Multi-Trip Multi-Trailer Drop-and-Pull Container Drayage Problem

Decheng Wang, Ilkyeong Moon, and Ruiyou Zhang, *Member, IEEE*

***Abstract*— This paper addresses a novel and interesting container drayage problem in which a tractor can pull multiple trailers and can perform multiple trips during a working day. A state-based method and an IF-THEN method are proposed to model the multi-trip feature. The problem is mathematically formulated as two mixed-integer programming models. Three families of valid inequalities are introduced to strengthen the models. An adaptive large neighborhood search (ALNS) algorithm with an embedded decoding mathematical model is proposed to solve realistic-sized instances of the problem. Numerical experiments based on a large number of benchmark instances indicate that the inequalities are valid and that the model based on IF-THEN constraints shows excellent performance. The ALNS algorithm outperforms the strengthened mathematical model considering large-sized instances. The multi-trailer mode is more suitable if customers are clustered in the area far away from the depot when compared to the single-trailer mode.**

***Index Terms*— Container drayage, multi-trip multi-trailer drop-and-pull transportation, valid inequality, adaptive large neighborhood search.**

I. Introduction

CONTAINER drayage refers to the truckload transportation activities around the depot. These activities are mainly the pre- and end-haulage activities including reposition of empty containers between the depot, exporters, and importers, and are typically performed by trucks [1]–[5]. Container drayage can provide door-to-door services, while the other transportation means (e.g., by vessels or by trains) usually cannot [6]. Despite this merit, this transportation mode still has to face the inescapable high costs contributed by investments on purchasing trucks which play a significant role in economic considerations. Recently, the reality of high truck-purchasing costs became insupportable, which called for an urgent reaction to the logistic mode. Deploying trucks to undertake more than one trip is a practical and popular option because this mode can substantially decrease the required size of truck fleets [7], and increase utilization rate of vehicles [8].

Fig. 1. One example of a road train in Australia (https://medium.com/knowledge-stew/the-incredible-australian-road-trains-6c41c7fb3ed7).

A truck typically consists of two parts: a motorized part that can move by itself (i.e., a tractor) and a non-motorized part (i.e., trailers or containers) that needs to be pulled by the motorized part. As a result, the transportation modes of container drayage can be classified as drop-and-pull (D&P) mode and stay-with mode. In the D&P mode, the motorized part can detach itself from the non-motorized part, park the non-motorized part at customer sites for packing/unpacking operations, and leave to carry out some other transportation tasks. After the packing/unpacking operations are finished, the motorized part comes back to pull the non-motorized part away. The detaching/attaching activities can be conveniently accomplished with a negligible operation time. In the stay-with mode, however, the two parts are not separable [9]–[12]. Compared to the stay-with mode, the D&P mode can save transportation costs by one quarter, on average [13]. Moreover, the D&P mode, which allows one tractor to pull two trailers, can decrease transportation costs by about 30% [14], compared to the one-tractor-one-trailer version. All these results validate the D&P mode and stimulate us to further study it.

The utilization of multi-trailer tractors, the so-called long combination vehicles [15] (see Fig. 1 for the configuration), was found to be significantly efficient from an economic standpoint [16]. In Australia, such long combination vehicles are called road trains, and they are considered to be the best way to move freight, because the Outback terrain of Australia is very barren and vast. Most articles addressing the D&P container drayage focus on the one-tractor-one-trailer mode [13], [17]–[19]. As far as we know, Ref [14] is the only article investigating the one-tractor-multi-trailer container drayage problem. However, the authors of [14] assumed that a tractor can only perform a single trip in a planning horizon. We, in this paper, address a multi-trip multi-trailer D&P container drayage (MTMTDPD) problem. Differently from most multi-trip problems (e.g., Ref [20]), vehicle routes in

Manuscript received 10 August 2021; revised 7 January 2022; accepted 28 February 2022. Date of publication 22 March 2022; date of current version 11 October 2022. This work was supported in part by the Natural Science Foundation of Guangxi under Grant 2021AC19255; and in part by the National Natural Science Foundation of China under Grant 71971050, Grant 71831006, and Grant 61762031. The Associate Editor for this article was N. Bekiaris-Liberis. *(Corresponding author: Ruiyou Zhang.)*

Decheng Wang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the College of Information Science and Engineering, Guilin University of Technology, Guilin 541006, China (e-mail: decheng_wang@glut.edu.cn).

Ilkyeong Moon is with the Department of Industrial Engineering and the Institute for Industrial Systems Innovation, Seoul National University, Seoul 08826, Republic of Korea (e-mail: ikmoon@snu.ac.kr).

Ruiyou Zhang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: zhangruiyou@ise.neu.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3156547

the MTMTDPD problem are interrelated because different vehicles might be involved in fulfilling precedence-required tasks. A change in one route usually causes changes in the other routes, and even make the routing and scheduling infeasible. Moreover, differently from most D&P research such as [18], [19], we provide freedom for tractors to flexibly choose their departure time from the depot. This setting can make the vehicle arrive at the customer locations at an appropriate time, and thus reduce the unnecessary waiting [21], and increase the utilization of vehicles [14].

Compared to the problem in [14], the MTMTDPD is an important and interesting problem not only from a practical viewpoint. Its importance from a theoretical viewpoint is also significant, and this makes it particularly attractive for conducting a systematic study. First, in the MTMTDPD, frequent dropping off and picking up of trailers at the depot is a setup that is needed in order to perform multiple trips, some of which are consecutive. Therefore, the number and type of trailers that should be dropped off or picked up from the depot are jointly determined by the trips. The combinatorial feature brought about by the multi- trailer tractors complicates the dropping and pulling operations and imposes theoretical challenges on the multi-trip issue (see later Section III.B). Second, modeling the multi-trip feature is difficult because the number of trips performed by vehicles is not easy to determine in advance. The standard method in existing literature is to introduce the maximum number of trips that the vehicle can perform. This number usually influences the construction of transportation plans. Moreover, this number increases the model size if a large value is assigned. In this study, we have designed a state-based method and an IF-THEN method to handle this issue. A third and most important point is that, given the interdependence of routes, obtaining the total working time of a transportation plan is difficult. Compared to the three-stage heuristic method used to handle a similar problem in [14], we developed in this study a mathematical model to solve this problem exactly.

In general, the contributions of this paper can be summarized as follows.

1) An MTMTDPD problem is formally presented. Different attempts as state-based method and IF-THEN method are conducted to handle the multi-trip issue accurately from the mathematical level. Two mixed-integer mathematical programming models are constructed accordingly.
2) Three families of valid inequalities are presented to strengthen the mathematical models. Extensive experiments are conducted to validate the inequalities and verify their effectiveness in improving the upper and lower bounds of the problem.
3) An adaptive large neighborhood search (ALNS) algorithm is developed, because even the strengthened mathematical models still cannot solve realistic-sized instances of the problem. In the ALNS algorithm, a model-based decoding method is proposed to obtain the optimal working time of tractors for given visiting sequences of customers. Numerical experiments validated the ALNS algorithm and provided several operational insights.

The remainder of this paper is organized as follows. Section II reviews the related literature from two viewpoints. Section III describes the MTMTDPD problem formally. Section IV presents two mathematical models to formulate the MTMTDPD problem. Section V introduces three families of valid inequalities. Section VI develops the ALNS algorithm. Computational results are analyzed in Section VII. Finally, Section VIII concludes this paper.

II. Literature Review

Vehicle routing problems involving pickups and deliveries can be grouped into two classes: transportation of freight from the depot to some customers and from some other customers to the depot, and transportation of freight and people between pickup and delivery locations [22]. The so-called pickup-and-delivery problem (PDP) and dial-a-ride problem (DARP) belong to the second class, and usually focus on the less-than-truckload transportation setup. The PDP handles the transportation of freight. However, the DARP deals with the transportation of people. Therefore, the DARP considers client-centric or service-based constraints (e.g., high-quality service, reduction of idle time, and maximum riding time). For example, Cordeau [23] addressed a DARP with a maximum ride time of passengers. The MTMTDPD problem belongs to the first class and deals with the truckload transportation under the D&P mode. Therefore, we focus on the relevant literature from two aspects — as multi-trip routing problems and as D&P routing problems — hereafter in this section.

A. Multi-Trip Routing Problems

Multi-trip routing problems have been studied in different application fields. The first field is the people transportation. For example, Tang *et al.* studied a multi-trip problem of providing pickup and delivery services for customers traveling to airports [24]. Liu *et al.* presented a multi-trip DARP arising in the transportation of elderly or disabled people [25]. They considered heterogeneous vehicles, single depot, and configurable vehicle capacity. Some other researchers investigated specific aspects of the DARP. For instance, Masmoudi *et al.* concerned a multi-depot multi-trip routing problem [26]. Zhang *et al.* considered the sterilization of ambulances between trips and lunch break for emergency management technicians [27]. Recently, Liu *et al.* addressed a multi-trip scheduling problem for multiple repairmen [28].

Routing problems regarding freight transportation constitute the most widely studied field of multi-trip problems. Most such studies focus on the classical vehicle routing problem (VRP) concerning less-than-truckload transportation. A large number of research on the multi-trip VRP [29] and its variants accommodating practical requirements such as distribution of emergency aid [30], uncertain travel times [31], release dates [32], [33], heterogeneous vehicles [20], [34], and time windows [35]–[37], have been extensively conducted. We cannot survey all the multi-trip VRP research here because

TABLE I
COMPARISON BETWEEN THIS STUDY AND THE RELATED MULTI-TRIP RESEARCH

Problem	Articles	Multi-trip	Routes interdependence	D&P
People transportation	Liu et al. [25]	√	√	
	Cordeau [23], Tang et al. [24], Zhang et al. [27], Masmoudi et al. [26], Liu et al. [28]	√		
VRP (less-than-truckload transportation)	François et al. [29], Hernandez et al. [35], Cattaruzza et al. [32], Coelho et al. [20], Moreno et al. [30], Wang et al. [31], Zhen et al. [33], Sethanan and Jamrus [34], Huang et al. [36], Pan et al. [37]	√		
Container drayage (truckload transportation)	Marković et al. [39], Zhang et al. [40], Zhang et al. [41], Bruglieri et al. [10]	√		
	This paper	√	√	√

of the limited space. Interested readers can refer to Ref [38] for a unified review.

Multi-trip container drayage problems, which normally concern the truckload transportation, have also been studied. For example, Marković *et al.* studied a truck-train intermodal transportation problem [39]. Zhang *et al.* addressed a container drayage problem with flexible orders based on a so-called determined-activities-on-vertex graph [40]. Zhang *et al.* proposed a state-transition method to handle a multi-size container drayage problem in which a truck might carry multiple containers [41]. Recently, Bruglieri *et al.* studied a multi-period container drayage problem with release and due dates requirements [10]. One common element in the aforementioned research is that the drayage is operated in the stay-with mode. However, we present a multi-trip drayage problem under the D&P mode. Table I presents a comparison between the studies on the multi-trip routing problems.

B. D&P Routing Problems

The D&P mode has been first applied in the truck-and-trailer routing problem (TTRP) which focuses on less-than-truckload transportation [42]. In the TTRP, a vehicle is composed of a truck and a trailer. The trailer is detachable from the truck. Cargoes can be shifted between the truck and the trailer, which is beyond the range of this research. References [43] and [44] have provided detailed overviews for the TTRPs. Therefore, we focus on the D&P problems in the other fields here.

A few studies present container drayage under the D&P mode, in most of which one tractor can pull at most one trailer. For example, Xue *et al.* formulated a mixed-integer programming model for a local container drayage problem [13]. A synchronized transportation routing problem with limited transportation resources and single trip was presented and solved using an ALNS algorithm in [17]. It was further solved with a branch-and-price-and-cut algorithm [19]. Furthermore, Song *et al.* adopted a branch-and-price-and-cut algorithm to solve a similar problem with multiple trips [18]. They required that each drayage task must be accomplished by the same tractor so that the vehicle routes do not depend on each other.

Quite few researches present multi-trailer D&P container drayage problems. Zhang *et al.* investigated a foldable container drayage problem, in which the operational mode is a little similar to the D&P mode at least from the mathematical perspective [47]. A truck can carry four or even six folded containers or only one loaded container. Moghaddam *et al.* proposed a generalized model for drayage with heterogeneous fleet, multi-container sizes and D&P transportation [4]. The model is effective to find high-quality solutions for small- and medium-sized instances. However, one task can only be handled by the same truck, and the truck cannot engage in multi-trip services. A similar problem with time windows was studied and solved by using a genetic algorithm [46]. You *et al.* addressed a so-called truck platooning problem, in which a human-driven truck can lead several automated trucks using the cooperative adaptive cruise control technique [7]. Quite recently, Xue *et al.* further analyzed the effects of the truck platooning on fuel consumptions [45]. Furthermore, Zhang *et al.* formulated a multi-trailer D&P container drayage problem into a mixed-integer linear programming model and solved it with a backtracking adaptive threshold accepting algorithm [14]. The problem in [14] is the most similar one to this research but considers single trip only. Table II summarizes the similarities and differences between the D&P studies.

III. MULTI-TRIP MULTI-TRAILER DROP-AND-PULL CONTAINER DRAYAGE

A. Problem Description

A trucking company provides container drayage services in a local area using a depot, a number of homogeneous tractors, trailers and containers. Each trailer can load exactly one container. The container stays with the trailer as one entire entity. Each tractor can drag up to $K (\geq 2)$ trailers. The tractors and trailers operate in the D&P mode. Specifically, a tractor picks up a number of trailers at the depot and drops the trailers at the customer's designated location. The tractor then returns to the depot after possibly picking up some other trailers. At the customer's designated location, it is unnecessary for the tractor that picks up the trailer to be the tractor that drops off the trailer. All tractors and trailers originally park at the depot, and should finally return to the depot after finishing drayage services. A *trip* is defined as the drayage activities of a tractor, since the tractor departs from the depot until it returns to the depot. A tractor might perform a number of trips during a planning horizon. A series of trips performed by a tractor form a *route*.

TABLE II
COMPARISON BETWEEN THIS STUDY AND THE RELATED D&P RESEARCH

Problem	Articles	Item detachable	No. of loadable items			Multi-trip	Routes Interdependence	Mixed transport [a]	Freedom [b]
			= 1	= 2	> 2				
Vehicle platooning (semi-automated driving)	You et al. [7]	Automated truck	√	√	√	√	√	√	
	Xue et al. [45]	Automated truck	√	√	√			√	
TTRP (less-than-truckload transportation)	Chao [42]	Trailer	√					√	
Container drayage (truckload transportation)	Moghaddam et al. [4]	Standard container	√	√				√	
	Yang et al. [46]	Standard container	√	√				√	
	Zhang et al. [47]	Foldable container	√[c]	√[d]	√[d]	√	√		√
	Xue et al. [13]	Trailer	√			√	√		
	Meisel and Kopfer [17]	Trailer	√				√		
	Song et al. [18]	Trailer	√			√			
	Tilk et al. [19]	Trailer	√				√		
	Zhang et al. [14]	Trailer	√	√	√		√	√	√
	This paper	Trailer	√	√	√	√	√	√	√

[a] Simultaneous transportation of empty and full containers
[b] Freedom for vehicles to choose departure time from the depot
[c] Only one container can be transported if it is full
[d] Only multiple empty foldable containers can be transported simultaneously

According to the flow directions of full containers, drayage services are classified into inbound orders and outbound orders. The full containers of inbound orders need to be delivered from the depot to their receivers, while full containers in outbound orders are delivered from their shippers to the depot. Each order corresponds to a customer, and is divided into a Phase I suborder and a Phase II suborder according to the feature of the D&P mode. Let I_1 and I_2 be the sets of Phase I and Phase II suborders of inbound orders, respectively. A suborder $i \in I_1$ requires a tractor to pick up a trailer with an inbound full container (an inbound full trailer, or an IFT) at the depot and deliver it to its receiver, where the container is unpacked. A suborder $i \in I_2$ requires a tractor to pick up the trailer with the emptied container (an empty trailer, or an ET) and take it to the depot, or directly to the shipper of outbound orders for future use. Similarly, let O_1 and O_2 be the sets of Phase I and Phase II suborders of outbound orders, respectively. A suborder $i \in O_1$ requires a tractor to deliver an ET to the shipper where the freight is packed into the container. A suborder $i \in O_2$ requires a tractor to pick up the trailer with the outbound full container (an outbound full trailer, or an OFT) and take it back to the depot. For a Phase I suborder i, the corresponding Phase II suborder is $\delta(i)$. Each Phase I suborder i has to be handled before its corresponding Phase II suborder $\delta(i)$. However, these two suborders might be served by different tractors.

All orders should be finished within a planning horizon H. For suborder $i \in I_1 \cup O_1$, the container packing/unpacking time is $p_i\ (\geq 0)$, while suborder $j \in I_2 \cup O_2$ has no packing/unpacking operations. Let the initial start from and final return to the depot be a virtual suborder (say Suborder 0). The traveling time of tractors with or without trailers between the locations of any two suborders i and j is $\tau_{ij}\ (\geq 0)$. The location of Suborder 0 is the depot. The location of the two suborders of an inbound order is its receiver. The location of the two suborders of an outbound order is its shipper. For any i, j, and k, the triangular inequality holds (i.e., $\tau_{ij} + \tau_{jk} \geq \tau_{ik}$).

The MTMTDPD problem looks for a set of routes to finish all the drayage orders in time period $[0, H]$ with the minimum total costs. The total costs consist of the fixed costs of employing tractors and the variable total working time costs of fulfilling drayage services. The total working time is the summation of time intervals of all tractors. The time interval of a tractor is the period from when it departs from the depot to perform its first trip until it returns to the depot after performing all its trips. Such a definition can reflect the variable costs and has been widely used (e.g., [48]).

B. Dropping and Pulling of Trailers at the Depot

We explain the dropping and pulling of trailers at the depot in detail in this subsection for the convenience of mathematical modeling. Dropping and pulling of trailers at the depot is sometimes required in order to perform a trip. For example, if a tractor without an IFT is going to serve a suborder of I_1, it has to go back to the depot to pull the IFT (Fig. 2 (a)). Similarly, a tractor without an ET has to go back to the depot to pull the ET to serve a suborder of O_1 in the next trip (Fig. 2 (b)). On the contrary, if a tractor with K trailers (i.e., the capacity is reached) is going to finish a Phase II suborder, it has to return to the depot to drop off some trailers first (Fig. 2 (c)).

For the one-tractor-multi-trailer transportation setup, the dropping and pulling of trailers at the depot is rather complicated, because the number and type of trailers to be dropped or pulled are jointly determined by the connected trips.

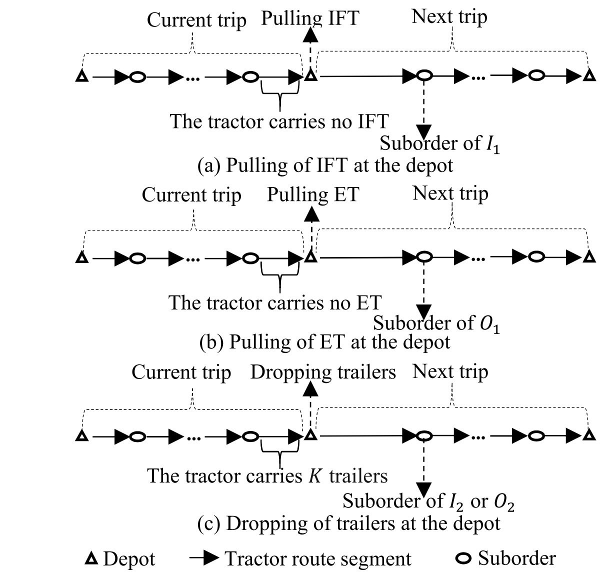


Fig. 2. Dropping and pulling of trailers between two trips.

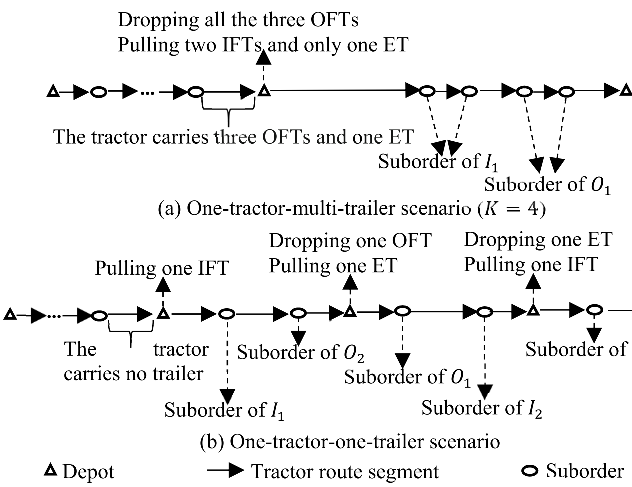


Fig. 3. Dropping and pulling operations.

Fig. 3 (a) illustrates a multi-trailer example in which $K = 4$. After serving some suborders, a tractor is carrying four trailers (i.e., one ET and three OFTs), but two suborders of I_1 and two suborders of O_1 need to be served. In this situation, the tractor has to go back to the depot to drop all three OFTs and pull two IFTs and one ET because it already carries one ET. Because the serving sequence of suborders cannot be predetermined, the trips as well as the number and type of trailers to be dropped and pulled cannot be determined in advance. All these are left as decisions to be made and create challenges. As a comparison, in the one-tractor-one-trailer case (e.g., [18]), at most one trailer is dropped and at most one trailer is pulled each time a tractor goes to the depot (see Fig. 3(b)). The type of trailers to be dropped or pulled can be easily determined. For example, after a tractor with an IFT serves the suborder of I_1, it carries no ET before serving the suborder of E_1. The tractor has to go back to the depot to carry an ET.

TABLE III

COMMON DECISION VARIABLES AND PARAMETERS OF MODELS SB AND LC

Decision variables	
x_{ij}	1 if arc $(i,j) \in A$ is used or 0 otherwise
y_{ij}^I	Number of IFTs that the tractor transferring on arc $(i,j) \in A$ is carrying. $y_{ij}^I \in \{0,1,\cdots,K\}$.
y_{ij}^O	Number of OFTs that the tractor transferring on arc $(i,j) \in A$ is carrying. $y_{ij}^O \in \{0,1,\cdots,K\}$.
y_{ij}^E	Number of ETs that the tractor transferring on arc $(i,j) \in A$ is carrying. $y_{ij}^E \in \{0,1,\cdots,K\}$.
s_i	Service starting time of node $i \in N\backslash\{0\}$. $\tau_{0i} \le s_i \le H - \tau_{i0}$.
t_{ij}	Transfer time of arc $(i,j) \in A^{TRF}$. $t_{ij} \ge 0$.
Parameters	
ρ_1	Fixed cost of the tractor
ρ_2	Cost in unit working time

TABLE IV

AUXILIARY VARIABLES INTRODUCED IN MODEL SB

x_{ij}^I	1 if the tractor transferring on arc $(i,j) \in A$ is carrying some IFTs or 0 otherwise
x_{ij}^E	1 if the tractor transferring on arc $(i,j) \in A$ is carrying some ETs or 0 otherwise
x_{ij}^T	1 if the tractor transferring on arc $(i,j) \in A$ is carrying K trailers or 0 otherwise

IV. MIXED-INTEGER PROGRAMMING MODELS

The MTMTDPD problem can be defined as a directed graph $G = (N, A)$, where $N = \{0\} \cup I_1 \cup I_2 \cup O_1 \cup O_2$ is the set of nodes and $A = \{(0,i) | i \in N\backslash\{0\}\} \cup \{(i,0) | i \in N\backslash\{0\}\} \cup A^{TRF}$ is the set of arcs. In A, $A^{TRF} = \{(i,j) | i \in N\backslash\{0\}, j \in N\backslash\{0,i\}\} A^{RVS}$ is a set of *intermediate* arcs, where $A^{RVS} = \{(\delta(i), i) | i \in I_1 \cup O_1\}$ is a set of *reverse* arcs that transfer from Phase II suborder node to its corresponding Phase I suborder node. The reverse arcs are excluded because no Phase II suborder could be accomplished prior to its corresponding Phase I suborder. Hereafter, the terms suborder, suborder node and node are used interchangeably where no confusion is involved.

Based on Graph G, we formulate the problem as a so-called state-based model (**Model SB**) and a logical-constraint model (**Model LC**, see Appendix A). The common decision variables and parameters used in the two models are listed in Table III. In Model SB, three indicators, x_{ij}^I, x_{ij}^E and x_{ij}^T were introduced to formulate the state of tractors (i.e., the number and type of trailers that a tractor is carrying) as shown in Table IV. They are defined as auxiliary variables to formulate trailer number variations and transfer times of arcs.

A. Objective Function

$$\min \rho_1 \sum_{i \in N\backslash\{0\}} x_{0i} + \rho_2 \left(\sum_{i \in N\backslash\{0\}} x_{i0}\,(s_i + \tau_{i0}) - \sum_{i \in N\backslash\{0\}} x_{0i}\,(s_i - \tau_{0i}) \right) \quad (1)$$

Objective function (1) minimizes the total costs, where ρ_1 and ρ_2 are the weighting coefficients. $\sum_{i\in N\backslash\{0\}} x_{0i}$ calculates the number of employed tractors. $\sum_{i\in N\backslash\{0\}} x_{i0}\left(s_i+\tau_{i0}\right)$ is the summation of the time points when tractors finally return to the depot. $\sum_{i\in N\backslash\{0\}} x_{0i}\left(s_i-\tau_{0i}\right)$ is the summation of the time points when tractors initially leave the depot.

B. Basic Constraints

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}=1,\quad \forall j\in N\backslash\{0\} \tag{2}$$

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}-\sum_{i\in\overleftarrow{A^{-1}}(j)} x_{ji}=0,\quad \forall j\in N\backslash\{0\} \tag{3}$$

$$s_i+t_{ij}-s_j\le M\left(1-x_{ij}\right),\quad \forall (i,j)\in A^{TRF} \tag{4}$$

$$s_i+p_i-s_{\delta(i)}\le 0,\quad \forall i\in I_1\cup O_1 \tag{5}$$

$$y^I_{ij}+y^O_{ij}+y^E_{ij}\le Kx_{ij},\quad \forall (i,j)\in A \tag{6}$$

$$\sum_{i\in N\backslash\{0\}} y^I_{i0}+\sum_{i\in N\backslash\{0\}} y^O_{0i}=0 \tag{7}$$

Constraint (2) guarantees that each suborder is served exactly once, where $\overrightarrow{A^{-1}}(j)=\{i\,|\,(i,j)\in A\}$. Constraint (3) ensures that the in-degree of any suborder node is equal to its out-degree, where $\overleftarrow{A^{-1}}(j)=\{i\,|\,(j,i)\in A\}$. Constraint (4) establishes the relationship of the service starting times for successively served suborders and eliminates sub-tours, where $M=H-\min_{i\in N\backslash\{0\}}\tau_{i0}$ is a sufficiently large positive constant. Constraint (5) imposes the precedence constraints. Constraint (6) imposes the capacity constraints. Constraint (7) prevents tractors from taking IFTs back to the depot at the end of services, and taking OFTs out of the depot at the beginning of services.

C. State Construction

$$y^I_{ij}/K\le x^I_{ij}\le y^I_{ij},\quad \forall (i,j)\in A \tag{8}$$

$$y^E_{ij}/K\le x^E_{ij}\le y^E_{ij},\quad \forall (i,j)\in A \tag{9}$$

$$y^I_{ij}+y^O_{ij}+y^E_{ij}-K+1\le x^T_{ij}\le\left(y^I_{ij}+y^O_{ij}+y^E_{ij}\right)/K,\ \forall (i,j)\in A \tag{10}$$

Constraints (8) and (9) construct x^I_{ij} and x^E_{ij}, respectively. In Constraint (8), if a tractor carries some IFTs, i.e., $y^I_{ij}>0$, we have $0<x^I_{ij}\le y^I_{ij}$, and the domain of x^I_{ij} forces it to be one; otherwise, x^I_{ij} is forced to be zero. Similarly, Constraint (9) makes $x^E_{ij}=1$ if $y^E_{ij}>0$; or $x^E_{ij}=0$ otherwise. Constraint (10) constructs x^T_{ij}. Specifically, if $y^I_{ij}+y^O_{ij}+y^E_{ij}=K$, both sides are equal to one and hence $x^T_{ij}=1$; otherwise, we have $y^I_{ij}+y^O_{ij}+y^E_{ij}-K+1\le 0$ and $\left(y^I_{ij}+y^O_{ij}+y^E_{ij}\right)\Big/K<1$, and the domain of x^T_{ij} forces it to be zero.

D. Trailer Number Formulation Considering I_1

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y^O_{ij}-K\left(1-\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}\right)\le\sum_{k\in\overleftarrow{A^{-1}}(j)} y^O_{jk}\le K\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij},\quad \forall j\in I_1 \tag{11}$$

$$\sum_{k\in\overleftarrow{A^{-1}}(j)} y^O_{jk}-\sum_{i\in\overrightarrow{A^{-1}}(j)} y^O_{ij}\le 0,\quad \forall j\in I_1 \tag{12}$$

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y^I_{ij}-1-K\left(1-\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}\right)\le\sum_{k\in\overleftarrow{A^{-1}}(j)} y^I_{jk}\le K\left(1-\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}\right)+\sum_{i\in\overrightarrow{A^{-1}}(j)} y^I_{ij}-1,\quad \forall j\in I_1 \tag{13}$$

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y^E_{ij}-K\left(1-\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}\right)\le\sum_{k\in\overleftarrow{A^{-1}}(j)} y^E_{jk}\le (K-1)\left(1-\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}\right)+\sum_{i\in\overrightarrow{A^{-1}}(j)} y^E_{ij},\quad \forall j\in I_1 \tag{14}$$

$$\sum_{k\in\overleftarrow{A^{-1}}(j)} y^E_{jk}+\sum_{k\in\overleftarrow{A^{-1}}(j)} y^I_{jk}\le K-1+K\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij},\quad \forall j\in I_1 \tag{15}$$

Constraints (11)-(15) formulate the number of trailers when a suborder $j\in I_1$ is served. Note that if a tractor carries some IFTs (i.e., $\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}=1$), it can go directly to serve suborder j; otherwise, it has to return to the depot before serving suborder j. Specifically, if $\sum_{i\in\overrightarrow{A^{-1}}(j)} x^I_{ij}=1$, the right side of Constraint (11) becomes K and hence is relaxed. Constraint (12) and the left side of Constraint (11) make $\sum_{k\in\overleftarrow{A^{-1}}(j)} y^O_{jk}$ equal to $\sum_{i\in\overrightarrow{A^{-1}}(j)} y^O_{ij}$ (i.e., the number of OFTs remains when a tractor leaves suborder j). Both sides of Constraint (13) become $\sum_{i\in\overrightarrow{A^{-1}}(j)} y^I_{ij}-1$, and we have $\sum_{k\in\overleftarrow{A^{-1}}(j)} y^I_{jk}=\sum_{i\in\overrightarrow{A^{-1}}(j)} y^I_{ij}-1$ (i.e., the number of IFTs decreases by one). Similarly, Constraint (14) makes that $\sum_{k\in\overleftarrow{A^{-1}}(j)} y^E_{jk}=\sum_{i\in\overrightarrow{A^{-1}}(j)} y^E_{ij}$. Constraint (15) becomes $\sum_{k\in\overleftarrow{A^{-1}}(j)} y^E_{jk}+\sum_{k\in\overleftarrow{A^{-1}}(j)} y^I_{jk}\le 2K-1$ and is relaxed.

If $\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{I} = 0$, the left side of Constraint (11) becomes $\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{O} - K$, and hence is relaxed. The right side of Constraint (11) and the domain of y_{jk}^{O} restrict that $\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} = 0$ (i.e., no OFTs are carried when a tractor leaves suborder j). Consequently, Constraint (12) is relaxed. The left side of Constraint (13) is relaxed, while the right side imposes an upper bound $K-1$ on $\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{I}$. The reason is that at most K IFTs might be pulled at the depot but one IFT must be dropped off for suborder $j \in I_1$. The left side of Constraint (14) is relaxed, and the right side becomes $K-1+\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{E}$ and is also relaxed. The reason is that at least one IFT must be pulled at the depot in order to serve suborder $j \in I_1$, and hence at most $K-1$ ETs can be pulled. As a result, the total number of IFTs and ETs will not exceed $K-1$ when a tractor leaves suborder $j \in I_1$ (Constraint (15)).

E. Trailer Number Formulation Considering O_1

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{O} - K\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}\right) \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} \leq K\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}, \quad \forall j \in O_1 \tag{16}$$

$$\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} - \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{O} \leq 0, \quad \forall j \in O_1 \tag{17}$$

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{I} - K\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}\right) \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{I} \leq (K-1)\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}\right) + \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{I}, \quad \forall j \in O_1 \tag{18}$$

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{E} - 1 - K\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}\right) \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{E} \leq K\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}\right) + \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{E} - 1, \quad \forall j \in O_1 \tag{19}$$

$$\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{E} + \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{I} \leq K-1+K\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E}, \quad \forall j \in O_1 \tag{20}$$

Constraints (16)-(20) formulate the number of trailers when a suborder $j \in O_1$ is served similarly as Constraints (11)-(15). If $\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E} = 1$, Constraints (16) and (17) guarantee that the number of OFTs remains. Constraint (18) ensures that the number of IFTs remains. Constraint (19) decreases the number of ETs by one. Constraint (20) is relaxed.

If $\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{E} = 0$, Constraint (16) and the domain of y_{jk}^{O} restrict that $\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} = 0$. Constraints (17)-(19) are relaxed. Constraint (20) works similarly as Constraint (15).

F. Trailer Number Formulation Considering I_2

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{O} - K\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T} \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} \leq K\left(1-\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T}\right), \forall j \in I_2 \tag{21}$$

$$\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{O} - \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{O} \leq 0, \quad \forall j \in I_2 \tag{22}$$

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{I} - K\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T} \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{I} \leq (K-1)\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T} + \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{I}, \quad \forall j \in I_2 \tag{23}$$

$$\sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{E} + 1 - (K+1)\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T} \leq \sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{E} \leq K\sum_{i\in\overrightarrow{A}^{-1}(j)} x_{ij}^{T} + \sum_{i\in\overrightarrow{A}^{-1}(j)} y_{ij}^{E} + 1, \quad \forall j \in I_2 \tag{24}$$

$$\sum_{k\in\overleftarrow{A}^{-1}(j)} y_{jk}^{E} \geq 1, \quad \forall j \in I_2 \tag{25}$$

Constraints (21)-(25) formulate the number of trailers when a suborder $j \in I_2$ is served. If the capacity limit of a tractor is not reached, it can go directly to serve suborder j. Otherwise, the tractor must return to the depot to drop some trailers. Specifically, if $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 0$, Constraints (21) and (22) keep the OFTs unchanged, similarly as Constraints (11) and (12). Constraint (23) keeps the IFTs unchanged. Constraint (24) increases the ETs by one. Constraint (25) always holds because the tractor picks up one ET at suborder $j \in I_2$.

If $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 1$, Constraints (21) and (22) work similarly as Constraints (11) and (12). For Constraint (23), the left side is relaxed, and the right side becomes $K - 1 + \sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^I$, and is also relaxed because $\sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^I$ will not exceed $K-1$ considering Constraint (25). Constraint (24) is relaxed.

G. Trailer Number Formulation Considering O_2

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^O + 1 - (K+1) \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T \le \sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^O \le K\left(1 - \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T\right) + 1, \quad \forall j \in O_2 \tag{26}$$

$$1 \le \sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^O \le \sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^O + 1, \forall j \in O_2 \tag{27}$$

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^I - K \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T \le \sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^I \le (K-1) \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T + \sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^I, \quad \forall j \in O_2 \tag{28}$$

$$\sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^E - K \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T \le \sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^E \le (K-1) \sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T + \sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^E, \quad \forall j \in O_2 \tag{29}$$

Constraints (26)-(29) formulate the number of trailers when a suborder $j \in O_2$ is served. If $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 0$, Constraints (26) and (27) force $\sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^O$ to be $\sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^O + 1$ (i.e., the number of OFTs increases by one). If $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 1$, Constraints (26) and (27) force $\sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^O$ to be one. That is, no OFTs are pulled at the depot, and one OFT is picked up at suborder j. Constraint (28) formulates the number of IFTs similarly as Constraint (23). For Constraint (29), if $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 0$, the number of ETs remains. If $\sum_{i\in\overrightarrow{A^{-1}}(j)} x_{ij}^T = 1$, the left side is relaxed; the right side becomes $K - 1 + \sum_{i\in\overrightarrow{A^{-1}}(j)} y_{ij}^E$, and is also relaxed because $\sum_{k\in\overleftarrow{A^{-1}}(j)} y_{jk}^E$ will not exceed $K-1$ when considering the left side of Constraint (27).

H. Transfer Time

$$t_{ij} = \left(1 - x_{ij}^I\right)\left(\tau_{i0} + \tau_{0j}\right) + x_{ij}^I \tau_{ij}, \quad \forall (i,j) \in A^{TRF}, \quad \forall j \in I_1 \tag{30}$$

$$t_{ij} = \left(1 - x_{ij}^E\right)\left(\tau_{i0} + \tau_{0j}\right) + x_{ij}^E \tau_{ij}, \quad \forall (i,j) \in A^{TRF}, \quad \forall j \in O_1 \tag{31}$$

$$t_{ij} = x_{ij}^T\left(\tau_{i0} + \tau_{0j}\right) + \left(1 - x_{ij}^T\right) \tau_{ij}, \quad \forall (i,j) \in A^{TRF}, \quad \forall j \in I_2 \cup O_2 \tag{32}$$

Constraints (30)-(32) determine the transfer time of arcs. If a tractor carries the trailer required by a suborder $j \in I_1 \cup O_1$, i.e., $x_{ij}^I = 1$ or $x_{ij}^E = 1$, we have $t_{ij} = \tau_{ij}$; otherwise, we have $t_{ij} = \tau_{i0} + \tau_{0j}$ according to Constraints (30) and (31). If a tractor carries K trailers before serving a suborder $j \in I_2 \cup O_2$ (i.e., $x_{ij}^T = 1$), $t_{ij} = \tau_{i0} + \tau_{0j}$; otherwise $t_{ij} = \tau_{ij}$ according to Constraint (32).

V. Valid Inequalities

This section presents three families of valid inequalities in order to compact Models SB and LC and to improve their solving efficiency. See Section VII.B.1 for their efficiencies.

A. Strengthened Sub-Tour Elimination Inequality

Models SB and LC eliminate sub-tours by Constraint (4) based on the fact that the decision variable s_i monotonically increases along the routes. However, the strengthened sub-tour elimination (SBE) inequality $x_{ij} + x_{ji} \le 1, \forall (i,j) \in A^{TRF}$, can be directly applied to eliminate such sub-tours that contain two suborder nodes. A similar inequality has also been used by, e.g., Ref [49].

B. Strengthened Deadlock Elimination Inequality

Deadlock in the MTMTDPD problem refers to the circular dependency of service starting times. For example, if $x_{\delta(i)j} = 1$ and $x_{\delta(j)i} = 1$ for nodes $i \in I_1 \cup O_1$ and $j \in I_1 \cup O_1 \backslash \{i\}$ hold, the value for $s_{\delta(i)}$, s_j, $s_{\delta(j)}$ and s_i can never be determined successfully. Deadlocks also exist among suborders of more than two orders in more than two routes. Such solutions that contain deadlocks are infeasible. Models SB and LC eliminate deadlocks similarly as they eliminate sub-tours. Similarly as Inequality SBE, the strengthened deadlock elimination (SDE) inequality $x_{\delta(i)j} + x_{\delta(j)i} \le 1, \forall i \in I_1 \cup O_1, \forall j \in I_1 \cup O_1 \backslash \{i\}$ can directly eliminate the deadlocks appearing in four nodes.

C. Working Time Inequality

The working time inequality (say Inequality WT) is formulated based on the lower bound of total working time. We have the following proposition:

Proposition 1: In the MTMTDPD problem, $\sum_{(i,j)\in A} x_{ij}\tau_{ij}$ is a lower bound of the total working time.

See Appendix B for the proof of Proposition 1. Let e_i^{RETURN} and e_i^{START} denote the time when a tractor finally returns to and initially departs from the depot, respectively. Similarly as in Ref [14], objective (1) can be linearized as

$$\min \rho_1 \sum_{j\in N\backslash\{0\}} x_{0j} + \rho_2 \left(\sum_{i\in N\backslash\{0\}} e_i^{RETURN} - \sum_{i\in N\backslash\{0\}} e_i^{START} \right) \quad (33)$$

$$\text{s.t. } e_i^{RETURN} \geq -H\left(1-x_{i0}\right) + s_i + \tau_{i0}, \quad \forall i \in N\backslash\{0\} \quad (34)$$

$$e_i^{START} \leq H x_{0i}, \forall i \in N\backslash\{0\} \quad (35)$$

$$e_i^{START} \leq H\left(1-x_{0i}\right) + s_i - \tau_{0i}, \quad \forall i \in N\backslash\{0\} \quad (36)$$

$$0 \leq e_i^{RETURN} \leq H, 0 \leq e_i^{START} \leq H, \quad \forall i \in N\backslash\{0\} \quad (37)$$

After replacing objective function (1) with objective (33) and Constraints (34)-(37), we introduce Inequality WT as

$$\sum_{i\in N\backslash\{0\}} e_i^{RETURN} - \sum_{i\in N\backslash\{0\}} e_i^{START} \geq \sum_{(i,j)\in A} x_{ij}\tau_{ij} \quad (38)$$

VI. An Adaptive Large Neighborhood Search Algorithm

Models SB and LC and even the models with valid inequalities can only solve small- or medium-sized instances (see Section VII.B.2). Therefore, we develop an adaptive large neighborhood search (ALNS) algorithm in this section. The ALNS was firstly introduced by Ref [50]. Since its introduction, ALNS has been successfully applied to solve a large number of combinatorial optimization problems including multi-trip VRP [51], VRP with time windows [52], and TTRPs [53].

ALNS is a combination of the large neighborhood search framework [54] and the ruin-and-recreate principle [55]. The key idea is searching for a better solution at each iteration by destroying a part of the current solution and reconstructing it in a different way. Generally, one destruction operator and one reconstruction operator are chosen from a number of candidate operators according to their weights at each iteration. The weights of operators are adjusted dynamically according to their performance. That is the adaptive nature of the ALNS.

A. The ALNS Framework

The proposed ALNS algorithm starts with an initial solution (see Section VI.B) which is considered as the current and best-so-far solution. An iterative procedure is applied to the current solution to obtain a better solution. At each iteration, a destruction operator and a reconstruction operator (see Sections VI.C and VI.D, respectively) are selected based on an adaptive mechanism to generate a neighbor solution. A local search encompassing three operators (see Section VI.E) is used to further improve the neighbor solution, in which process a backtracking adaptive threshold accepting mechanism decides the acceptance of the solution [56, 57]. The iterative procedure continues until a maximum number of iterations are reached. The feasibility checking and objective calculating are presented in Section VI.F.

The adaptive mechanism is similar to that of [58] except for the *reward* rules. Reference [58] reward the operators by adding σ_1, σ_2 or σ_3 to their scores if they provide a new best solution, a solution that is better than the current one, or an accepted non-improving solution, respectively. We reward an operator additionally by adding σ_4 to its scores if it provides a feasible but unacceptable solution to fully use the exploring abilities.

Let $g(\cdot)$ denote the total working time of a solution. A solution hereafter always refers to a feasible one unless explicitly stated. Multiple depot nodes are merged into one if they appear continuously in a solution. Given parameters N^{ITER}, N^{SEG}, I^{NO-MAX}, and ω, the step-by-step algorithm can be formulated as follows:

Step 1: Generate an initial solution S. Let $S^{CUR} = S$, $S^B = S$, $I^{ITER} = 0$, $I^{SEG} = 0$, $T^{MAX} = g\left(S^{CUR}\right)$, $T^H = T^{MAX}$, $T^H = \omega T^{MAX}$, and $I^{NO} = 0$. Initialize equal weights for all the destruction and reconstruction operators.

Step 2: $I^{ITER} = I^{ITER} + 1$. $I^{SEG} = I^{SEG} + 1$. Let $S^{ITER} = S^{CUR}$. Select a destruction operator and a reconstruction operator based on the roulette wheel, and successively apply them to S^{ITER}. If S^{ITER} is successfully reconstructed as a solution, say S^{NEI}, go to Step 3; otherwise, go to Step 10.

Step 3: Let loc be the first operator of the local search. If $g\left(S^{NEI}\right) \geq g\left(S^B\right)$, go to Step 4; otherwise, let $S^{CUR} = S^{NEI}$, $S^B = S^{NEI}$, and go to Step5.

Step 4: If $g\left(S^{NEI}\right) - g\left(S^{CUR}\right) \geq T^H$, $I^{NO} = I^{NO} + 1$, and go to Step 7; otherwise, let $S^{CUR} = S^{NEI}$, and go to Step 5.

Step 5: Apply operator loc to S^{CUR} to generate a solution, say S^{LOC}. If $g\left(S^{LOC}\right) \geq g\left(S^B\right)$ go to Step 6; otherwise, let $S^{CUR} = S^{LOC}$, $S^B = S^{LOC}$, $I^{NO} = 0$, and go to Step 9.

Step 6: If $g\left(S^{LOC}\right) - g\left(S^{CUR}\right) < T^H$, let $S^{CUR} = S^{LOC}$, $T^H = T^H - \Delta T^H$, and go to Step 7; otherwise, $I^{NO} = I^{NO} + 1$.

Step 7: If $T^H \leq 0$, go to Step 8; otherwise, go to Step 9.

Step 8: $T^H = cT^{MAX}$, where c is a random number evenly distributed in the range of [0, 1]. If $I^{NO} \geq I^{NO-MAX}$, let $I^{NO} = 0$ and $S^{CUR} = S^B$.

Step 9: If loc is the last operator of the local search, go to Step 10; otherwise, let it be the next one, and return to Step 5.

Step 10: If $I^{SEG} < N^{SEG}$, go to Step 11; otherwise, let $I^{SEG} = 0$, and update the weights for all the destruction and reconstruction operators.

Step 11: If $I^{ITER} < N^{ITER}$, return to Step 2; otherwise, output S^B and terminate.

B. Initial Solution

A trip is in the form of $(0, v_1, v_2, \cdots, v_n, 0)$, $v_i \in N\backslash\{0\}$, $i = 1, 2, \cdots, n$. A solution is composed of a set of routes each of which is the trip(s) performed by a tractor. The routes in the initial solution are formed in two sequential operations as trip construction and route construction. In the trip construction, we resort to the two-stage merging method of Ref [14] adapted from the Clarke-and-Wright algorithm [59]. Such two trips that can generate a solution with the smallest total working time are merged. Taking trips T_1 and T_2 as an example, merging them means deleting the last depot node of T_1 and the first depot node of T_2, and appending T_2 to T_1. In the route construction, two operators as *route combination* and *route insertion* are successively used in a circular process to reduce the number of routes. The circular process continues until no routes can be reduced.

The route combination appends one route to another directly, and always combines two routes that can generate a solution with the smallest total working time.

Definition 1: Given a node i to be inserted into a set of routes, a *candidate position* $P(i)$ is defined as the position k satisfying any of the following conditions:

(i) Inserting i into k violates the capacity constraints. However, inserting a depot node into the trip based on the insertion of i at an optimum position (i.e., with the least increased total traveling time) can fix the capacity constraints.

(ii) Inserting i into k directly generates a solution.

Definition 2: Given a node i to be inserted into a set of routes, the *best candidate position* $P^B(i)$ is defined as a candidate position that can generate a solution with the shortest total working time.

The operator of route insertion removes node $i \in N\backslash\{0\}$ from a route and inserts it into $P^B(i)$. Given a route of a solution, if all the node $i \in N\backslash\{0\}$ is removed, the route is removed. The operator is executed once on each route. The specific process is as follows.

Step 1: Given a solution S, let $S^{REM} = S$. Let r_i be the first route of S^{REM}.

Step 2: Let v_i be the first non-depot node of r_i.

Step 3: Let $S^* = S^{REM}$, remove v_i from S^*. If $P^B(v_i)$ exists in S^*, insert v_i into $P^B(v_i)$, let $S^{REM} = S^*$, and go to Step 4; otherwise, go to Step 5.

Step 4: If there still exist non-depot nodes in r_i, return to Step 2; otherwise, remove r_i from S^{REM}, let $S = S^{REM}$, and go to Step 6.

Step 5: If v_i is the last non-depot node of r_i, let $S^{REM} = S$, and go to Step 6; otherwise, let it be the next non-depot node, and return to Step 3.

Step 6: If r_i is not the last route of S^{REM}, let it be the next one, and return to Step 2; otherwise, output S and the process terminates.

C. Destruction Operators

This subsection presents two destruction operators: *relatedness-based removal* and *worst removal*. Each of them removes the Phase I suborder node and its corresponding Phase II suborder node simultaneously, and removes a total number of q nodes.

1) Relatedness-Based Removal

Definition 3: The *relatedness* between two nodes $i \in I_1 \cup O_1$ and $j \in I_1 \cup O_1\backslash\{i\}$ in a solution is defined as

$$\varpi(i, j) = \psi^T \frac{\tau_{ij} + \tau_{\delta(i)\delta(j)}}{D} + \psi^R \left(\varphi_{ij} + \varphi_{\delta(i)\delta(j)}\right) + \psi^S \frac{\left|s_i - s_j\right| + \left|s_{\delta(i)} - s_{\delta(j)}\right|}{H}, \quad i \in I_1 \cup O_1, j \in I_1 \cup O_1\backslash\{i\} \tag{39}$$

where ψ^T, ψ^R and ψ^S are given weights, $D = \max\left\{\tau_{ij} \,|\, (i, j) \in A\right\}$, and φ_{ij} is equal to one if nodes i and j are in the same route, or zero otherwise. Note that s_i is the service starting time of node i, as defined in the two models. The term weighted by ψ^T describes the distance between two customer locations, the term weighted by ψ^R indicates the node location distances in the routes, and the term weighted by ψ^S measures the temporal connectedness.

The step-by-step procedure of this operator is as follows:

Step 1: Given a solution S and an empty set Q, randomly choose a node $i \in I_1 \cup O_1$ from S, move i and $\delta(i)$ from S to Q.

Step 2: Randomly choose a node $j \in I_1 \cup O_1$ from Q. Select such a node $i \in I_1 \cup O_1$ that minimizes $|\varpi(i, j) - \mu|S||$ from S, and move i and $\delta(i)$ from S to Q, where μ is a random number between zero and one, and $|\cdot|$ is the number of elements in the set.

Step 3: If $|Q| < q$, return to Step 2; otherwise, output S and the operator terminates.

2) Worst Removal

Definition 4: Given a node $i \in I_1 \cup O_1$ of a solution S, the *removal cost* on the total working time is defined as $\Delta(i, S) = g(S) - g_{-(i)}(S)$, in which $g_{-(i)}(S)$ is the total working time of S without i and $\delta(i)$.

The worst removal removes nodes based on the removal cost. The specific removing procedures are as follows:

Step 1: Given a solution S, assign a weight $g(S) + \Delta(i, S)$ to each node i and $\delta(i)$, $i \in I_1 \cup O_1$. Let Q be an empty set.

Step 2: Select node i using the roulette wheel mechanism from S, $i \in I_1 \cup O_1$. Move node i and $\delta(i)$ from S to Q.

Step 3: If $|Q| < q$, return to Step 2; otherwise, output S and the operator terminates.

D. Reconstruction Operators

1) Deep Greedy Insertion Operator: The deep greedy insertion operator inserts the removed node into the position generating the smallest total working time. It consists of two stages. First, it fixes those trips that are infeasible in capacity constraints using the removed nodes. Second, it iteratively inserts the best node of the remaining removed nodes at its best candidate position. A Phase II suborder node will be selected for insertion only if its corresponding Phase I suborder node has been successfully inserted. The operation terminates if no trips can be restored, or some removed nodes can never be inserted feasibly. The specific procedures are as follows:

Step 1: Given q removed nodes, put the Phase I and Phase II suborder nodes into empty sets Q and Q^{II}, respectively. Let v_i be the first node of Q. Let S be the destroyed solution and f be its first capacity-infeasible trip.

Step 2: Check all possible inserting positions of f to find a $P(v_i)$ generating a solution with the smallest traveling time. If such $P(v_i)$ does not exist, go to Step 3; otherwise, remove v_i from Q, insert it into $P(v_i)$, and go to Step 4.

Step 3: If v_i is not the last node of Q, let it be the next one, and return to Step 2; otherwise, the operator terminates.

Step 4: If $v_i \in I_1 \cup O_1$, move $\delta(v_i)$ from Q^{II} to Q. If f is the last capacity-infeasible trip of S, go to Step 5; otherwise, let it be the next one, let v_i be the first node of Q, and return to Step 2.

Step 5: If $Q \neq \emptyset$, where $\emptyset$ denotes an empty set, go to Step 6; otherwise, output S, and the operator terminates.

Step 6: Attempt to find $P^B(i)$ for each node $i \in Q$. If such a position exists, remove the best node v from Q, insert it at $P^B(v)$, and go to Step 7; otherwise, terminate.

Step 7: If $v \in I_1 \cup O_1$, move $\delta(v)$ from Q^{II} to Q. Return to Step 5.

2) Regret Insertion Operator: The regret insertion operator is similar to the deep greedy insertion except for the node selecting manner in the second stage. Let $reg_i = g^S - g^B$ be the *regret* value of node i, where g^S and g^B are the total working time of inserting i at its second best and first best candidate positions, respectively. The node with the highest regret value is selected for insertion.

E. Local Search Operators

There are three sequentially executed operators: *trip piece insertion*, *node exchange,* and *node insertion*. All the operators operate in both inter- and intra-route ways.

Trip Piece Insertion: A trip piece is a sequence with $k \in \{1, 2, \ldots, K\}$ nodes. Given a trip including n non-depot nodes, $n + 1 - K$ trip pieces can be identified if $n \geq K$. Each trip piece includes K successive nodes. If $n < K$, there is only one trip piece with n nodes. For example, supposing $K = 2$, trip $(0, v_1, v_2, v_3, 0)$ includes two trip pieces as (v_1, v_2) and (v_2, v_3), while trip $(0, v_4, 0)$ includes only one trip piece (v_4). The operator removes the best trip piece, and inserts it at its best position to generate a solution with the smallest total working time.

Node Exchange: This operator exchanges two best nodes to generate a solution with the smallest total working time.

Node Insertion: Similarly as the trip piece insertion, this operator removes the best node, and inserts it at its best position.

F. Feasibility and Optimum of Solutions

A solution is feasible only if its routes satisfy the capacity constraints, precedence constraints, no-deadlock constraints and planning horizon constraints. We resort to the three-vector procedure of [14] to check the capacity constraints. Considering the precedence constraints, the Phase II suborder node cannot be seated before its corresponding Phase I suborder node if both nodes exist in the same route. The other situations are checked in the calculation process of total working time.

Calculating the total working time is difficult due to the interdependence among routes. Regarding this issue, Zhang *et al.* proposed a two-stage heuristic method [47]. First, the service starting time of nodes is temporarily given according to the idea of topological sorting. The service starting time is then adjusted in the reverse order of topological sorting. Zhang *et al.* improved the method by adding a third stage named *forward adjustment* [14]. However, both methods cannot guarantee the optimality of service starting times for solutions. Here, we tackle this problem by delivering the route information to a solver by constructing a mathematical model for the routes.

Before presenting this approach, we take a deeper look at the proposed ALNS algorithm. In the algorithm, a set of routes excluding node $i \in I_2 \cup O_2$ or node $j \in I_1 \cup O_1$ and its corresponding node $\delta(j)$ are allowed. However, cases in which the Phase II suborder node appears while its corresponding Phase I suborder node is absent from the routes are forbidden.

Now, given a solution, its routes can be defined on a graph $G^* = (N^*, A^*)$, where $N^* = \{0\} \cup N^{II} \cup N^I \cup \tilde{N^I}$ is the set of nodes, $A^* = A^{IND} \cup A^D \cup \{(i, j) | i \in N^*, j \in N^* \backslash \{i\}, (i, j) \notin A^{IND} \cup A^D, (i, j) \notin A^{RVS}\}$ is the set of arcs. In N^*, N^{II} is the set of Phase II suborder nodes existing in the routes, N^I is the set of Phase I suborder nodes corresponding to the nodes of N^{II}, and $\tilde{N^I}$ is the set of such Phase I suborder nodes that appear in the routes but the corresponding Phase II suborder nodes are absent from the routes. In A^*, A^{IND} is the set of arc (i, j) in which nodes i and j are connected via depot node in the routes, and A^D is the set of arc $(i, j) \notin A^{IND}$ in which nodes i and j are directly connected after deleting the depot node between trips for the routes.

With decision variables x_{ij}, t_{ij}, s_i (see Section IV), e_i^{RETURN} and e_i^{START} (see Section V.C), the total working time of a solution can be derived from the following model.

$$\min \sum_{i \in N^* \backslash \{0\}} e_i^{RETURN} - \sum_{i \in N^* \backslash \{0\}} e_i^{START} \tag{40}$$

$$s.t.\ e_i^{RETURN} \geq -H(1 - x_{i0}) + s_i + \tau_{i0}, \quad \forall i \in N^* \backslash \{0\} \tag{41}$$

$$e_i^{START} \leq H x_{0i}, \quad \forall i \in N^* \backslash \{0\} \tag{42}$$

$$e_i^{START} \leq H(1 - x_{0i}) + s_i - \tau_{0i}, \quad \forall i \in N^* \backslash \{0\} \tag{43}$$

$$0 \leq e_i^{RETURN} \leq H, 0 \leq e_i^{START} \leq H, \quad \forall i \in N^* \backslash \{0\} \tag{44}$$

$$x_{ij} = 1, \quad \forall (i, j) \in A^D \cup A^{IND} \tag{45}$$

$$x_{ij} = 0, \quad \forall (i, j) \in A^* \backslash A^D \backslash A^{IND} \tag{46}$$

$$t_{ij} = \tau_{ij}, \quad \forall (i, j) \in A^D, i \neq 0, j \neq 0 \tag{47}$$

$$t_{ij} = \tau_{i0} + \tau_{0j}, \quad \forall (i, j) \in A^{IND} \tag{48}$$

$$s_i + t_{ij} - s_j \leq M(1 - x_{ij}), \forall (i, j) \in A^*, i \neq 0, j \neq 0 \tag{49}$$

$$s_i + p_i - s_{\delta(i)} \leq 0, \quad \forall i \in N^I \tag{50}$$

$$\tau_{0i} \leq s_i \leq H - \tau_{i0}, \quad \forall i \in N^* \backslash \{0\} \tag{51}$$

$$t_{ij} \geq 0, \quad \forall (i, j) \in A^*, i \neq 0, j \neq 0 \tag{52}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A^* \tag{53}$$

In the model, Objective (40) minimizes the total working time. Constraints (41)-(44) correspond to Constraints (34)-(37), respectively. Constraints (45) and (46) present the visiting sequences. Constraints (47) and (48) present the transfer times. Constraints (49) and (50) correspond to Constraints (4) and (5), respectively. Constraints (51)-(53) define the decision variables.

VII. Computational Experiments

This section conducts extensive computational experiments to validate the models and ALNS algorithm. All models were solved with CPLEX (version 12.6.1, 32 bits). All procedures

TABLE V
RESULTS OF MODEL SB WITH DIFFERENT VALID INEQUALITIES

Instance	Obj.	CPU (s)				
		None	SBE	SDE	WT	All
R1	20.76	11.79	**7.54**[*]	**7.21**	**1.15**	**1.08**
R2	21.57	256.55	**194.90**	**218.47**	**49.35**	**45.97**
R3	22.53	293.65	**268.97**	300.57	**48.84**	**45.51**
R4	21.69	5532.13	6881.35	7773.48	**1831.75**	**1703.63**

[*] Boldface indicates shorter computation time than that without inequalities

TABLE VI
RESULTS OF MODEL LC WITH DIFFERENT VALID INEQUALITIES

Instance	Obj.	CPU (s)				
		None	SBE	SDE	WT	All
R1	20.76	9.13	**7.89**[a]	9.48	**4.71**	**2.09**
R2	21.57	502.19	**382.02**	**497.25**	**46.66**	**45.27**
R3	22.53	1004.95	**564.68**	**477.32**	**42.29**	**40.56**
R4	21.69	NA[b]	NA	20298.30	4266.96	1862.79

[a] Boldface indicates shorter computation time than that without inequalities

[b] No solution is obtained due to limitation of memory

TABLE VII
THE NUMBER (AMONG 25) OF INSTANCES WITH IMPROVED BOUNDS BY MODEL SB WITH INEQUALITIES, COMPARED TO THAT WITHOUT INEQUALITIES

Inequality SBE		Inequality SDE		Inequality WT		All inequalities	
UB	LB	UB	LB	UB	LB	UB	LB
12	6	14	14	22	25	18	25

were coded in C++. All experiments were carried out on a personal computer with a 3.40 GHz CPU and 8 GB RAM.

Section VII.A describes the test instances and experimental settings. Section VII.B validates the inequalities and Models SB and LC. Section VII.C assesses the performances of the ALNS algorithm. Finally, Section VII.D explores the applicable scenario of the multi-trip multi-trailer D&P transportation.

A. Instances and Parameter Setting

Random instances R1-R25 with the number of orders ranging from 4 to 60 and clustered instances C17-C25 with the number of orders ranging from 20 to 60 are taken from the single-trailer instances of Ref [13] (see https://github.com/HiWangH/datasets.git). The planning horizon H is set as 16 hours based on the time limit of work shifts [60]. For all experiments, we set $K = 2$ because one tractor pulling two trailers is the most common scenario in real world applications. Moreover, we set $\rho_1 = 10$ and $\rho_2 = 1$ according to Ref [13]. For each instance, the ALNS algorithm runs for five times independently and the best solution is presented unless explicitly stated.

Parameter N^{ITER} is set as 200 for Instances R1-R20 and C17-C20, and 100 for Instances R21-R25 and C21-C25. Parameters N^{SEG}, σ_1, σ_2, σ_3 and σ_4 are respectively set as 5, 100, 20, 10, and 5. Parameter q ranges from $\lfloor 0.1 \, |N \backslash \{0\}| \rfloor$ to $\lceil \xi \, |N \backslash \{0\}| \rceil$, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are respectively the downward and upward integer functions, and ξ is a given parameter. Parameters ψ^T, ψ^R, ψ^S, ω, I^{NO-MAX}, γ and ξ are sequentially tuned based on Instances R17, R20, C17 and C20 using the popular method of [50], where γ is a reaction factor controlling weight adjustments of the adaptive mechanism. The final values are: $\psi^T = 0.8, \psi^R = 1.2, \psi^S = 1.6$, $\omega = 0.04$, $I^{NO-MAX} = 4$, $\gamma = 0.3$, and $\xi = 0.2$.

B. Performance of the Exact Approaches

1) Effectiveness of Valid Inequalities: We first validate the inequalities based on small-sized Instances R1-R4. Tables V and VI summarize the solutions of Models SB and LC under different scenarios. Column *None* reports the computation time of the model without any inequalities. The next three columns list the computation time of the model with one of the inequalities. Column *All* reports the computation time with all the inequalities.

We can observe from Table V that the objective values for the model with and without inequalities are the same. The inequalities, Inequality WT in particular, can sharply shorten the computation time in most cases. When all the inequalities are added, the computation times for R1-R4 are respectively decreased from 11.79, 256.55, 293.65 and 5532.13 seconds to 1.08, 45.97, 45.51 and 1703.63 seconds. The average decrease is 81.66%. The valid inequalities show similar effects on Model LC (see Table VI). Note that Model LC without inequalities cannot solve Instance R4. However, when inequalities (except Inequality SBE) are added, the instance becomes solvable. All these results validate the effectiveness of the inequalities.

We carried out experiments on 25 medium- and large-sized instances (R5-R23, C17-C21, and C23) to investigate the performance of the inequalities in improving the bounds at the root, which can further validate the effectiveness to some extents. We set the maximum number of nodes solved by the branch-and-cut algorithm of CPLEX as zero. The computational results show that the inequalities, except Inequality SBE, can decrease the upper bounds for more than half of the instances. Inequality WT is outstanding among the inequalities. It decreases the upper bounds for instances except Instances R9, R16 and R20. The average decrease is 18.32%. Inequality WT also shows its strength in improving the lower bounds. It lifts the lower bounds from negative values to positive values. The average improvement is 124.84%. Generally, if all the inequalities are added, the upper and lower bounds are improved by 12.74% and 125.42% on average, respectively.

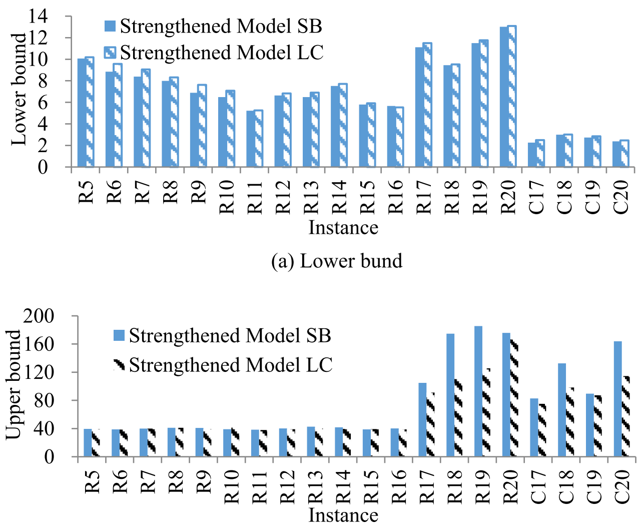
For Model LC, although solutions for some instances cannot be found due to the limit of solvable nodes, each inequality can still improve the upper and lower bounds for about half of the instances. When all the inequalities are added, the upper bounds are a little worse than those without any inequalities, but the lower bounds are improved by up to 141.20% on average. Tables VII and VIII summarize the total number of improved upper bounds (UB) and lower bounds (LB), obtained by Models SB and LC under different scenarios.

2) Comparison Between Models SB and LC: We compare the performance of Models SB and LC with all valid inequalities (i.e., strengthened Models SB and LC) based on 20 instances (R5-R20 and C17-C20). The *depth-first search* strategy is used in CPLEX. The time limit for solving each

TABLE VIII

THE NUMBER (AMONG 25) OF INSTANCES WITH IMPROVED BOUNDS BY MODEL LC WITH INEQUALITIES, COMPARED TO THAT WITHOUT INEQUALITIES

Inequality SBE		Inequality SDE		Inequality WT		All inequalities	
UB	LB	UB	LB	UB	LB	UB	LB
15	11	13	15	13	15	9	14



(a) Lower bund

(b) Upper bound

Fig. 4. Performance of Models SB and LC with all valid inequalities.

instance is set as one hour. Fig. 4 presents the results on lower bounds and upper bounds. The results indicate that the strengthened Model LC provides better lower bounds for 19 instances and better upper bounds for 17 instances, compared to the strengthened Model SB. The strengthened Model LC yields better solutions for 17 instances. The maximum upper bound difference is 36.79% reached at R18. The average upper bound difference is 8.51%. However, realistic-sized instances are still hard to solve, which necessitates the design of the ALNS algorithm.

C. Performance of the ALNS Algorithm

1) Comparison to the Strengthened Model LC: The ALNS algorithm is compared with the strengthened Model LC based on 23 instances (R1-R20 and C17-C20). The comparative results indicate that the ALNS algorithm can provide better solutions in shorter computation time (as shown in Table IX). The ALNS algorithm provides optimum solutions for the four small-sized instances (R1-R4). The ALNS provides better solutions for 18 out of the other 20 instances. The ALNS algorithm provides the same number of tractors as the strengthened Model LC, but obtains a shorter total working time for 10 instances (R7-R16) among them. The objective difference ranges from 0.96% to 5.30%. The ALNS algorithm provides smaller value for both the number of employed tractors and the total working time for the other eight instances. The maximum objective improvement is up to 49.79% which is reached at C20. The average objective improvement over all instances is 12.76%. The savings on the computation time, employed tractors and working time are 55601.46 seconds, 15 and 158.97 hours, respectively.

TABLE IX

PERFORMANCE OF THE ALNS ALGORITHM COMPARED TO THE STRENGTHENED MODEL LC WHEN SOLVING 24 INSTANCES

Total saving			Objective decrease (%)		
CPU(s)	Tractors	Working time (h)	Min.	Max.	Average
55601.46	15	158.97	0.96	49.79	12.76

TABLE X

PERFORMANCE OF THE ALNS ALGORITHM WITHOUT CERTAIN COMPONENTS

Excluded component	Average deterioration on obj. (%)	Average deterioration on CPU (%)
Relatedness-based removal	0.45	9.42
Worst removal	0.72	5.08
Deep greedy insertion	0.76	13.07
Regret insertion	0.27	5.25
Trip piece insertion	0.97	-8.68
Node exchange	1.27	-3.45
Node insertion	1.09	-17.75

(a) Variation of average employed tractor numbers

(b) Variation of average total working time

Fig. 5. Performance of different transportation modes.

2) Contribution of ALNS Components: This subsection demonstrates contributions of the ALNS components based on Instances R1-R20 and C17-C20. We ran the ALNS algorithm excluding one operator while retaining the others to solve each instance. The deteriorated solution is compared to that of the intact ALNS algorithm. Overall, the worst removal is the most effective destruction operator, the deep greedy insertion is the most effective reconstruction operator, and the node exchange is the most effective local search operator (as illustrated in Table X). The absence of destruction or reconstruction operators incurs an increase of computation time. Their utilization is reasonable considering the caused deterioration on objectives. The absence of local search operators saves computation times between 3.45% and 17.75%.

However, the objective deterioration of 0.97%-1.09% is significant. Therefore, it is reasonable to embed them in the algorithm.

D. Applicable Scenario of Multi-Trip Multi-Trailer D&P

This section explores which (one-tractor-two-trailer or one-tractor-one-trailer) mode is more suitable for application when customer distribution is changed, based on Instance R14. The original geographical locations of customers are randomly distributed in a square Euclidean plane with a side length of 200 kilometers. The depot is located at the center point $(100, 100)$. We now choose the customers according to a probability of 0.2, 0.4, 0.6, 0.8, and 1, and regenerate their geographical locations in an area with both axes ranging from 300 to 400 kilometers randomly. For each probability, 10 instances are generated and solved using the ALNS algorithm. The average number of employed tractors and total working time are presented in Fig. 5(a) and Fig. 5(b), respectively. We can find that the two indexes for both the two- and single-trailer modes increase with the choice probability. However, the difference between them becomes obvious for each index. The increasing trend of the multi-trailer mode is flatter. As a result, the multi-trailer mode is more suitable if customers are clustered in the area far away from the depot.

$$\sum_{k\in \overleftarrow{A}^{-1}(j)} y_{jk}^{O} \begin{cases} = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{O}, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} > 0; \text{or } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} > 0; \text{or } j\in I_2, \\ & \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) < K \\ = 0, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} = 0; \text{or } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} = 0; \text{or } j\in I_2, \\ & \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) \geq K \\ = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{O}+1, & \text{if } j\in O_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) < K \\ = 1, & \text{if } j\in O_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) \geq K \end{cases} \tag{54}$$

$$\sum_{k\in \overleftarrow{A}^{-1}(j)} y_{jk}^{I} \begin{cases} = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I}-1, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} > 0 \\ \leq K-1, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} = 0; \text{or } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} = 0; \text{or } j\in I_2\cup O_2, \\ & \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) \geq K \\ + \sum_{k\in \overleftarrow{A}^{-1}(j)} y_{jk}^{E} \leq K-1, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} = 0; \text{or } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} = 0 \\ = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I}, & \text{if } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} > 0; \text{or } j\in I_2\cup O_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) < K \end{cases} \tag{55}$$

$$\sum_{k\in \overleftarrow{A}^{-1}(j)} y_{jk}^{E} \begin{cases} = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E}, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} > 0; \text{or } j\in O_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) < K \\ \leq K-1, & \text{if } j\in I_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{I} = 0; \text{or } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} = 0; \text{or } j\in O_2, \\ & \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) \geq K \\ = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E}-1, & \text{if } j\in O_1, \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E} > 0 \\ = \sum_{i\in \overrightarrow{A}^{-1}(j)} y_{ij}^{E}+1, & \text{if } j\in I_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) < K \\ \geq 1, & \text{if } j\in I_2, \sum_{i\in \overrightarrow{A}^{-1}(j)} \left(y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O}\right) \geq K \end{cases} \tag{56}$$

$$t_{ij} = \begin{cases} \tau_{ij}\ \forall\ (i,j)\in A^{TRF}, & \text{if } j\in I_1, y_{ij}^{I} > 0; \text{or } j\in O_1, y_{ij}^{E} > 0; \text{or } j\in I_2\cup O_2, y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O} < K \\ \tau_{i0}+\tau_{0j}\ \forall\ (i,j)\in A^{TRF}, & \text{if } j\in I_1, y_{ij}^{I} = 0; \text{or } j\in O_1, y_{ij}^{E} = 0; \text{or } j\in I_2\cup O_2, y_{ij}^{I}+y_{ij}^{E}+y_{ij}^{O} \geq K \end{cases} \tag{57}$$

The reason is that the traveling times between customers and the depot are long. Benefiting from large transportation capacity, the multi-trailer mode requires a small number of tractors to accomplish more orders, and avoids frequent shuttling back and forth between customers and the depot.

VIII. Conclusion and Further Directions

This paper investigated a MTMTDPD problem. Two mixed-integer programming models are constructed to formulate the problem. Three families of valid inequalities are presented to strengthen the models. An ALNS algorithm is developed to solve realistic-sized instances of the problem, in which a mathematical decoding method is proposed to obtain the optimal service starting time. Experimental results indicate that the inequalities are valid and the mathematical model based on IF-THEN constraints outperforms the state-based model. The ALNS algorithm further outperforms the strengthened mathematical model. The multi-trailer mode is more suitable for transportation services if the customers are clustered far away from the depot, when compared to the single-trailer mode.

The limitations of this research are as follows. First, this research considers a popular static scenario, which limits its application in real-life scenarios to a certain degree. Second, we focus on the handling of a multi-trip feature assuming that the travel time includes no uncertainty and that the customers have no time window constraints. Third, the tractors and trailers, as well as the containers, are assumed to be homogeneous.

This research might be followed in several ways. For example, some features, including limited resources, heterogeneous tractors/trailers/containers, time window constraints, and multiple depots, might be introduced into the problem. Consideration of greenhouse gas emissions could be another vein of research that might yield solutions for making transportation more environmentally friendly. Finally, introducing dynamic elements such as the appearance of new orders, cancellations of existing orders, weather changes, and breakdowns of vehicles, into the MTMTDPD problem are also interesting topics for further research. However, the problem with the aforementioned features is almost definitely to be more difficult to solve and requires even stronger solution strategies.

Appendix A
Model LC

Constraints (11), (12), (16), (17), (21), (22), (26) and (27) in Model SB can be reformulated as the following IF-THEN form, (54), shown at the bottom of the previous page.

Similarly, Constraints (13), (15), (18), (20), (23) and (28) can be reformulated as, (55), shown at the bottom of the previous page.

Constraints (14), (19), (24), (25) and (29) can be reformulated as, (56), shown at the bottom of the previous page.

Constraints (30)-(32) can be reformulated as, (57), shown at the bottom of the previous page.

Now we get Model LC with objective function (1), Constraints (2)-(7) and (54)-(57). See https://github.com/HiWangH/Model-Code.git for the code.

Appendix B
Proof of Proposition 1

Let w_i be the waiting time of the tractor serving node $i \in N\backslash\{0\}$. For the total working time of tractors, we have $\sum_{i\in N\backslash\{0\}} x_{i0}(s_i+\tau_{i0}) - \sum_{i\in N\backslash\{0\}} x_{0i}(s_i-\tau_{0i}) = \sum_{i\in N\backslash\{0\}} w_i + \sum_{(i,j)\in A^{TRF}} x_{ij}t_{ij} + \sum_{i\in N\backslash\{0\}} x_{i0}\tau_{i0} + \sum_{i\in N\backslash\{0\}} x_{0i}\tau_{0i}$. Moreover, considering the triangular inequality and Constraints (30)-(32), we get $t_{ij} \geq \tau_{ij}$, $(i,j) \in A^{TRF}$. Therefore, $\sum_{(i,j)\in A^{TRF}} x_{ij}t_{ij} \geq \sum_{(i,j)\in A^{TRF}} x_{ij}\tau_{ij}$, and

$$\begin{aligned}&\sum_{i\in N\backslash\{0\}} w_i + \sum_{(i,j)\in A^{TRF}} x_{ij}t_{ij} + \sum_{i\in N\backslash\{0\}} x_{i0}\tau_{i0} + \sum_{i\in N\backslash\{0\}} x_{0i}\tau_{0i}\\ &\geq \sum_{i\in N\backslash\{0\}} w_i + \sum_{(i,j)\in A^{TRF}} x_{ij}\tau_{ij} + \sum_{i\in N\backslash\{0\}} x_{i0}\tau_{i0} + \sum_{i\in N\backslash\{0\}} x_{0i}\tau_{0i}\\ &\geq \sum_{(i,j)\in A^{TRF}} x_{ij}\tau_{ij} + \sum_{i\in N\backslash\{0\}} x_{i0}\tau_{i0} + \sum_{i\in N\backslash\{0\}} x_{0i}\tau_{0i} \end{aligned} \quad (58)$$

According to the definition of set A, we have $\sum_{(i,j)\in A} x_{ij}\tau_{ij} = \sum_{(i,j)\in A^{TRF}} x_{ij}\tau_{ij} + \sum_{i\in N\backslash\{0\}} x_{i0}\tau_{i0} + \sum_{i\in N\backslash\{0\}} x_{0i}\tau_{0i}$. As a result,

$$\begin{aligned}&\sum_{i\in N\backslash\{0\}} x_{i0}(s_i+\tau_{i0}) - \sum_{i\in N\backslash\{0\}} x_{0i}(s_i-\tau_{0i})\\ &\qquad \geq \sum_{(i,j)\in A} x_{ij}\tau_{ij}\end{aligned} \quad (59)$$

which is exactly Proposition 1. □

Acknowledgment

The authors would like to thank the Editor-in-Chief, the Handling Editor, and the anonymous reviewers for their valuable and constructive comments to improve this paper.

References

[1] A. Caris and G. K. Janssens, "A local search heuristic for the pre and end-haulage of intermodal container terminals," *Comput. Oper. Res.*, vol. 36, no. 10, pp. 2763–2772, Oct. 2009.
[2] K. Braekers, A. Caris, and G. K. Janssens, "Integrated planning of loaded and empty container movements," *OR Spectr.*, vol. 35, no. 2, pp. 457–478, Mar. 2013.
[3] S. Shiri and N. Huynh, "Optimization of drayage operations with time-window constraints," *Int. J. Prod. Econ.*, vol. 176, no. 1, pp. 7–20, Jun. 2016.
[4] M. Moghaddam, R. H. Pearce, H. Mokhtar, and C. G. Prato, "A generalised model for container drayage operations with heterogeneous fleet, multi-container sizes and two modes of operation," *Transp. Res. E, Logistics Transp. Rev.*, vol. 139, p. 101973, Jul. 2020.
[5] H. Cui, S. Chen, R. Chen, and Q. Meng, "A two-stage hybrid heuristic solution for the container drayage problem with trailer reposition," *Eur. J. Oper. Res.*, vol. 299, no. 2, pp. 468–482, Jun. 2022.
[6] M. H. Akyâz and C.-Y. Lee, "Service type assignment and container routing with transit time constraints and empty container repositioning for liner shipping service networks," *Transp. Res. B, Methodol.*, vol. 88, pp. 46–71, Jun. 2016.
[7] J. You, L. Miao, C. Zhang, and Z. Xue, "A generic model for the local container drayage problem using the emerging truck platooning operation mode," *Transp. Res. B, Methodol.*, vol. 133, pp. 181–209, Mar. 2020.
[8] D. A. Neira, M. M. Aguayo, R. De la Fuente, and M. A. Klapp, "New compact integer programming formulations for the multi-trip vehicle routing problem with time windows," *Comput. Ind. Eng.*, vol. 144, Jun. 2020, Art. no. 106399.

[9] M. Lai, T. G. Crainic, M. D. Francesco, and P. Zuddas, "An heuristic search for the routing of heterogeneous trucks with single and double container loads," *Transp. Res. E, Logistics Transp. Rev.*, vol. 56, pp. 108–118, Sep. 2013.
[10] M. Bruglieri, S. Mancini, R. Peruzzini, and O. Pisacane, "The multi-period multi-trip container drayage problem with release and due dates," *Comput. Oper. Res.*, vol. 125, Oct. 2021, Art. no. 105102.
[11] R. Chen, S. Chen, H. Cui, and Q. Meng, "The container drayage problem for heterogeneous trucks with multiple loads: A revisit," *Transp. Res. E, Logistics Transp. Rev.*, vol. 147, Jul. 2021, Art. no. 102241.
[12] A. Ghezelsoflu, M. Di Francesco, A. Frangioni, and P. Zuddas, "A multiperiod drayage problem with customer-dependent service periods," *Comput. Oper. Res.*, vol. 129, May 2021, Art. no. 105189.
[13] Z. Xue, C. Zhang, W.-H. Lin, L. Miao, and P. Yang, "A Tabu search heuristic for the local container drayage problem under a new operation mode," *Transp. Res. E, Logistics Transp. Rev.*, vol. 62, pp. 136–150, Feb. 2014.
[14] R. Zhang, D. Wang, and J. Wang, "Multi-trailer drop-and-pull container drayage problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5708–5720, Sep. 2021.
[15] P. Nagl, "Longer combination vehicles (LCV) for Asia and the pacific region: Some economic implications," United Nations, New York, NY, USA, UNESCAP Work. Papers WP/07/02, 2007. [Online]. Available: https://repository.unescap.org/handle/20.500.12870/1198
[16] J. Woodrooffe and L. Ash, *Economic Efficiency of Long Combination Transport Vehicles in Alberta*. Edmonton, AB, Canada: Alberta Transportation, 2001. [Online]. Available: https://www.osti.gov/etdeweb/biblio/20468835
[17] F. Meisel and H. Kopfer, "Synchronized routing of active and passive means of transport," *OR Spectr.*, vol. 36, no. 2, pp. 297–322, Mar. 2014.
[18] Y. Song, J. Zhang, Z. Liang, and C. Ye, "An exact algorithm for the container drayage problem under a separation mode," *Transp. Res. E, Logistics Transp. Rev.*, vol. 106, pp. 231–254, Oct. 2017.
[19] C. Tilk, N. Bianchessi, M. Drexl, S. Irnich, and F. Meisel, "Branch-and-price-and-cut for the active-passive vehicle-routing problem," *Transp. Sci.*, vol. 52, no. 2, pp. 300–319, Mar. 2018.
[20] V. N. Coelho, A. Grasas, H. Ramalhinho, I. M. Coelho, M. J. F. Souza, and R. C. Cruz, "An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints," *Eur. J. Oper. Res.*, vol. 250, no. 2, pp. 367–376, Apr. 2016.
[21] M. W. P. Savelsbergh, "The vehicle routing problem with time windows: Minimizing route duration," *ORSA J. Comput.*, vol. 4, no. 2, pp. 146–154, May 1992.
[22] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems," *J. Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, Jun. 2008.
[23] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Oper. Res.*, vol. 54, no. 3, pp. 573–586, 2006.
[24] J. Tang, Y. Yu, and J. Li, "An exact algorithm for the multi-trip vehicle routing and scheduling problem of pickup and delivery of customers to the airport," *Transp. Res. E, Logistics Transp. Rev.*, vol. 73, pp. 114–132, Jan. 2015.
[25] M. Liu, Z. Luo, and A. Lim, "A branch-and-cut algorithm for a realistic dial-a-ride problem," *Transp. Res. B, Methodol.*, vol. 81, pp. 267–288, Aug. 2015.
[26] M. A. Masmoudi, M. Hosny, K. Braekers, and A. Dammak, "Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem," *Transp. Res. E, Logistics Transp. Rev.*, vol. 96, pp. 60–80, Jun. 2016.
[27] Z. Zhang, M. Liu, and A. Lim, "A memetic algorithm for the patient transportation problem," *Omega*, vol. 54, pp. 60–71, Jul. 2015.
[28] S. Liu, S. Qin, and R. Zhang, "A branch-and-price algorithm for the multi-trip multi-repairman problem with time windows," *Transp. Res. E, Logistics Transp. Rev.*, vol. 116, pp. 25–41, Aug. 2018.
[29] V. François, Y. Arda, Y. Crama, and G. Laporte, "Large neighborhood search for multi-trip vehicle routing," *Eur. J. Oper. Res.*, vol. 255, no. 2, pp. 422–441, Dec. 2016.
[30] A. Moreno, D. Alem, D. Ferreira, and A. Clark, "An effective two-stage stochastic multi-trip location-transportation model with social concerns in relief supply chains," *Eur. J. Oper. Res.*, vol. 269, no. 3, pp. 1050–1071, Sep. 2018.
[31] D. Wang, J. Zhu, X. Wei, T. C. E. Cheng, Y. Yin, and Y. Wang, "Integrated production and multiple trips vehicle routing with time windows and uncertain travel times," *Comput. Oper. Res.*, vol. 103, pp. 1–12, Mar. 2019.
[32] D. Cattaruzza, N. Absi, and D. Feillet, "The multi-trip vehicle routing problem with time Windows and release dates," *Transp. Sci.*, vol. 50, no. 2, pp. 676–693, May 2016.
[33] L. Zhen, C. Ma, K. Wang, L. Xiao, and W. Zhang, "Multi-depot multi-trip vehicle routing problem with time windows and release dates," *Transp. Res. E, Logistics Transp. Rev.*, vol. 135, Apr. 2020, Art. no. 101866.
[34] K. Sethanan and T. Jamrus, "Hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry," *Comput. Ind. Eng.*, vol. 146, Jun. 2020, Art. no. 106571.
[35] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud, "Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 249, no. 2, pp. 551–559, Mar. 2016.
[36] N. Huang, J. Li, W. Zhu, and H. Qin, "The multi-trip vehicle routing problem with time windows and unloading queue at depot," *Transp. Res. E, Logistics Transp. Rev.*, vol. 152, Aug. 2021, Art. no. 102370.
[37] B. Pan, Z. Zhang, and A. Lim, "Multi-trip time-dependent vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 291, no. 1, pp. 218–231, May 2021.
[38] D. Cattaruzza, N. Absi, and D. Feillet, "Vehicle routing problems with multiple trips," *Ann. Operations Res.*, vol. 271, no. 1, pp. 127–159, Dec. 2018.
[39] N. Markovi, H. Drobnjak, and P. Schonfeld, "Dispatching trucks for drayage operations," *Transp. Res. E, Logistics Transp. Rev.*, vol. 70, pp. 99–111, Jun. 2014.
[40] R. Zhang, J.-C. Lu, and D. Wang, "Container drayage problem with flexible orders and its near real-time solution strategies," *Transp. Res. E, Logistics Transp. Rev.*, vol. 61, pp. 235–251, Jan. 2014.
[41] R. Zhang, W. Y. Yun, and H. Kopfer, "Multi-size container transportation by truck: Modeling and optimization," *Flexible Services Manuf. J.*, vol. 27, nos. 2–3, pp. 403–430, Sep. 2015.
[42] I.-M. Chao, "A Tabu search method for the truck and trailer routing problem," *Comput. Oper. Res.*, vol. 29, no. 1, pp. 33–51, Jan. 2002.
[43] M. Drexl, "Applications of the vehicle routing problem with trailers and transshipments," *Eur. J. Oper. Res.*, vol. 227, no. 2, pp. 275–283, Jun. 2013.
[44] M. Schiffer, M. Schneider, G. Walther, and G. Laporte, "Vehicle routing and location routing with intermediate stops: A review," *Transp. Sci.*, vol. 53, no. 2, pp. 319–343, 2019.
[45] Z. Xue, H. Lin, and J. You, "Local container drayage problem with truck platooning mode," *Transp. Res. E, Logistics Transp. Rev.*, vol. 147, Mar. 2021, Art. no. 102211.
[46] X. Yang, H. A. Daham, and A. Salhi, "Combined strip and discharge delivery of containers in heterogeneous fleets with time windows," *Comput. Oper. Res.*, vol. 127, Aug. 2021, Art. no. 105141.
[47] R. Zhang, H. Zhao, and I. Moon, "Range-based truck-state transition modeling method for foldable container drayage services," *Transp. Res. E, Logistics Transp. Rev.*, vol. 118, pp. 225–239, Mar. 2018.
[48] O. Bräysy, W. Dullaert, G. Hasle, D. Mester, and M. Gendreau, "An effective multirestart deterministic annealing Metaheuristic for the fleet size and mix vehicle-routing problem with time windows," *Transp. Sci.*, vol. 42, no. 3, pp. 371–386, Aug. 2008.
[49] Ç. Koç, T. Bektaé, O. Jabali, and G. Laporte, "The fleet size and mix location-routing problem with time windows: Formulations and a heuristic algorithm," *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 33–51, Jan. 2016.
[50] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transp. Sci.*, vol. 40, no. 4, pp. 455–472, 2006.
[51] P. Grangier, M. Gendreau, F. Lehuédé, and L.-M. Rousseau, "An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization," *Eur. J. Oper. Res.*, vol. 254, no. 1, pp. 80–91, Oct. 2016.
[52] H. Ben Ticha, N. Absi, D. Feillet, and A. Quilliot, "Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 104, pp. 113–126, Mar. 2019.
[53] S. N. Parragh and J.-F. Cordeau, "Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows," *Comput. Oper. Res.*, vol. 83, pp. 28–44, Jul. 2017.
[54] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Proc. Int. Conf. Princ. Pract. Constraint Program.*, vol. 1520. Berlin, Germany: Springer, 1998, pp. 417–431.

[55] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, "Record breaking optimization results using the ruin and recreate principle," *J. Comput. Phys.*, vol. 159, no. 2, pp. 139–171, Apr. 2000.
[56] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, "A threshold accepting Metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *Eur. J. Oper. Res.*, vol. 152, no. 1, pp. 148–158, Jan. 2004.
[57] K. Braekers, A. Caris, and G. K. Janssens, "Bi-objective optimization of drayage operations in the service area of intermodal terminals," *Transp. Res. E, Logistics Transp. Rev.*, vol. 65, pp. 50–69, 2014.
[58] G. M. Ribeiro and G. Laporte, "An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem," *Comput. Oper. Res.*, vol. 39, no. 3, pp. 728–735, Mar. 2012.
[59] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, 1964.
[60] A. M. Campbell and M. Savelsbergh, "Efficient insertion heuristics for vehicle routing and scheduling problems," *Transp. Sci.*, vol. 38, no. 3, pp. 369–378, Aug. 2004.

Decheng Wang received the B.S. degree in automation from the Lanzhou University of Technology, Lanzhou, China, in 2013, and the M.S. and Ph.D. degrees in systems engineering from the College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2016 and 2021, respectively.

He is currently a Lecturer with the College of Information Science and Engineering, Guilin University of Technology, Guilin, China. His major research interests include modeling and optimization with applications in logistics and transportation and theory and method of planning and scheduling.

Ilkyeong Moon received the B.S. and M.S. degrees in industrial engineering from Seoul National University, South Korea, and the Ph.D. degree in operations research from Columbia University. He is currently a Professor of industrial engineering at Seoul National University. He has published over 130 articles in international journals. His research interests include supply chain management, logistics, and inventory management. He was the President of the Korean Institute of Industrial Engineers (KIIE) in which he had served from 2019 to 2020. He is a fellow of the Asia–Pacific Industrial Engineering and Management Society. He is the Vice Chair of WG 5.7 of IFIP. He was the Former Editor-in-Chief of *Journal of the Korean Institute of Industrial Engineers* which is a flagship journal of KIIE. He serves as an Associate Editor for several journals, including *European Journal of IE* and *Flexible Services and Manufacturing*.

Ruiyou Zhang (Member, IEEE) received the B.Sc. degree in automation and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China, in 2002 and 2007, respectively.

From 2007 to 2008, he was a Post-Doctoral Fellow with the Department of Industrial Engineering, Pusan National University, Busan, South Korea. From 2010 to 2011, he was a Visiting Scholar with the School of Industrial and Systems Engineering, Georgia Institute of Technology, GA, USA. He is currently a Long-Termed Professor with the College of Information Science and Engineering, Northeastern University. He has published more than 40 refereed technical articles on international/domestic journals. His major research interests include modeling and optimization, intelligent manufacturing, logistics and transportation, and service science and engineering.