{S}_k \mathbf{a}_c + t_k^{ED} - t_k^{PA}) \leq p_k \tag{3.7}$$

3.2.2 Piece-wise Fitting and Relaxation for the Fuel Cost Function

The major challenge in reformulating the problem is that the fuel cost function is discontinuous between $N_l = 0$ and $N_l = 1$, as previously shown in (2.14), which we restate here

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

where $a_{k,l}$ and $b_{k,l}$ are the constants depending on the truck and the link as described in (2.15) and (2.16). Since N_l is a non-negative integer by definition, a piece-wise fitting is proposed as in Figure 2.4 without affecting the objective value on all positive N_l feasible points. 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

$$\alpha_{k,l,j} = \frac{a_{k,l}}{j+1} - \frac{a_{k,l}}{j} \tag{3.8}$$

$$\beta_{k,l,j} = \frac{a_{k,l}}{j} + b_{k,l} - j(\frac{a_{k,l}}{j+1} - \frac{a_{k,l}}{j}) \tag{3.9}$$

where $j = 1, 2, \cdots, J$. The piece-wise fitting function is then with (2.17) substituted in,

$$\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \max\{\alpha_{k,l,j}\eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) + \beta_{k,l,j}\} \quad j = 1, 2, \cdots, J \tag{3.10}$$

$\eta_l(\mathbf{a}_c, \mathbf{a}_{-c})$, first introduced in (2.17), is a linear function given by

$$\eta_l(\mathbf{a}_c, \mathbf{a}_{-c}) = \sum_{k \in \mathcal{K}_c} x_{k,l} + N_{l,-c} \tag{3.11}$$

where $N_{l,-c}$ is treated as a constant based on $\mathbf{a}_{-c}$. (3.10) 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 (3.11) into (3.10),

$$\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \max\{\alpha_{k,l,j}(\sum_{k' \in \mathcal{K}_c \setminus k} x_{k',l} + x_{k,l}) + N_{l,-c}) + \beta_{k,l,j}\} \quad j = 1, 2, \cdots, J \tag{3.12}$$

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

$$x_{k,l}\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) = \begin{cases} \hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) & x_{k,l} = 1 \\ 0 & x_{k,l} = 0 \end{cases} \tag{3.13}$$

An auxiliary $u_{k,l}$ is introduced for the relaxation of the fuel cost as in

$$x_{k,l}\hat{f}_{k,l}(\mathbf{a}_c, \mathbf{a}_{-c}) \leq u_{k,l} \tag{3.14}$$

Applying point-wise maximum method,

$$\alpha_{k,l,j}x_{k,l}^2 + \alpha_{k,l,j}x_{k,l} \sum_{k' \in \mathcal{K}_c \setminus k} x_{k',l} + (\alpha_{k,l,j}N_{-c,l} + \beta_{k,l,j})x_{k,l} \leq u_{k,l} \tag{3.15}$$

where $j = 1, \cdots, J$. 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

$$(\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c,l} + \beta_{k,l,j})x_{k,l} + \alpha_{k,l,j}v_{k,l} \leq u_{k,l} \quad j = 1, \cdots, J \tag{3.16}$$

with

$$v_{k,l} \leq (K_c - 1)x_{k,l} \tag{3.17}$$

$$v_{k,l} \leq \sum_{k' \in \mathcal{K}_c \setminus k} x_{k',l} \tag{3.18}$$

(3.17) guarantees that no platooning effect on links where the truck does not traverse, while (3.18) limits the platooning effects. K_c is the number of trucks in the fleet, so that $K_c - 1$ is sufficiently large. With the method, the fuel cost is relaxed into a linear form with auxiliary variables.

3.2.3 Overview of the Problem Relaxation

In Section 3.2, we apply several relaxation methods on the constraints of $\boldsymbol{\mathcal{P}}$ to preserve linearity. We now combine the original problem $\boldsymbol{\mathcal{P}}$ with the constraints (3.6), (3.7), (3.15), (3.17) and (3.18) to arrive at a relaxed problem formulation, as given in $\boldsymbol{\mathcal{R}}$

$$\begin{aligned} \min_{\mathbf{a}_c} \quad & \sum_{k \in \mathcal{K}_c} \left[\sum_{l \in \mathcal{E}(\mathcal{G}_k)} u_{k,l} + g_k(\mathbf{a}_c) + p_k \right] \\ \text{subject to} \quad & (3.6), (3.7), (3.15), (3.17), (3.18) \end{aligned} \tag{$\boldsymbol{\mathcal{R}}$}$$

The feasible solutions of $\boldsymbol{\mathcal{R}}$ are at integer points, making the piece wise fitting equal to the original objective values at all feasible points. Relaxation methods introduce three types of auxiliary variables, which only serving as place holders without changing feasible objective values. $\boldsymbol{\mathcal{R}}$ is guaranteed to have the same optimum objective value as $\boldsymbol{\mathcal{P}}$. It also matches the form of a standard MILP problem. The non-linearity in the cost function $f_{k,l}(\cdot)$ and $h_k(\cdot)$ are replaced by relaxations. We have taken advantage of the binary feature of the decision variables to linearize the quadratic and bilinear terms. On the other hand, the integer value constraints make the problem generally hard to solve. A few examples in the following section show the increasing trend in time complexity for solving the problem formulation $\boldsymbol{\mathcal{R}}$.

3.3 Numerical Examples

In this simulation, we focus on the time complexity of the formulation $\boldsymbol{\mathcal{R}}$ with a centralized solver. The formulation in $\boldsymbol{\mathcal{R}}$ does not change the objective value in feasible regions, thus the result is optimal, and therefore the expected result. However, a MILP problem is generally NP-hard to solve, which may result in an infeasible running time of the algorithm. The simulation focus on the scalability of the algorithm in different scenarios.

3.3.1 Simulation Set-up

There are two main goals in this simulation, including testing the time complexity increment when first, scaling up the graph size and then, enlarging the number of trucks in the fleet. 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 as in Figure 3.1 to represent a connected transportation network. The graph is also known as the Hanan grid, which is adapted for similar simulations in previous studies [17, 22]. The number of nodes used in the simulations are 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

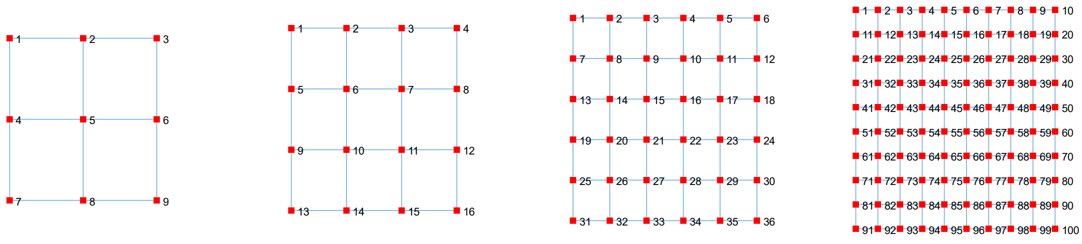

Figure 3.1: Test Hanan grids used in the example, 9-node, 16-node,36-node and 100-node

are randomly chosen from the graph. The latest arrival time is chosen in a way that feasible paths exist. The preferred arrival time is also randomly selected during the time window in a way that it 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. All links are assumed to be the same, and all weights are the same for each truck. In the setup, we consider the solo operation of a fleet, which implies that all $N_{-c,l} = 0$. Other constant settings are shown in the Table 3.1. Numerical examples computations are performed using CVX solver [28, 29]. Simulation on each fleet size setting has been repeated for 50 times.

Truck parameters			
Fleet size	K_c	Range [0, 50]	Step size 5
Time cost weight	$w_{k,t}$	5	
Preference penalty weight - late	$w_{k,l}$	5	
Preference penalty weight - early	$w_{k,e}$	5	
Piece-wise 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

Table 3.1: Parameters setting for the simulation

3.3.2 Result and Discussion

The results about the time complexity in the small- and medium-size graph are shown in Figure 3.3, where the plots indicate time vs fleet size. In the subgraph Figure 3.3 (a) and (b) for the 9 and 16 nodes, although the fleet sizes increases, the elapsed time is relatively short. It is noted that in a medium-size problem with 36 nodes as shown in Figure 3.3 (c), the time consuming for solving the problem is generally acceptable. In Figure 3.3 (d), we enlarge the regions of 0-800s to show that the time complexity 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 infeasible computation time. In a large scale problem with 100 nodes in the graph, the extreme

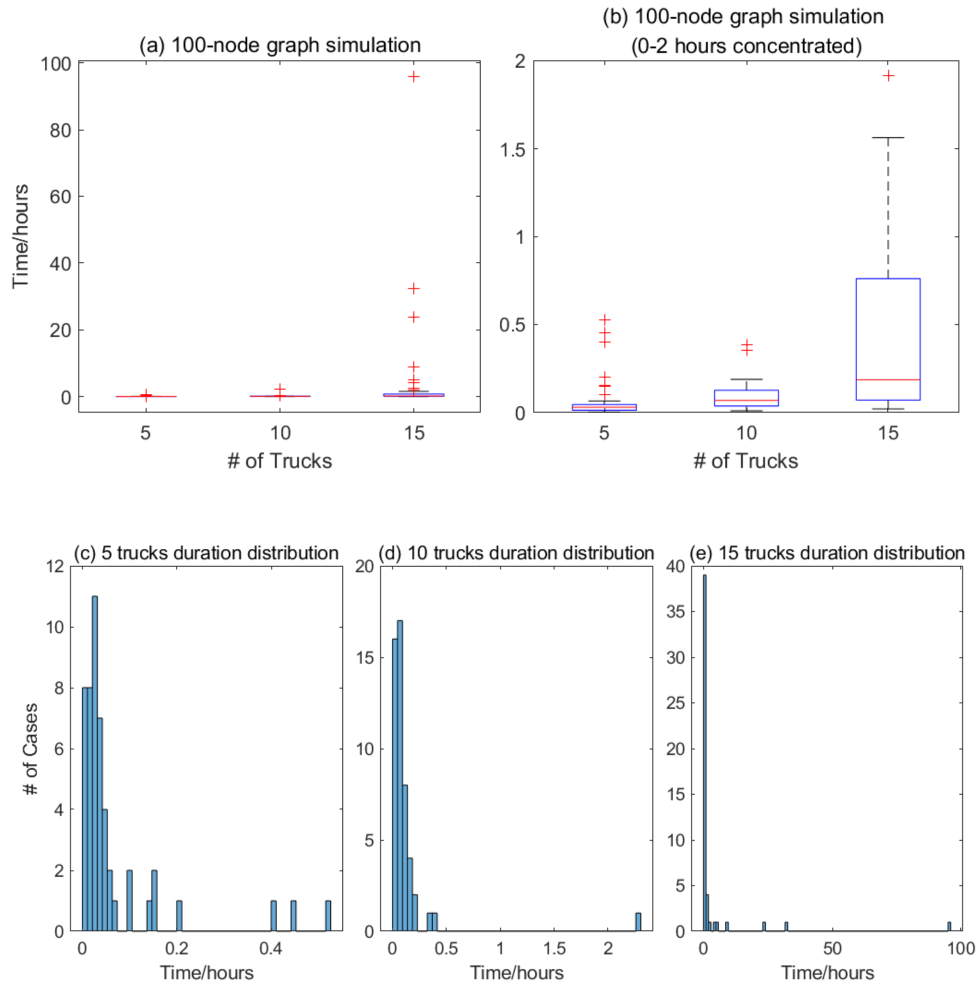

Figure 3.2: 100-node graph time complexity simulation result with centralized solver for varying number of trucks

cases take unacceptable amount of time to compute, thus the simulation terminates before raising the fleet size to 20. As shown in Figure 3.2 (a), there are extreme cases that cost infeasible amount of time. The subplot is then enlarged as shown in Figure 3.2 (b), suggesting that the time complexity may vary significantly with the fleet size increasing. Three histograms of fleet size 5 to 15 are given in Figure 3.2 (c) - (e), in which the y axis represent the amount of cases that falls 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% cases that are infeasible. The result also suggests that the time complexity 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.

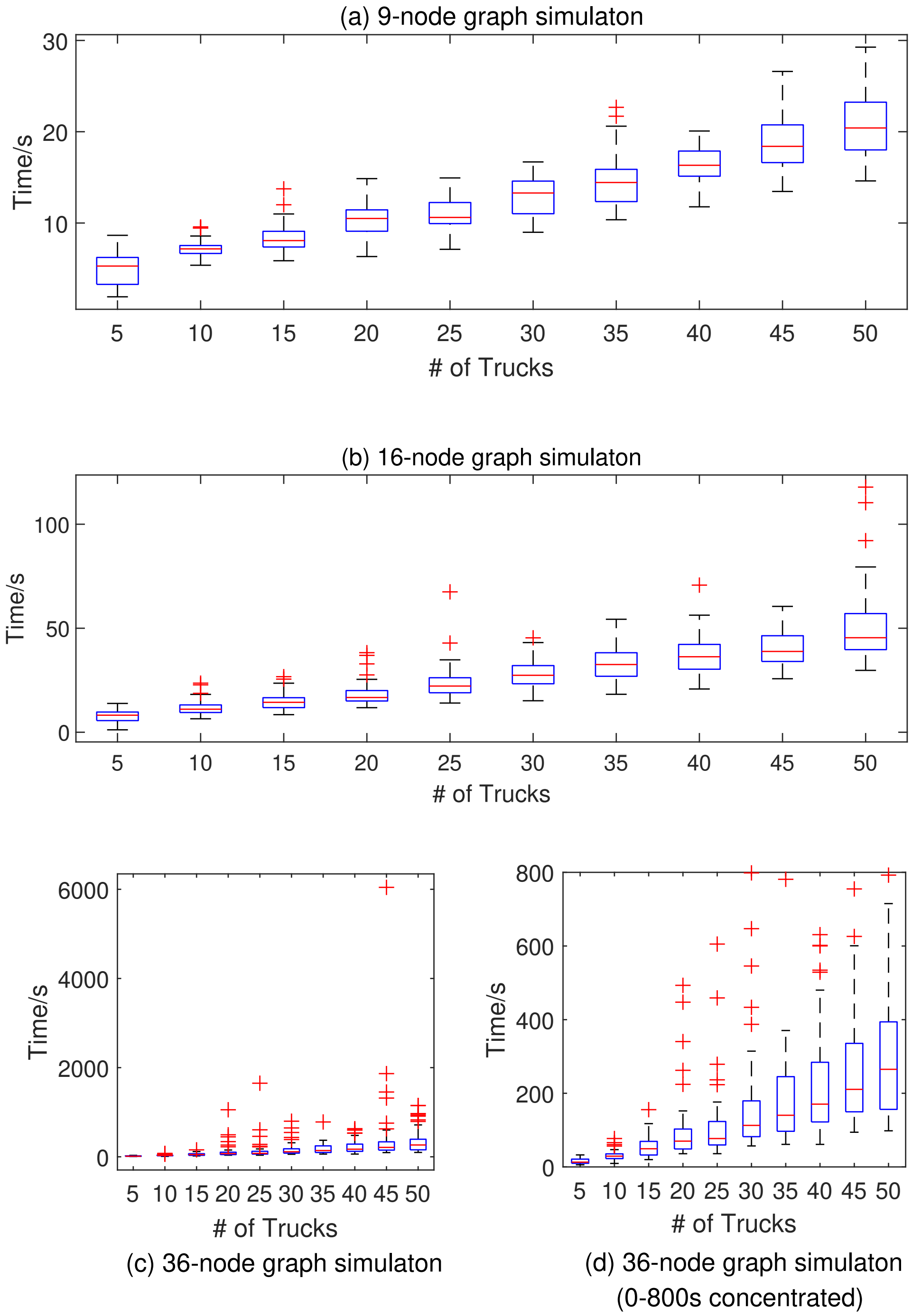

Figure 3.3: Time complexity simulation result with centralized solver for varying number of trucks

A representative demand set may reveal a more practical time complexity challenge in truck platooning coordination. The demand issues are addressed in the simulation in Chapter 5.

To deal with the limitation in centralized exact solution methods, the decentralized approach is introduced in the truck platooning problem. The decentralized approach has advantages in reducing communication requirements, sharing computation burden and flexibility, and scalability [30]. With the sustained development of autonomous vehicle technologies, vehicles can perform local computation and communicate with other entities. The vehicles are interconnected, and they may cooperate to achieve a desired global objective. In the next chapter, we will propose a method to decompose the problem to each truck instead of approximating the result in a centralized node. The large scale problem may be decomposed into a series of subproblem which are associated to each truck which may achieve a feasible time complexity.

Decentralized Method for Best-response Subproblem

4

With the observations from the numerical examples in Chapter 3, we introduce a decentralized method for solving the problem, targeting a more feasible time complexity. For easier denotations and showing the mathematical structure of the optimization problem, we reformulate the $\boldsymbol{\mathcal{R}}$. The problem is then clearly a separable problem with a series of coupled linear inequality constraints. The details and insights of such an algorithm are discussed. Last but not least, we perform numerical tests to show the time complexity, suboptimal level, and savings for the algorithm.

4.1 Decentralized Form Reformulation

For an easier denotation, we convert $\boldsymbol{\mathcal{R}}$ into a matrix form with its linear nature, which is given by

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

$$\text{subject to} \quad \mathbf{A}_k\mathbf{S}_k\mathbf{a}_c = \mathbf{b}_k \quad \forall k \in \mathcal{K}_c \tag{4.1}$$

$$\mathbf{a}_c \in [0,1]^{L_c} \quad \forall k \in \mathcal{K}_c \tag{4.2}$$

$$w_{k,l}(\mathbf{t}_k^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}) \le p_k \quad \forall k \in \mathcal{K}_c \tag{4.3}$$

$$-w_{k,e}(\mathbf{t}_k^T\mathbf{S}_k\mathbf{a}_c + t_k^{ED} - t_k^{PA}) \le p_k \quad \forall k \in \mathcal{K}_c \tag{4.4}$$

$$(\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c,l} + \beta_{k,l,j})x_{k,l} + \alpha_{k,l,j}v_{k,l} \le u_{k,l} \quad \forall j, \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \tag{4.5}$$

$$v_{k,l} \le (K_c - 1)x_{k,l} \quad \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \tag{4.6}$$

$$v_{k,l} \le \sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \quad \forall k \in \mathcal{K}_c, l \in \mathcal{E}(\mathcal{G}_k) \tag{4.7}$$

where constraints (4.1) to (4.7) are based on (3.6), (3.7), (3.15), (3.17) and (3.18). By reformulating the subproblem into a matrix form, the formulation is compact. 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 part of this section. Let $\mathbf{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{(2L_k+1)}$ be a vector for each truck,

$$\mathbf{z}_k = [\mathbf{x}_k^T, p_k, \mathbf{u}_k^T, \mathbf{v}_k^T]^T \tag{4.8}$$

where

$$\mathbf{u}_k = [u_{k,1}, \cdots, u_{k,l}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}_k) \tag{4.9}$$

$$\mathbf{v}_k = [v_{k,1}, \cdots, v_{k,l}]^T \quad \forall l \in \mathcal{E}(\mathcal{G}_k) \tag{4.10}$$

here $\mathbf{u}_k$ is the vector for all fuel cost of truck k. $\mathbf{v}_k$ represents the product of the decision variable $x_{k,l}$ and the number of platoon partners. It indicates the level of platooning benefits for truck k and thus equals to 0 because $x_{k,l} = 0$ when the truck k is not present on the link l. $\mathbf{z}_k$ 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 $\boldsymbol{\mathcal{R}}$,

$$\mathbf{c}_k^T\mathbf{z}_k = g_k(\mathbf{a}_c) + p_k + \sum_{l\in\mathcal{E}(\mathcal{G}_k)} u_{k,l} = w_{t,k}\mathbf{t}_k^T\mathbf{x}_k + p_k + \mathbf{1}_{L_k}^T\mathbf{u}_k \tag{4.11}$$

where $\mathbf{c}_k$,

$$\mathbf{c}_k = [w_{t,k}\mathbf{t}_k^T, 1, \mathbf{1}_{L_k}^T, \mathbf{0}_{L_k}^T]^T \tag{4.12}$$

Likewise, we may reformulate the equality constraint from (4.1),

$$\bar{\mathbf{A}}_k\mathbf{z}_k = \bar{\mathbf{b}}_k \tag{4.13}$$

where

$$\bar{\mathbf{A}}_k = [\mathbf{A}_k, \mathbf{0}_{L_k\times(2L_k+1)}] \tag{4.14}$$

$$\bar{\mathbf{b}}_k = [\mathbf{b}_k^T, \mathbf{0}_{(2L_k+1)}^T]^T \tag{4.15}$$

which indicates the flow conservation constraints, ensuring the $\mathbf{x}_k$ representing a feasible path. For the inequality constraints (3.6), (3.7), (3.15) and (3.17), they may be rewritten as

$$\underbrace{\begin{bmatrix} w_{k,l}\mathbf{t}_k^T & -1 & \mathbf{0}_{L_k}^T & \mathbf{0}_{L_k}^T \\ -w_{k,e}\mathbf{t}_k^T & -1 & \mathbf{0}_{L_k}^T & \mathbf{0}_{L_k}^T \\ \mathbf{C}_{k,x} & \mathbf{0}_{JL_k} & \mathbf{C}_{k,u} & \mathbf{C}_{k,v} \\ (K_c-1)\mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & -\mathbf{I}_{L_k\times L_k} \\ -\mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} \\ \mathbf{I}_{L_k\times L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} & \mathbf{0}_{L_k} \end{bmatrix}}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ \mathbf{p}_k \\ \mathbf{u}_k \\ \mathbf{v}_k \end{bmatrix}}_{\mathbf{z}_k} \le \underbrace{\begin{bmatrix} -w_{k,l}(t_k^{ED} - t_k^{PA}) \\ w_{k,e}(t_k^{ED} - t_k^{PA}) \\ \mathbf{0}_{JL_k} \\ \mathbf{0}_{L_k} \\ \mathbf{0}_{L_k} \\ \mathbf{1}_{L_k} \end{bmatrix}}_{\mathbf{d}_k} \tag{4.16}$$

where $[\mathbf{C}]_i$ denote the ith row of martix $\mathbf{C}$,

$$[\mathbf{C}_{k,x}]_{J(l-1)+l}\mathbf{x}_k = (\alpha_{k,l,j} + \alpha_{k,l,j}N_{-c,l} + \beta_{k,l,j})x_{k,l} \tag{4.17}$$

$$[\mathbf{C}_{k,v}]_{J(l-1)+l}\mathbf{v}_k = \alpha_{k,l,j}v_{k,l} \tag{4.18}$$

$$[\mathbf{C}_{k,u}]_{J(l-1)+l}\mathbf{u}_k = -u_{k,l} \tag{4.19}$$

Meanwhile, the last two rows in the matrix on the left side in (4.16) constraint the binary variable $x_{k,l}$ into the range of $[0, 1]$. The complex formulation in (4.16) may be simplified as

$$\mathbf{H}_k\mathbf{z}_k \le \mathbf{d}_k \tag{4.20}$$

The matrix $\mathbf{H}_k$ and $\mathbf{d}_k$ represents the feasible half space defined by inequality constraints The remaining constraints (3.18) may be written as

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k\mathbf{z}_k \le \mathbf{0}_{L_c} \tag{4.21}$$

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

$$\sum_{k\in\mathcal{K}_c} [\mathbf{D}_k]_l \mathbf{z}_k = v_{k,l} - \sum_{k'\in\mathcal{K}_c\setminus k} x_{k',l} \tag{4.22}$$

Notably, (4.20) and (4.13) are local for a truck, let $\mathcal{Z}_k$ be a set

$$\mathcal{Z}_k = \{\mathbf{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{2L_k+1} | \mathbf{H}_k\mathbf{z}_k \leq \mathbf{d}_k, \bar{\mathbf{A}}_k\mathbf{z}_k = \bar{\mathbf{b}}_k\} \tag{4.23}$$

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

$$\begin{aligned} \min_{\mathbf{z}_1,\cdots,\mathbf{z}_{K_c}} \quad & \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k && (\boldsymbol{\mathcal{R}}_d) \\ \text{subject to} \quad & \sum_{k\in\mathcal{K}_c} \mathbf{D}_k\mathbf{z}_k \leq \mathbf{0}_{L_c} && (\boldsymbol{\mathcal{R}}_d.1) \\ & \mathbf{z}_k \in \mathcal{Z}_k \quad \forall k \in \mathcal{K}_c && (\boldsymbol{\mathcal{R}}_d.2) \end{aligned}$$

$\boldsymbol{\mathcal{R}}_d$ is a MILP problem with coupled inequality constraints among trucks. It worth mentioning that constraint ($\boldsymbol{\mathcal{R}}_d$.1) is connected to every truck. But the rest of constraints are local. To decompose the problem, we adopt the decentralized method with minor modification in the notations, which is proposed in [31].

4.2 Decentralized Dual Subgradient Method

In this section, we introduce the algorithm, which is a variant dual subgradient method based on [31]. To provide more insights in this method and analysing the applicability in the problem of this thesis, we will analysis some key parts of the algorithm.

4.2.1 Dual Decomposition

It is common to look at a nondifferentiable optimization problem in its dual domain to obtain a favorable properties including a differentiable dual function or simpler nondifferentiable terms. The dual problem of $\boldsymbol{\mathcal{R}}_d$ is formulated as

$$\begin{aligned} \min_{\mathbf{z}_1,\cdots,\mathbf{z}_{K_c},\boldsymbol{\lambda}} \quad & \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k + \boldsymbol{\lambda}^T (\sum_{k\in\mathcal{K}_c} \mathbf{D}_k\mathbf{z}_k) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}_{L_c} \\ & \mathbf{z}_k \in \mathcal{Z}_k \quad \forall k \in \mathcal{K}_c \end{aligned} \tag{$\boldsymbol{\mathcal{D}}$}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{L_c}$ is the Lagrange multiplier for the coupled inequality constraints. Notably, the formulation in ($\boldsymbol{\mathcal{D}}$) is not a typical dual formulation as we keep the local constant sets $\mathcal{Z}_k$ within the constraints. The reason is straight forward that if we let

$$L(\mathbf{z}_1, \cdots, \mathbf{z}_k, \boldsymbol{\lambda}) = \sum_{k\in\mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k + \boldsymbol{\lambda}^T (\sum_{k\in\mathcal{K}_c} \mathbf{D}_k\mathbf{z}_k) \tag{4.24}$$

The Lagrangian L is decomposable as given in

$$L(\mathbf{z}_1, \cdots, \mathbf{z}_k, \boldsymbol{\lambda}) = \sum_{k \in \mathcal{K}_c} L_k(\mathbf{z}_k, \boldsymbol{\lambda}) = \sum_{k \in \mathcal{K}_c} (\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k)\mathbf{z}_k \tag{4.25}$$

That is for a given $\boldsymbol{\lambda}$, solving for $\mathbf{z}_k$ is a smaller subproblem as in

$$\mathbf{z}_k(\boldsymbol{\lambda}) \in \arg \min_{\mathbf{z}_k \in \mathcal{Z}_k} \quad (\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k)\mathbf{z}_k \tag{4.26}$$

$\boldsymbol{\lambda}$ is then the global variable that should be exchanged in the communication network. The method, also know as projected dual subgradient method, is to literately solving the following sub-problems,

$$\mathbf{z}_{k,i+1} = \arg \min_{\mathbf{z}_k \in \mathcal{Z}_k} \quad (\mathbf{c}_k^T + (\boldsymbol{\lambda}_i^T \mathbf{D}_k)\mathbf{z}_k \tag{4.27}$$

$$\boldsymbol{\lambda}_{i+1} = \left[\boldsymbol{\lambda}_i + \alpha(i)(\sum_{k \in \mathcal{K}_c} \mathbf{D}_k \mathbf{z}_{k,i}) \right]_+ \tag{4.28}$$

with an abuse of notation, we denote $\mathbf{z}_{k,i}, \boldsymbol{\lambda}_i$ as the value of corresponding $\mathbf{z}_k, \boldsymbol{\lambda}$ at the ith iteration. $[\cdot]_+$ is the projection to non-negative orthant. $\alpha(i)$ is the step size at iteration i. With a centralized fusion center for $\boldsymbol{\lambda}$ 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 [31]. As a matter of fact, the method provide an optimal solution of a relaxed problem

$$\begin{aligned} \min_{\mathbf{z}_1, \cdots, \mathbf{z}_{K_c}} \quad & \sum_{k \in \mathcal{K}_c} \mathbf{c}_k^T \mathbf{z}_k \\ \text{subject to} \quad & \sum_{k \in \mathcal{K}_c} \mathbf{D}_k \mathbf{z}_k \leq \mathbf{0}_{L_c} \\ & \mathbf{z}_k \in \text{conv}(\mathcal{Z}_k) \quad \forall k \in \mathcal{K}_c \end{aligned} \tag{$\boldsymbol{\mathcal{R}}_{LP}$}$$

where $\text{conv}(\mathcal{Z}_k)$ is the convex hull of the local set $\mathcal{Z}_k$. To offer a more detailed insights on the duality gap between $\boldsymbol{\mathcal{R}}_d$ and $\boldsymbol{\mathcal{D}}$, a theorem from [32] is proposed as

Theorem 4.2.1. *(Bound on Duality Gap, **Theorem 2.3 in** [32]).*There exists an $\bar{\mathbf{z}}_k \in \mathcal{Z}_k$ such that $\mathbf{D}_k \bar{\mathbf{z}}_k \leq \mathbf{D}_k \mathbf{z}_k, \forall \mathbf{z}_k \in \text{conv}(\mathcal{Z}_k)$, then

$$J^*_{\boldsymbol{\mathcal{R}}_d} - J^*_{\boldsymbol{\mathcal{D}}} \leq K_c \max_{k \in \mathcal{K}_c} \gamma_k, \quad \gamma_k = \max_{\mathbf{z}_k \in \mathcal{Z}_k} \mathbf{c}_k^T \mathbf{z}_k - \min_{\mathbf{z}_k \in \mathcal{Z}_k} \mathbf{c}_k^T \mathbf{z}_k \tag{4.29}$$

The Theorem 4.2.1 suggests the existence of a feasible solution within the bound given in (4.29). 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 [32]. We present the related parts in the next section.

4.2.2 Primal Problem Modification

The method proposed is based on tightening the feasible region limited by (4.21) with an appropriate contraction. The contraction is to tighten the inequality constraints to guarantee the feasibility, though may increase suboptimal level. The tighten version of the relaxed problem $\boldsymbol{\mathcal{R}}_{LP}$ is given by

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

in which the non-negative $\boldsymbol{\rho} \in \mathbb{R}^{L_c}$ is the tightening vector. Consequently, the dual problem of $\boldsymbol{\mathcal{R}}_{LP,\boldsymbol{\rho}}$ is given by

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

Notably, the contraction problem does not affect the subproblem as in the (4.29). Let $\tilde{\boldsymbol{\rho}}$ be a proper selection of $\boldsymbol{\rho}$

$$[\tilde{\boldsymbol{\rho}}]_j = K_c \left\{ \max_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k - \min_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k \right\} \tag{4.30}$$

where $[\cdot]_j$ is the jth row of the corresponding matrix. Let $\tilde{\boldsymbol{\mathcal{R}}}_{LP,\rho}$ and $\tilde{\boldsymbol{\mathcal{D}}}_\rho$ be the primal-dual pairs where $\boldsymbol{\rho} = \tilde{\boldsymbol{\rho}}$.

The contraction method with $\tilde{\boldsymbol{\rho}}$ is different from the original formulation in [32], in which they assume that the length of $\boldsymbol{\rho}$ is considerably smaller than the amount of agents. In general, this assumption does not hold in truck platooning problem as in formulation $\boldsymbol{\mathcal{R}}_d$, as $L_c \geq K_c$. We may assume that for a optimal solution set $\tilde{\mathbf{z}}_{LP}^* = [\tilde{\mathbf{z}}_{1,LP}^*; \cdots ; \tilde{\mathbf{z}}_{K_c,LP}^*]$ of $\boldsymbol{\mathcal{R}}_{LP,\boldsymbol{\rho}}$, a subset $\mathcal{K}_{c,1} \subseteq \mathcal{K}_c$ exists such that

$$\tilde{\mathbf{z}}_{k,LP}^* = \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) \quad \forall k \in \mathcal{K}_{c,1} \tag{4.31}$$

where $\tilde{\boldsymbol{\lambda}}^*$ is an optimal solution of $\boldsymbol{\mathcal{D}}_\rho$ and $\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*)$ is the corresponding solutions given by (4.26). The subset $\mathcal{K}_{c,1}$ contains the trucks which have the same optimal plan with or without the contraction method implemented. By dividing the trucks into two sets, we have the following derivations,

$$\begin{aligned} \sum_{k\in\mathcal{K}_c} \mathbf{D}_k\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) &= \sum_{k\in\mathcal{K}_{c,1}} \mathbf{D}_k\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) + \sum_{k\in\mathcal{K}_c\setminus\mathcal{K}_{c,1}} \mathbf{D}_k\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) \\ &= \sum_{k\in\mathcal{K}_{c,1}} \mathbf{D}_k\tilde{\mathbf{z}}_{k,LP}^* + \sum_{k\in\mathcal{K}_c\setminus\mathcal{K}_{c,1}} \mathbf{D}_k\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) \\ &= \sum_{k\in\mathcal{K}_c} \mathbf{D}_k\tilde{\mathbf{z}}_{k,LP}^* + \sum_{k\in\mathcal{K}_c\setminus\mathcal{K}_{c,1}} \Big(\mathbf{D}_k\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) - \mathbf{D}_k\tilde{\mathbf{z}}_{k,LP}^*\Big) \end{aligned} \tag{4.32}$$

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

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k \tilde{\mathbf{z}}_{k,LP}^* \leq -\tilde{\boldsymbol{\rho}} \tag{4.33}$$

Otherwise, if $\boldsymbol{\mathcal{R}}_{LP,\boldsymbol{\rho}}$ has multiple optima, when solving the problem in a decentralized set-up, there is no guarantees that all trucks get the same optima solution as they work independently at this step, which possibly failing the inequality in (4.33). On the other hand, based on (4.26), the $\mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*)$ is determined by $\tilde{\boldsymbol{\lambda}}^*$. If $\tilde{\boldsymbol{\lambda}}^*$ is not unique, this may results in a uncertainty in selection of $\tilde{\boldsymbol{\lambda}}^*$ of each truck, resulting a failure in (**??**). To sum up, the following assumption is proposed

Assumption 4.2.1. *(Uniqueness, **Assumption 2.4** in [32]).* $\tilde{\boldsymbol{\mathcal{R}}}_{LP,\rho}$ and $\tilde{\boldsymbol{\mathcal{D}}}_\rho$ have unique solutions $\tilde{\mathbf{z}}_{LP}^*$ and $\tilde{\boldsymbol{\lambda}}^*$.

With Assumption 4.2.1, we have (4.33) held and

$$\sum_{k\in\mathcal{K}_{c,2}} \left([\mathbf{D}_k]_j \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) - [\mathbf{D}_k]_j \tilde{\mathbf{z}}_{k,LP}^*\right) \leq K_c \max_{k\in K_c} \left\{\max_{\mathbf{z}_k\in\mathcal{Z}_k} [\mathbf{D}_k]_j \mathbf{z}_k - \min_{\mathbf{z}_k\in\mathrm{conv}(\mathcal{Z}_k)} [\mathbf{D}_k]_j \mathbf{z}_k\right\} \tag{4.34}$$

This is straight forward to see that there are at most K_c trucks in $\mathcal{K}_c \backslash \mathcal{K}_{c,1}$. It is clear that based on (4.22), $[\mathbf{D}_k]_j \mathbf{z}_k$ is nothing but 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. This supports that (4.30) is a reasonable choice. Following (4.32),

$$\sum_{k\in\mathcal{K}_c} \mathbf{D}_k \tilde{\mathbf{z}}_{k,LP}^* + \sum_{k\in\mathcal{K}_c\backslash\mathcal{K}_{c,1}} \left(\mathbf{D}_k \mathbf{z}_k(\tilde{\boldsymbol{\lambda}}^*) - \mathbf{D}_k \tilde{\mathbf{z}}_{k,LP}^*\right) \leq -\tilde{\boldsymbol{\rho}} + \tilde{\boldsymbol{\rho}} = \mathbf{0}_{L_c} \tag{4.35}$$

To summarize, the following theorem is given.

Theorem 4.2.2. *(Feasibility guarantees, **Theorem 3.1** in [32])* If Assumption 4.2.1 is satisfied, $\mathbf{z}^*(\tilde{\boldsymbol{\lambda}}^*) = [\mathbf{z}_1^*(\tilde{\boldsymbol{\lambda}}^*); \cdots; \mathbf{z}_{K_c}^*(\tilde{\boldsymbol{\lambda}}^*)]$ is feasible for $\boldsymbol{\mathcal{R}}_d$.

However, it worth mentioning that in truck platooning problem, the chosen of $\tilde{\boldsymbol{\rho}}$ is unnecessarily large, compromising on the suboptimal level. We may build a tighter $\boldsymbol{\rho}_o$ based on (4.22),

$$\boldsymbol{\rho}_o = 2(K_c - 1) \cdot \mathbf{1}_{L_c} \tag{4.36}$$

Even with $\boldsymbol{\rho}_o$, the tightening is strict and assuming the worst case. 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 [31]. We modified the algorithm to meet the properties of our problem.

4.2.3 Decentralized Truck Platooning Coordination Algorithm

The proposed decentralized truck platooning algorithm is given as Algorithm 1. Instead of fixing the value of $\boldsymbol{\rho}$, the algorithm is adaptive in contracting the primal problem

Algorithm 1 Decentralized Truck Platooning Coordination

1: $i = 0$ {% *Round of iteration*}
2: $\boldsymbol{\lambda}(i) = \mathbf{0}$
3: $\bar{\mathbf{s}}(i) = -\infty,\ k = 1, \cdots, K_c$
4: $\underline{\mathbf{s}}(i) = +\infty,\ i = 1, \cdots, K_c$
5: **repeat**
6: **for** $k = 1$ **to** K_c **do**
7: $\mathbf{z}_k(i+1) \leftarrow \arg\min_{\mathbf{z}_k \in \hat{\mathcal{Z}}_k} \{(\mathbf{c}_k^T + \boldsymbol{\lambda}^T \mathbf{D}_k)\mathbf{z}_k\}$
8: **end for**
9: $\bar{\mathbf{s}}_k(i+1) \leftarrow \max\{\bar{\mathbf{s}}_k(i), \mathbf{D}_k \mathbf{z}_k(i+1)\}, \quad k = 1, \cdots, K_c$
10: $\underline{\mathbf{s}}_k(i+1) \leftarrow \min\{\underline{\mathbf{s}}_k(i), \mathbf{D}_k \mathbf{z}_k(i+1)\}, \quad k = 1, \cdots, K_c$
11: $\boldsymbol{\rho}_k(i+1) = \bar{\mathbf{s}}_k(i+1) - \underline{\mathbf{s}}_k(i+1), \quad k = 1, \cdots, K_c$
12: $\boldsymbol{\rho}(i+1) = K_c \max\{\boldsymbol{\rho}_1(i+1), \cdots, \boldsymbol{\rho}_{K_c}(i+1)\}$
13: $\alpha(i) = a/\sqrt{(i+1)}, \quad a > 0$
14: $\boldsymbol{\lambda}(i+1) = [\boldsymbol{\lambda}(i) + \alpha(i)(\sum_{i=1}^{m} \mathbf{D}_k \mathbf{z}_k + \boldsymbol{\rho}(i+1))]_+$
15: $i \leftarrow i + 1$
16: **until** some stopping criterion is met.

to achieve a better suboptimal level. There are two main parts of preparing procedure before the implementation of the algorithm. To satisfy the Assumption 4.2.1, a perturbation in the cost vector $\mathbf{c}_k$ may be required. 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 assumption in previous works is that in step 7, the feasible set must be bounded [31]. 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

$$v_{k,l} \geq 0, \quad u_{k,l} \leq \alpha_{k,l,1} + \beta_{k,l,1} \tag{4.37}$$

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 place holder 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 the (4.37) as

$$\underbrace{[\mathbf{0}_{L_k}^T, 0, \mathbf{1}_{L_k}^T, -\mathbf{1}_{L_k}^T]}_{\mathbf{M}_k} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ p_k \\ \mathbf{u}_k \\ \mathbf{v}_k \end{bmatrix}}_{\mathbf{z}_k} \leq \underbrace{\begin{bmatrix} \mathbf{0}_{L_k}^T \\ 0 \\ \boldsymbol{\alpha}_{k,1} + \boldsymbol{\beta}_{k,1} \\ \mathbf{0}_{L_k}^T \end{bmatrix}}_{\mathbf{n}_k} \tag{4.38}$$

The extra constraints given by (4.38) is also linear and local, thus for a simple notation in the algorithm, we define

$$\hat{\mathcal{Z}}_k = \mathcal{Z}_k \cap \{\mathbf{z}_k \in \mathbb{Z}^{L_k} \times \mathbb{R}^{2L_k+1} | \mathbf{M}_k \mathbf{z}_k \leq \mathbf{n}_k\} \tag{4.39}$$

The algorithm is a variant of dual sub-gradient method for the tighten primal problem. After initialization, the algorithm locally compute a subproblem in step 7, which is a variant of (4.27). The following steps from 9 to 12 is an adaptive step for finding a minimum $\boldsymbol{\rho}$ for a better suboptimal level. The difference is quite significant as in [32] the contraction is fixed. Since taking values from finite sets, the sequence of $\boldsymbol{\rho}(i)$ converges to some $\bar{\boldsymbol{\rho}}$. In a similar manner, with $\boldsymbol{\rho} = \bar{\boldsymbol{\rho}}$, the contraction is still sufficient to ensure feasibility. We recommend the reader to refer to [31] for the proof. With the latest update on the $\boldsymbol{\rho}$, the centralized node perform an update on $\boldsymbol{\lambda}$ in step 14, which origins form (4.28). $\alpha(i)$ is a nonsummable diminishing step size with a positive scalar a [33]. Since the algorithm is based on a decentralized set-up, there may not be a clear stop criterion unless for information is sent to the centralized node. It is also practical to set upper bounds for iterations.

To summarize the flow of the Algorithm 1, at iteration i, each truck updates locally the tentative solution $\mathbf{z}_k(\boldsymbol{\lambda}(i-1))$ and sent 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 [31], though we build up the algorithm in a synchronous setup for a comparable result with centralized result.

4.3 Numerical Examples

A comparison simulation illustrates the improvement in time complexity with the decentralized method in this section. The simulation also addresses the compromise in optimality as a result of decomposing the problem. A simple decentralized method is applied as the benchmark for the Algorithm 1, which is opportunistic platooning on each truck's solo optimal plan.

The simulation is performed on the same set-up as introduced in Section 3.3.1 but only for the medium-sized 36-node Hanan graph. At each iteration of Algorithm 1, step 6-8 is 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 step 9-14. The time complexity of iteration i is the sum of the worst agent's local computation time and the centralized computation, assuming zero communication latency.

4.3.1 Results and Discussions

The result of time complexity improvement is given in Figure 4.1, showing the duration of the solving process based on the assumptions given in the previous paragraph and the centralized solver running time. The centralized solver elapsed time is significantly influenced by the demand of trucks, with relatively longer cases and possibly infeasible in a large scale problem, as shown in Figure 4.1 (a). By contrast, the decentralized algorithm shows better consistency in solving problems with different demands and sizes as in Figure 4.1 (b). The result of decentralized method is enlarged to highlight the range from 30s to 90s as in Figure 4.1 (c). Although Algorithm 1 takes more time in small problems, it shows the advantage of solving a large scale problem. Given the size of the graph, the problem is broken down into smaller problems with almost the same size problems, 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 time complexity

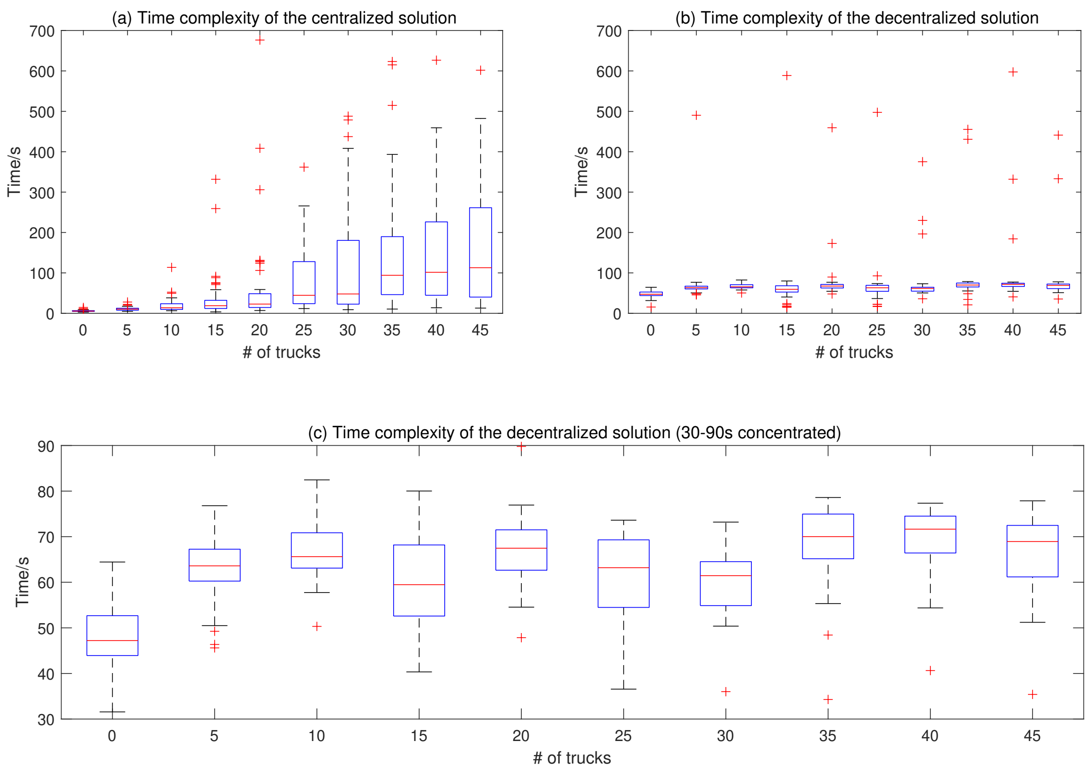

Figure 4.1: Time complexity comparison between centralized solution and decentralized solution

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

On the other hand, the improvement is the result of a compromise in optimality. The result is given in Figure 4.2, showing the performance in benefiting the stake-holders compared to the centralized solution and forming platoons without coordination beforehand. The graph in Figure 4.2 (a) is the relative gap between the optimal cost given by the centralized method and 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, which are all trucks operate without coordination beforehand, and form 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 4.2 (b). Although the reduction is insignificant, the lowest subplot indicates the algorithm

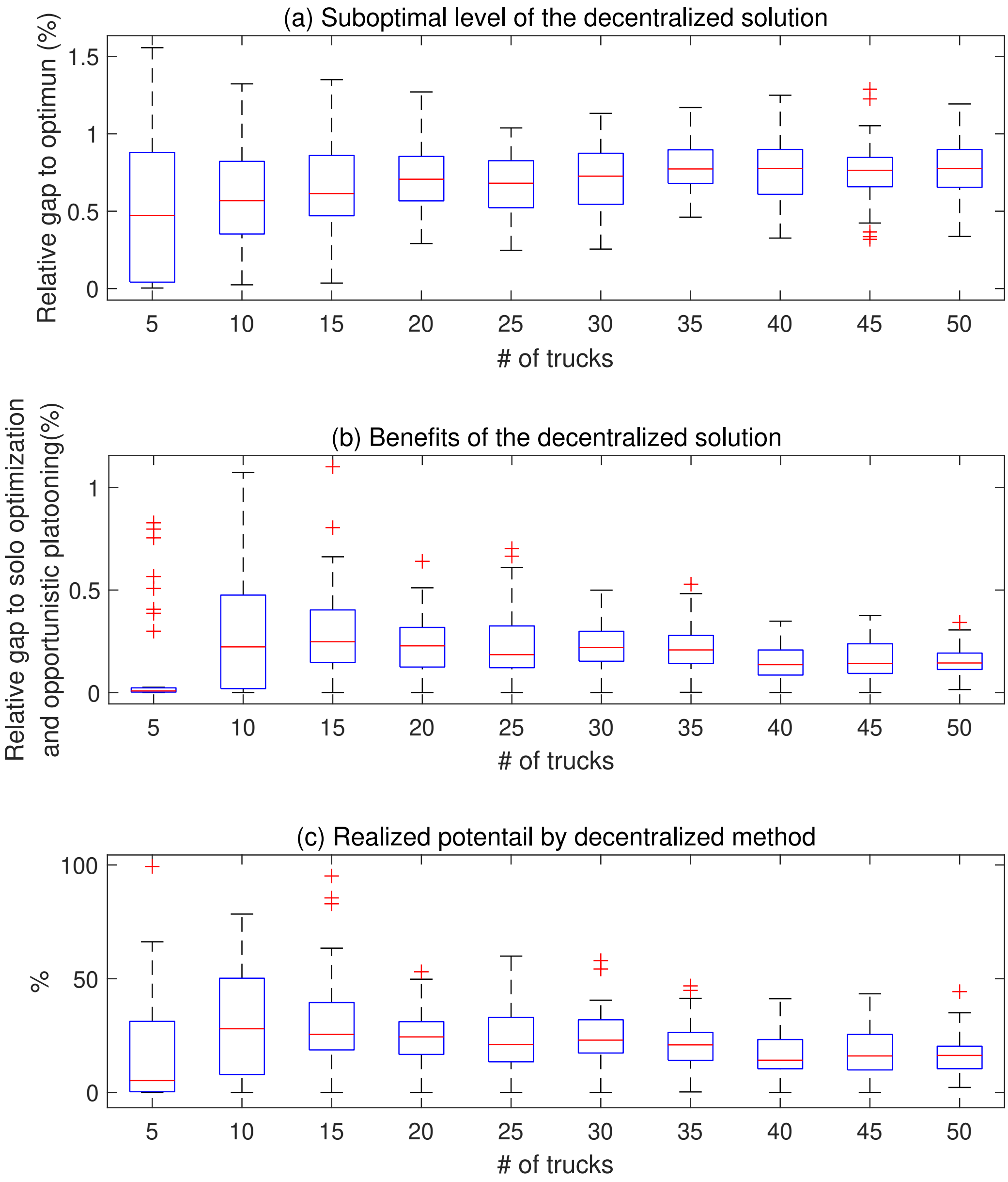

Figure 4.2: Sub-optimal level and benefits of the decentralized solution

realizes about 20% of the potential as shown in Figure 4.2 (c).

With the result so far, we learn that the implementation of Algorithm 1 is likely to benefit the shake holders of the fleet as it has significant improvement in time complexity. We are interested in the performance of the best-response search with a feasible way of solving the subproblem of best-response search. However, the absolute and relative objective value reduction depends on the demand, including the truck assignment distribution among the geographical graph and the period. In the next chapter, we introduce the generation of an analogous demand set from real-world information to indicate the real improvement in objective values and the effects of best-response searching.

5 Simulation and Discussion

The previous chapter analysis the performance of the decentralized algorithm for the best-response subproblem. In this chapter, we will build a representative test case to test the best response search method with the subproblems solved by Algorithm 1. Real shake holders are unwilling to share information of interest. Thus a privacy-preserving way of information exchange is required for our best-response search. The privacy-preserving way of information exchanged is introduced in the first section of the chapter to show more insights into the real implementation challenges. Since this thesis is one of the first work to build such a test case, we offer detailed reasoning while generating the dataset. In the last section, we present the results and offer our suggestions on the implementations and future work.

5.1 Privacy-preserving Communication Exchange

For the fleet c, the information of interests is the combination of all other fleets' plans, which is indicated by $\mathbf{a}_{-c}$. More specifically, the number of trucks N_l on all links, which is defined in (2.17) are the critical information . Then $\mathbf{a}_{-c}$ may be formulated as

$$\mathbf{a}_{-c} = \sum_{k \in \mathcal{K} \setminus \mathcal{K}_c} \bar{\mathbf{x}}_k \tag{5.1}$$

where $\bar{\mathbf{x}}_k$, defined in (2.2), is the decision vector of a truck without the refining process, which is introduced in Section 2.3. Let $\mathbf{y}_c \in \mathbb{R}^L$ be the information vector exchanged among fleets in a best response search set-up

$$\mathbf{y}_c = \sum_{k \in \mathcal{K}_c} \bar{\mathbf{x}}_k \tag{5.2}$$

Consequently,

$$\mathbf{a}_{-c} = \sum_{c' \in \mathcal{C} \setminus c} \mathbf{y}_{c'} \tag{5.3}$$

Let $\bar{\mathbf{y}}$ be the element-wise average of all $\mathbf{y}_c$, then

$$\mathbf{a}_{-c} = C\bar{\mathbf{y}} - \mathbf{y}_c \tag{5.4}$$

where $\bar{\mathbf{y}} = \frac{1}{C}\sum_{c \in \mathcal{C}} \mathbf{y}_c$. Assuming C, the number of all fleets, is known to all, the only information to exchange is $\bar{\mathbf{y}}$.

Estimating the average value among all agents without revealing the information of its own is a crucial topic in many applications. The algorithm for solving such problem is well developed in a series of studies [34, 35]. The details of such algorithm is not the focus of this study. In the previous work, the algorithm is based on a connected and possibly varying graph, thus it is possible to modify it for a decentralized set-up [34]. We address the algorithm as a function denoted by $(\bar{\mathbf{y}}) = Privacy\_Averaging(\mathbf{y}_c, \mathbf{y}_{-c})$.

By denoting the Algorithm 1 as

$$(\mathbf{z}_1, \cdots, \mathbf{z}_k) = Decentral\_TPC([\mathbf{c}_1, \cdots, \mathbf{c}_k], [\mathcal{Z}_1, \cdots, \mathcal{Z}_k], [\mathbf{D}_1, \cdots, \mathbf{D}_k])$$

we introduce the routine for the multi-fleet coordination in the Algorithm 2, where $\mathbf{c}_1, \cdots, \mathbf{c}_k$ are cost vectors given in (4.11), $\mathcal{Z}_1, \cdots, \mathcal{Z}_k$ are defined in (4.23) and $[\mathbf{D}_1, \cdots, \mathbf{D}_k]$ are given in (4.22). It is noted that the change in $\mathbf{y}_{-c}$ only affects the local set of a truck $\mathcal{Z}_k$. The process is packaged as a subfunction named $(\mathcal{Z}_k) = Build\_Zk(\mathbf{y}_{-c})$, in which we substituting the element from $\mathbf{y}_{-c}$ to corresponding $N_{-c,l}$, as defined in (3.11). At step 8, $\mathbf{R}_{c,k}$ serves as a linear transformation matrix to

Algorithm 2 Multi-fleet Best Response Search Routine

1: $i = 0$ {% *rounds of iteration*}
2: $\bar{\mathbf{y}}_i = \mathbf{0}$ {% *initialization*}
3: $\mathbf{y}_{-c,i} = \mathbf{0}, \forall c \in \mathcal{C}$
4: **repeat**
5:   **for** $c = 1$ **to** C **do**
6:     $\mathbf{y}_{-c,i} = C\bar{\mathbf{y}}_i - \mathbf{y}_{c,i}$
7:     $(\mathcal{Z}_{k,i}) \leftarrow Build\_Zk(\mathbf{y}_{-c,i}), \forall k \in \mathcal{K}_c$
8:     $(\mathbf{z}_{1,i}, \cdots, \mathbf{z}_{k,i}) = Decentral\_TPC([\mathbf{c}_1, \cdots, \mathbf{c}_k], [\mathcal{Z}_{1,i}, \cdots, \mathcal{Z}_{k,i}], [\mathbf{D}_1, \cdots, \mathbf{D}_k])$
9:     $\mathbf{y}_{c,i} = \sum_{k \in \mathcal{K}_c} \mathbf{R}_{c,k} \mathbf{z}_{k,i}$
10:     $\bar{\mathbf{y}}_i = Privacy\_Averaging(\mathbf{y}_{c,i}, \mathbf{y}_{-c,i})$
11:   **end for**
12: **until** some stopping criterion is met.

convert all $\mathbf{Z}_k$ to the $\mathbf{y}_c$. This routine operates under strict assumption on the synchronization of all fleets. The privacy-preserving feature relies on iterations to converge, indicating a possibly considerable latency. This issue is not yet addressed in this thesis but assuming the time complexity of the averaging process can be ignored for the best-response search. We recommend the reader to find more insights about the elapsed time of such algorithm in the previous work [34].

5.2 Simulation Set-up

5.2.1 Demand and Geographical Map

As shown in the simulation in Chapter 3 and Chapter 4, the time complexity on the graph varies significantly with randomly generated demand input. To reveal a more representative time complexity in real-world scenarios, we introduce our settings in this section.

The geographical map is constructed with a simplified Dutch road network, as shown in Figure 5.1. There are 67 nodes in the graph, representing main cities or intersections on the highway. The edge of the graph is assigned with a distance at 5km discrete

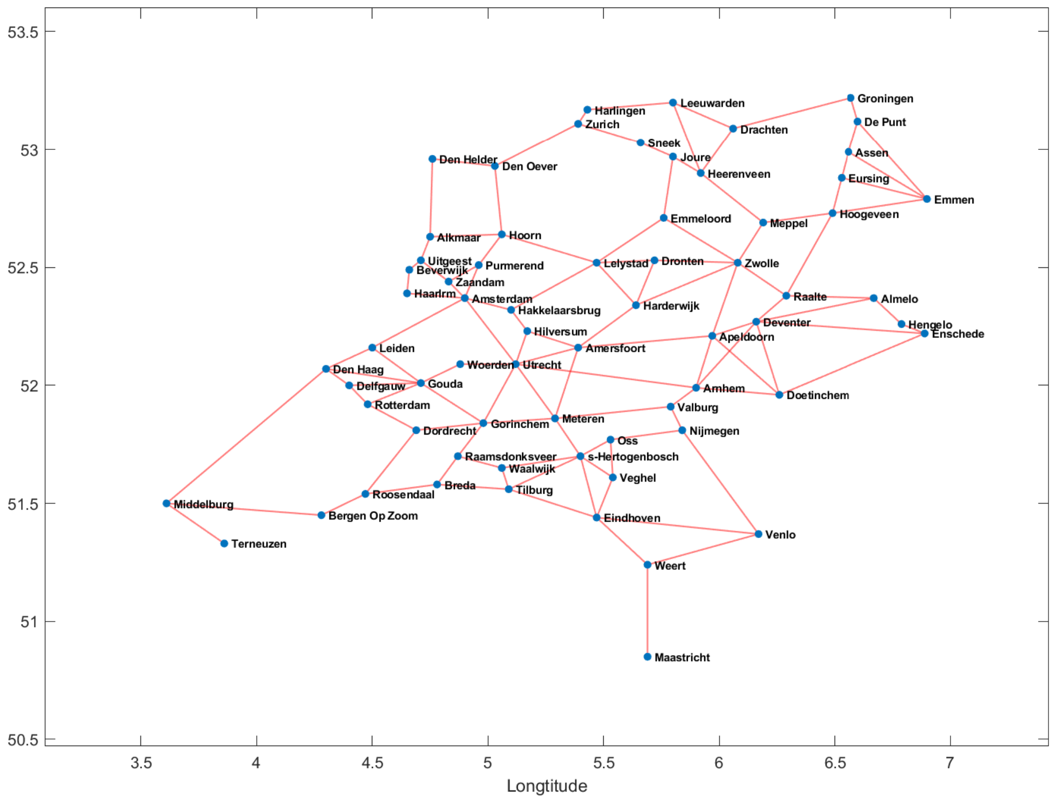

Figure 5.1: Dutch road network with node labeled with corresponding physical locations

Supermarket Chains	Distribution Centers
A	Delfgauw, Metern, Zaandam, Zwolle
B	Raalte, Eursing, Breda, Woerden, Veghel
C	Waalwijk, Heerenveen

Table 5.1: Distribution centers of three supermarket chains in the Netherlands

scales. The demand for freight transportation is considered private information. Thus there is no available public demand set. We then collect the distribution of three major supermarket chains in the Netherlands to represent the demand distribution. It is assumed that with more dense stores in certain areas, there is more demand. We then assume, on average, each store needs one truck. The stores are gathered to nearby nodes on the map for a simplified demand distribution, and the distribution is shown in Figure 5.2. The demanding stores are considered geographical destinations, while we assume the distribution centers of each supermarket chain as the geographical origins. The details of each supermarket's demands are shown in Table 5.1. We assume that destinations are served by the closest distribution centers of its own company.

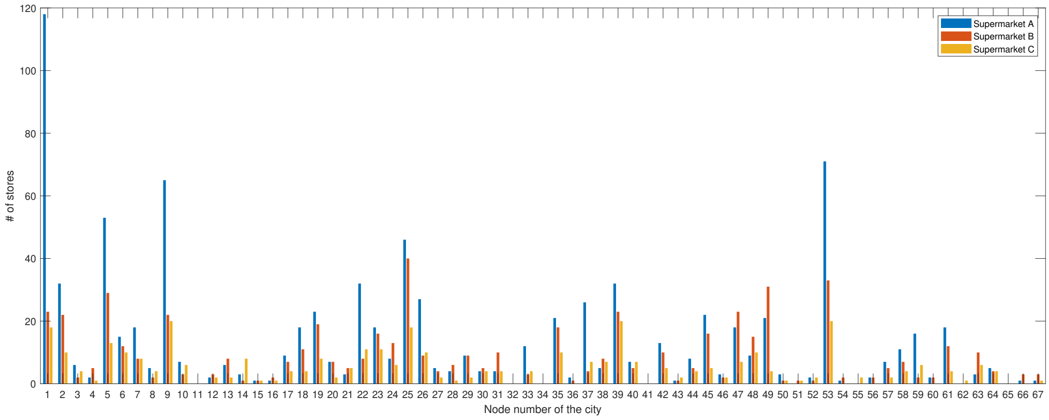

Figure 5.2: Supermarket distributions among cites in the Netherlands [2, 3, 4]

5.2.2 Parameters Assumptions and Settings

It is reasonable that supermarket chains avoid peak hours to eliminate the influence of traffic. We consider the operation hours are limited within 10:00 - 15:00 to represent the non-peak hour operation features. The bounds are assigned to be the earliest departure time and latest arrival time. However, the preference is generally challenging to assume, as it may vary based on the owners' interest. Though slightly reducing the complexity, we set the weights of the preference to zeros.

To properly convert objective values into a proper monetary cost, in this section, we offer the motivation on our selection of the cost. A more detailed table with reference source is given in Table 5.2. The longest distance from the distribution center to stores

Parameter	Description	Value	Unit	Source
w_f	Price of oil	0.002123	€/g	[36]
θ_l	Road gradient	0		Assumption in [16]
M_k	Total vehicle weight	20000	kg	[16]
ξ	Fuel-to-mass ratio	1		[37]
κ	Heating value of the fuel	44	kJ/g	[37]
ρ	Density of air	1.2041	kg/m^3	[37]
g	Gravitational constant	9.81	m/s^2	[37]
ϕ_0	Air reduction factor	0.32		[37]
E_k	Engine speed	33	rev/s	[37]
μ_k	Engine friction factor	0.2	kJ/rev/L	[37]
V_k	Engine displacement	5	L	[37]
$c_{d,k}$	Aerodynamic drag coefficient	0.7		[37]
A_k	Front area of the truck	3.912	m^2	[37]
$c_{r,k}$	Rolling resistance coefficient	0.01		[37]
ε_k	Drive train efficiency	0.4		[37]
ϖ_k	An efficiency parameter for the engine	0.9		[38]
$w_{t,k}$	Time cost for operation	1.40775	€/7.5 min	[16]
$w_{e,k}$	Weight for early arrival penalty	0	€/7.5 min	Assumption in this thesis
$w_{l,k}$	Weight for late arrival penalty	0	€/7.5 min	Assumption in this thesis

Table 5.2: Parameters settings for Dutch supermarket chains simulation

is 170km. Considering the real-world limitations and simpler graph modeling, we offer two-speed options for trucks, 40 km/h, and 80 km/h. Thus, all assignments are finished in a reasonable period. Without losing generality, we assign the fuel cost in 5km step for each truck with the same expectation value with random perturbation, guaranteeing the Assumption 4.2.1 being followed. With the selection of the speed, we divide the operation hours into 40 intervals, which identically represent 7.5 min.

Another factor about the complexity is the number of segments, J, which is introduced in (3.8) and (3.9. It is to set to 2 with the second segment set horizontal, which is realized by setting $\alpha_{k,l,2} = 0$ to prevent negative fuel cost. It is also a reasonable choice as in the real world, long platoons may jam the highway traffic. 2-vehicle platoons are feasible scenarios as the overall length of the platoon is acceptable [39]. In our model, there is no limitation for the partner amount, but by setting $J = 2$, there are no extra benefits in forming platoons with more than two vehicles. In terms of fuel savings, the simulation result has the same behavior as if the longer platoons are broken into multiple 2-vehicle platoons.

In the aspect of time cost, the cost of time is remarkably more expensive than the fuel, the trucks that are assigned with the same origin-destination pairs will be travelling together on the fastest path. Since the trucks will operate in platoons along the way, we may consider them as one virtual truck with higher unit cost in both time and fuel. The trucks with the same assignments may be considered as one single truck with every cost weights multiplied by the real amount of trucks, which remarkably scale down the size of the problem.

Given the limited number of fleets in this demand set, the upper bounds for best-response search iteration is set to 3, which does not influence the cost value as shown in the latter section.

5.3 Results and Conclusion

The main goal of this simulation is to study the performance of the proposed method in the real-world featured demand set. The time complexity of the best-response search with Algorithm 1 is compared with the independent centralized optimization of each company. The effect on the cost, especially the performance of best-response search, is also revealed in this simulation.

5.3.1 Simulation Results and Discussions

With refining and pruning methods, the size of the problem is reduced in the truck platooning application, enabling the centralized solver to efficiently process this numerical example. The elapsed time comparison between employing centralized method without best-response search and the proposed approach in this thesis is given in Figure 5.3. The results still suggests that the proposed approach is not sensitive to the size of the fleet, as all three companies spend about the same computational time despite the variation in the fleet size and demand. In cases of Company A and B, in which there are larger logistical demand, it is revealed that the best-response search is more computationally cheap in terms of time if assuming ideal communication latency and

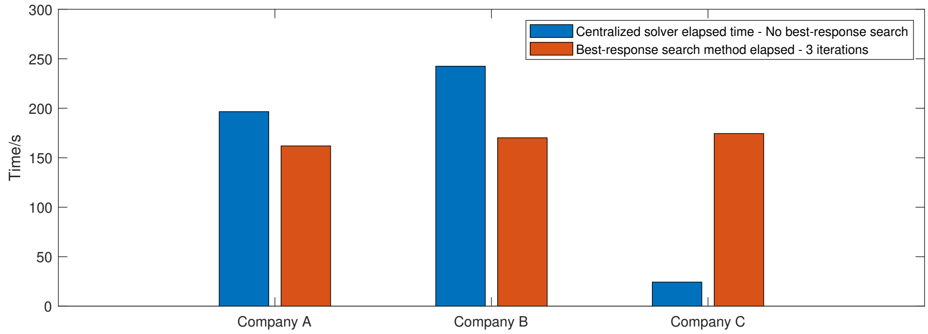

Figure 5.3: Elapsed time comparison between independent centralized optimization and the proposed decentralized best-response search

cost. On the other hand, Company C spent more time in best-response search that solving the optimization problem independently.

The other critical result is the cost reduction, which is listed in Table 5.3. The

	Company A	Company B	Company C
Centralized optimization within single fleet	11.944	10.123	8.710
Best-response search with Algorithm 1	11.944	10.123	8.710

Table 5.3: Cost comparison between two approaches (unit: 1000 Euros)

messages delivered from this result are both positive and negative in the best-response search method with Algorithm 1. First of all, the cost is the same, meaning the Algorithm 1 delivers the optimal result. For Company A and B, even with more rounds of iterations, the overall elapsed time is reduced without compromising the optimality. However, the results also reveals that in this demand set, network structure and parameter setting, the multi-fleet truck platooning is not beneficial. In the case for Company C, more computational cost are spent for no return in operation cost.

The reason of this result may be closely related to the feature of the demand set. As mentioned in the last section, the dominating time cost leads to a strict limitation on the detouring and eliminates the opportunity to reduce velocity for more platooning chances. Unlike the Hanan grid, real-world highway is less likely to find an equal-length alternative path. The very limited platooning coordination chances makes it easier for Algorithm 1 to reach optimality but eliminate the benefit potential in cooperation with other fleets.

6 Conclusions and Future Work

6.1 Conclusions and Suggestions

In this thesis, we proposed a distributed framework for multi-fleet truck platooning coordination. The interactions among non-cooperative agents are modeled as a finite game that employs a collaborative best-response search for equilibrium. The search routines are broken down into an optimization problem, which we refer to as the best-response subproblem. The best-response subproblem is converted into a MILP and we propose a decentralized algorithm to achieve a solution with feasible time complexity. A more detailed review on goals of the thesis and the reflection on corresponding chapters is given in Table 6.1.

Now we give a brief summary about the contribution of this thesis. We also introduce a privacy-preserving average consensus method for information exchange among fleets, removing one of the practical barriers for the stake holders. In the aspect of cooperative truck platooning, we involve different aspects of cost in the objective function while introducing decision parameters including routines, schedule and speed. The assumption on the trucks' type is removed allowing users to configure the parameters of each truck. On the other hand, although we preserve the standard MILP form of the problem with relaxation skills, the off-the-shelf solver meets infeasible cases in terms of solving time. The limitation in computation of the exact solution method is addressed in this thesis and improved by decentralized optimization methods.In addition, we employ a real-world featured demand set for the simulation, in contrast to a randomly generated demand in the previous works.

Based on the result, for cooperative truck platooning coordination, we suggest that the decentralized algorithm for solving the MILP problem to be further studied in a real-time scenario. Since Algorithm 1 can be readily modified as an asynchronous distributed algorithm, it is likely that the algorithm may be repeated in a more frequent manner without causing a significant load in the inter-vehicle communication network as it only requires local information exchange. In the aspect of multi-fleet operation, the problem is a variant of the congestion game, which requiring further studies on the existence of pure strategic equilibrium and search method. It is also important to perform a potential research on the multi-fleet truck platooning coordination with real world demand data and weights setting. If the potential is considerably better than pure opportunistic platooning among fleets, a more detailed study on an asynchronous framework for privacy-preserved multi-fleet coordination is supposed to be launched.

Goals of this thesis	Review on the corresponding chapters
A framework for solving truck platooning coordination as a non-cooperative coordination problem for self-interested fleets.	In Chapter 2, we model the interaction among fleets as a non-cooperative finite game and propose the best-response search algorithm. The equilibrium searching problem is broken down into subproblems, which are optimization problems of cooperative truck platooning coordination.
A mathematical model for cooperative truck platooning, in which cost functions consist of different cost other than simply fuel cost and while routes, speed, types of trucks and schedule for trucks are decision variables to optimize.	In Chapter 2, a spatiotemporal road network model is introduced and the plans of trucks are presented by vectors with binary elements.The vector is known as the decision vector, together with the spatiotemporal network, which may present routes and speed choices of the truck. Based on the network model and the decision vector, we formulate the cost function of a fleet, which contains fuel cost, time cost and schedule preference penalty. In Chapter 3, the subproblem is reformulated with relaxations into a standard MILP form. The performance with a off-the self solver is also given in this chapter.
A decentralized algorithm for solving the multi-fleet truck platooning coordination problem.	A decentralized method for solving the best-response subproblem, which is in MILP form is given in Chapter 4. We introduce the principle of the algorithm and test it with a medium size problem to compare with a centralized solver.
A simulation with real-world featured demand input to test the performance of the decentralized algorithm.	In Chapter 5, we introduced the way we generated the demand set based on the dutch highway network and the distribution of three supermarket chains. The simulation is performed to test the time complexity and cost reduction. The results suggest that in this specific setting, the propsed approach may bring computation time reduction if the fleet size is relatively large, while the best-response search does not deliver extra benefits.

Table 6.1: Review on the goals and corresponding chapters

6.2 Future Work

The work of this thesis may still be improved in three aspects. One of the significant efforts in this thesis is to build a representative model, in which the cost function contains the concerned parts of real shake-holders. Nevertheless, in a problem with considerable speed options, we may have to build the spatiotemporal with short time intervals or short unit distance. The micro-scale movements of trucks for forming platoons, which can be ignored on a larger scale, may bring extra costs. The inefficient process of forming platoons is not yet modeled, and maybe the further studied for a more accurate model.

Secondly, the convergence of the best-response search is not yet studied in the area of game theory. Although the truck platooning coordination game is based on a congestion game, the finite pure strategy profile prevents easy proof of the equilibrium. Related works in game theory are still ongoing. However, on the other hand, the negotiation process among non-cooperative players is noted by many researchers. In a scenario like a supermarket chain gaming, the process may be repeated, which brings the opportunity of a protocol pursuing a lower cost, even no equilibrium exists. Building a privacy-preserving protocol is likely of the interest of the real stake-holders.

In the aspect of the algorithm itself, the strict assumption on synchronous operations is less favorable for real users. Mostly, solving the subproblem and reaching consensus are two distributed processes, which both may operate in an asynchronous set-up. However, there remains a problem with the influence on each other if they work simultaneously. Ideally, if the convergence is guaranteed, the algorithm may be fully distributed and taking more advantages on the connected vehicle's technologies.

Bibliography

[1] Peleton. [Online]. Available: https://peloton-tech.com/how-it-works/

[2] "Albert heijn winkels in nederland." [Online]. Available: https://www.ah.nl/winkels

[3] "winkelvinder jumbo supermarket." [Online]. Available: https://www.jumbo.com/winkels

[4] "Spar winkels in nederland." [Online]. Available: https://www.spar.nl/winkels/

[5] A. Schroten, G. Warringa, and M. Bles, "Marginal abatement cost curves for heavy duty vehicles," *Background Report. CE Delft, Delft*, 2012.

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

[7] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on automatic control*, vol. 38, no. 2, pp. 195–207, 1993.

[8] S. Tsugawa, "Results and issues of an automated truck platoon within the energy its project," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 642–647.

[9] B. McAuliffe, M. Croken, M. Ahmadi-Baloutaki, and A. Raeesi, "Fuel-economy testing of a three-vehicle truck platooning system," 2017.

[10] L. Zhang, F. Chen, X. Ma, and X. Pan, "Fuel economy in truck platooning: A literature overview and directions for future research," *Journal of Advanced Transportation*, vol. 2020, 2020.

[11] A. K. Bhoopalam, N. Agatz, and R. Zuidwijk, "Planning of truck platoons: A literature review and directions for future research," *Transportation research part B: methodological*, vol. 107, pp. 212–228, 2018.

[12] J. Larson, C. Kammer, K.-Y. Liang, and K. H. Johansson, "Coordinated route optimization for heavy-duty vehicle platoons," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1196–1202.

[13] J. Larson, K.-Y. Liang, and K. H. Johansson, "A distributed framework for coordinated heavy-duty vehicle platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 419–429, 2014.

[14] E. Larsson, G. Sennton, and J. Larson, "The vehicle platooning problem: Computational complexity and heuristics," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 258–277, 2015.

[15] A. Nourmohammadzadeh and S. Hartmann, "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, 2016, pp. 46–57.

[16] W. Zhang, E. Jenelius, and X. Ma, "Freight transport platoon coordination and departure time scheduling under travel time uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 98, pp. 1–23, 2017.

[17] V. Sokolov, J. Larson, T. Munson, J. Auld, and D. Karbowski, "Maximization of platoon formation through centralized routing and departure time coordination," *Transportation Research Record*, vol. 2667, no. 1, pp. 10–16, 2017.

[18] A. Nourmohammadzadeh and S. Hartmann, "Fuel efficient truck platooning with time restrictions and multiple speeds solved by a particle swarm optimisation," in *International Conference on Theory and Practice of Natural Computing.* Springer, 2018, pp. 188–200.

[19] N. Boysen, D. Briskorn, and S. Schwerdfeger, "The identical-path truck platooning problem," *Transportation Research Part B: Methodological*, vol. 109, pp. 26–39, 2018.

[20] A. Johansson, E. Nekouei, K. H. Johansson, and J. Mårtensson, "Multi-fleet platoon matching: A game-theoretic approach," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC).* IEEE, 2018, pp. 2980–2985.

[21] A. Johansson and J. Mårtensson, "Game theoretic models for profit-sharing in multi-fleet platoons," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC).* IEEE, 2019, pp. 3019–3024.

[22] M. Abdolmaleki, M. Shahabi, Y. Yin, and N. Masoud, "Itinerary planning for cooperative truck platooning," *Available at SSRN 3481598*, 2019.

[23] F. Farokhi, I. Shames, and K. H. Johansson, "Private and secure coordination of match-making for heavy-duty vehicle platooning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7345–7350, 2017.

[24] N. Masoud and R. Jayakrishnan, "A decomposition algorithm to solve the multi-hop peer-to-peer ride-matching problem," *Transportation Research Part B: Methodological*, vol. 99, pp. 1–29, 2017.

[25] J. Nash, "Non-cooperative games," *Annals of mathematics*, pp. 286–295, 1951.

[26] A. Sureka and P. R. Wurman, "Using tabu best-response search to find pure strategy nash equilibria in normal form games," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 1023–1029.

[27] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science (sfcs 1975).* IEEE, 1975, pp. 184–193.

[28] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[29] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[30] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278 – 305, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1367578819300082

[31] A. Falsone, K. Margellos, and M. Prandini, "A decentralized approach to multi-agent milps: finite-time feasibility and performance guarantees," *Automatica*, vol. 103, pp. 141–150, 2019.

[32] R. Vujanic, P. M. Esfahani, P. J. Goulart, S. Mariéthoz, and M. Morari, "A decomposition method for large scale milps, with performance guarantees and a power system application," *Automatica*, vol. 67, pp. 144–156, 2016.

[33] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," *lecture notes of EE392o, Stanford University, Autumn Quarter*, vol. 2004, pp. 2004–2005, 2003.

[34] Q. Li, I. Cascudo, and M. G. Christensen, "Privacy-preserving distributed average consensus based on additive secret sharing," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[35] Q. Li, R. Heusdens, and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5895–5899.

[36] "Netherlands gasoline prices." [Online]. Available: https://www.globalpetrolprices.com/Netherlands/gasoline_prices/

[37] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektaş, and G. Laporte, "The time-dependent pollution-routing problem," *Transportation Research Part B: Methodological*, vol. 56, pp. 265 – 293, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0191261513001446

[38] S. Tsugawa, S. Kato, and K. Aoki, "An automated truck platoon for energy saving," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4109–4114.

[39] G. Janssen, J. Zwijnenberg, I. Blankers, and J. de Kruijff, "Truck platooning: Driving the future of transportation," 2015.