

A BI-OBJECTIVE MODEL FOR ROBUST LOCAL CONTAINER DRAYAGE PROBLEM

JINTAO YOU^{1,2}, CANRONG ZHANG^{1,2}, ZHAOJIE XUE^{3,*}, LIXIN MIAO^{1,2} AND BIN YE⁴

Abstract. Local Container Drayage Problem (LCDP) refers to the optimization of the process of planning and scheduling container trucks between a terminal and customers to offer door-to-door service to customers in a local area. The time required for (un)packing containers at customers' sites are often relatively long and uncertain due to the current (un)packing work level and unexpected deviations in operational situations, which has a significant influence on the planning and scheduling of the container transportation process. This paper examines the LCDP with Separable tractors and trailers, and additionally with consideration of (un)packing time Uncertainties (LCDPSU). A proactive strategy is employed to tackle the uncertainty by proposing a “model robust” bi-objective optimization model to balance the tradeoff between operational cost, which includes traveling cost and tractor deployment setup cost, and robustness, which is represented as an exponential expression of the time buffer between two stages of each individual task. The deterministic version of our problem is proved to be NP hard, and an Ant Colony Optimization (ACO) scheme is therefore proposed to search for feasible solutions in which the Zoutendijk feasible direction algorithm is embedded in order to tackle the nonlinearity brought in by the robustness of the model. Numerical experiments are conducted to validate the efficiencies and effectivenesses of the proposed models and methods, and managerial implications are drawn from the numerical results.

Mathematics Subject Classification. 90B06.

Received January 7, 2019. Accepted June 12, 2019.

1. INTRODUCTION

Container transportation is safe, convenient, and low cost, and it can carry out intermodal transportation through waterways, railways, and highways to realize door-to-door transportation services between customers all over the world. In the system of intermodal container transportation, vessels and rails are the main modes of

Keywords. Local Container Drayage Problem, (un)packing time uncertainties, time buffer insertion, Ant Colony Optimization, Zoutendijk feasible direction algorithm.

¹ Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China.

² Logistics Engineering and Simulation Laboratory, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, PR China.

³ Department of Transportation Engineering, College of Civil Engineering, Shenzhen University, Shenzhen 518060, PR China.

⁴ School of Environmental Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, PR China.

*Corresponding author: zjxue@szu.edu.cn

long-distance trunk transportation, and container trucks play an important role in branch transportation. The container transportation by trucks between a terminal and customers is usually referred to as container drayage operation, and it provides container pickup and delivery services for shippers and consignees.

In container drayage operation, a container truck consists of a tractor and a trailer carrying a container; the tractor and the trailer are not operationally separable in the traditional stay-with mode. In the stay-with mode, when a truck hauls an empty/loaded container from the terminal to the location of a customer, the driver and the entire container truck will wait at the customers' site for packing/unpacking. Only after the packing/unpacking work is completed, can the driver and tractor continue to haul the container back to the terminal. The (un)packing works always last several hours or even longer which significantly restrict the utilization of container trucks and manpower. In the tractor-trailer separable mode, the tractor can be separated from the trailer when one required container is transported to a customer, and the tractor continues to serve other customers while the trailer and the container remain at the customer location for packing/unpacking. Through this mode of operation, the planner can effectively use the idle waiting times of drivers and tractors, thereby improving their utilization rate. The new tractor-trailer separable operation mode also give rise to more possibilities for inserting time buffers between tasks, since the driver and tractor are able to leave the customer while (un)packing work is being conducted, hence the idle time can be utilized efficiently.

The durations of (un)packing times required at the customers' sites are critical to container trucks' schedules under the tractor-trailer separable mode and deserve close attention. In practice, one frequently used operation strategy is that customers make commitments on the length of (un)packing times to the carrier before the planning and scheduling process, then the carrier uses these commitment values and deterministic optimization modeling tools to generate baseline schedules. However, there are often deviations between the realized (un)packing times and the committed values, which make the baseline schedules become infeasible, and operators may face severe losses for rearrangements. Theoretically, there are two kinds of strategies for dealing with uncertainties: reactive and proactive scheduling. Reactive strategies are those that respond to uncertainties only after they are realized, while proactive strategies are designed to anticipate uncertain situations before complete information is obtained. In this paper, we focus on proactive strategies for LCDP under the tractor-trailer separable operation mode, which involves developing baseline schedules that incorporate anticipations of uncertainties. Robust optimization and stochastic optimization can be adopted to deal with scenario uncertainties and random distribution uncertainties respectively. However, since it is difficult to obtain specific uncertain information, in this paper, we have adopted an idea from the model robust optimization technique by introducing a time buffer insertion technique to deal with the uncertainties, instead of using stochastic optimization techniques.

This paper studies the LCDPSU by applying a proactive strategy, where we adopt the time buffer insertion method to improve the robustness of tractor schedule plans under uncertainties without knowing any specific uncertainty information in advance. A "model robust" bi-objective optimization model is proposed, and a two-stage heuristic based on the Ant Colony Optimization(ACO) scheme is developed to solve the model, in which Zoutendijk feasible direction algorithm is incorporated to resolve the nonlinearity that emerges in the model. Numerical experiments are conducted to verify the effectiveness of the schedules obtained using the proposed algorithms and to obtain some management insights.

The rest of the paper is organized as follows: a literature review is given in Section 2; the LCDPSU is described in detail in Section 3 along with a mathematical model; the solution algorithm is presented in Section 4, followed by the numerical experiments and analysis in Section 5; and the conclusions are drawn in the last section.

2. LITERATURE REVIEW

2.1. The Local Container Drayage Problem

LCDP has been studied extensively in recent years, but most of those studies have concentrated on the deterministic models under various container drayage operation modes and the techniques for finding their solutions. Wang and Regan [22] studied the local truckload pickup and delivery problem and modeled it as

a multiple Traveling Salesman Problem with Time Window(m-TSPTW), and they applied a time-window partition method to solve this problem. Jula *et al.* [11] studied the Local Container Drayage Problem (LCDP), which was defined as planning the movements of containers around a metropolitan (local) area by a truck company. In their paper, each customer has a service requirement to deliver a container from an origin to a destination (a O-D pair), and the O-D pairs are merged into one node and the entire problem is modeled as an asymmetric multi-travelling salesman problem with time windows (am-TSPTW). Coslovich *et al.* [3] proposed a decomposition method to solve the fleet management problem for an Italian container company, in which they combined the drivers' labor costs, trucks' deployment costs, container repositioning costs, and routing costs. Tjokroamidjojo *et al.* [20] studied the benefits and costs of sharing load information in advance of the pickup and delivery of containers. Imai *et al.* [10] studied the problem that emerges in picking and delivering full container load from/to an intermodal terminal. They achieved time and cost savings by linking the pickup and delivery tasks on restricted conditions, and a Lagrangian relaxation based sub-gradient heuristic was adopted to find the near optimal solution. Zhang *et al.* [26] considered a container transportation problem with resource constraints where empty containers are treated as transportation resources, and the problem is based on one-terminal and one-depot local area. Zhang *et al.* [25] further extended the truck scheduling problem for container transportation with limited empty container resources into a multiple depots and multiple terminals problem. Xue *et al.* [23] studied the LCDP under a special operation mode where the tractor and trailer can be separated, in which they specifically considered the (un)packing time needed at each customer location, and a tabu heuristic was applied to solve this problem. Xue *et al.* [24] also tried to solve the LCDP under tractor-trailer separable mode model exactly by adopting the combinatorial Benders' cut algorithm, but they did not present any results for large sized instances. Shiri and Huynh [16] addressed a problem in which the intermodal terminal requires trucks to have an appointment. The problem was formulated as an m-TSPTW, and was solved using an algorithm based on Reactive Tabu Search(RTS). Song *et al.* [17] considered the LCDP under a separation mode where some empty containers should return to the depot for maintenance, and they presented an arc-flow formulation and applied a branch-and-price-and-cut algorithm to solve this problem exactly. Braekers *et al.* [2] proposed a bi-objective model for LCDP and three algorithms to solve it. However, the bi-objective model they studied is a deterministic LCDP. Ghezelsoufi *et al.* [5] proposed a set-covering model with single and double container loads, where all the possible routes are enumerated, and the model can be solved efficiently by a commercial solver.

2.2. Robust optimization methods and applications

In recent years, robust optimization has played an important role in dealing with uncertain optimization problems. In traditional robust optimization techniques, uncertain parameters are described by discrete scenarios or continuous intervals, and the goal of robust optimization is to seek a solution that has acceptable performance under all uncertain situations. The solution philosophy behind most of the robust optimization problems is worst-case optimization.

Han *et al.* [8] studied a proactive approach to deal with the berth and quay crane scheduling problem with uncertain arrival and handling times, and a simulation based Genetic Algorithm (GA) was applied to generate robust berth and QC schedule proactively. Zhen and Chang [27] applied a bi-objective model to study the proactive strategy for robust berth allocation and scheduling, where the arriving time and operation time are uncertain. In their paper, they designed a heuristic to generate baseline schedules, and adopted an exponential expression to describe the robustness. Wang and Meng [21] studied the liner ship route scheduling problem where the wait time and container handling time are uncertain. The formulated mixed-integer nonlinear stochastic programming model was solved by using a combination of a sample average approximation method, linearization techniques, and a decomposition scheme. As for the VRP, Sungur *et al.* [19] studied a robust optimization approach for a capacitated VRP under demand uncertainty, whose objective is to find routes that minimize transportation costs while satisfying demands in a given bounded uncertainty set. They proved that in certain formulation and specific uncertainty sets, the solution procedure of the robust model will not be harder than the deterministic model. Gounaris *et al.* [6] also studied capacitated VRP under demand uncertainty. Their

goal was to determine a minimum cost delivery plan that is feasible for all anticipated demand realizations, and they discussed the conditions when a robust VRP can be solved as a deterministic problem. Han *et al.* [7] studied the VRP with uncertain travel times and the time exceeding the planning horizon was penalized in the objective. Instead of using a traditional stochastic programming approach in which precise knowledge of the probabilities are required, they described the uncertain travel times as time intervals and replaced the point estimates of travel times by range estimates. Lee *et al.* [13] studied a robust VRP with deadlines and vehicle capacity limitations, where the demand of each customer and the travel times between customers are uncertain. They proposed a Dantzig-Wolfe decomposition approach in which the uncertain data are dealt in the column generation subproblem and solved using a dynamic programming based algorithm. Liu *et al.* [14] proposed a bi-objective model for yard allocation scheduling for outbound containers; their solution technique is a heuristic, and the nonlinearity is transformed by discretization.

To the best of our knowledge, most of the researches on LCDP were conducted for deterministic problems, whose optimization goals include setup cost, operation cost, and penalty cost. Meanwhile, robust optimization techniques have been widely used in VRP studies, and robust optimization has the advantage that the specific distribution of uncertain data is not necessary. Compared with the above literature, the main contributions of this paper are as follows:

- (1) A buffer-based robust model is proposed to address the LCDP under tractor-trailer separable mode, where the commitment values of (un)packing are given to planners by customers before the planning process; however, the realistic (un)packing times can be different due to disruption in the operational process.
- (2) Similar to previous literature, in order to measure the robustness more accurately, the time buffers are treated as exponential expressions. Unlike the previous papers such as Liu *et al.* [14], and Zhen and Chang [27], which adopted a discretized way to handle the nonlinearly exponential term, this paper adopts a continuous way by using Zoutendijk feasible direction algorithm.
- (3) Based on the concept “model robust” proposed by Mulvey *et al.* [15], a bi-objective model is proposed to balance the operational cost and robustness, and due to the NP-hardness, the model is solved by using an ACO scheme embedded with Zoutendijk feasible direction algorithm. Numerical experiments are conducted to validate the model and solution techniques, and management suggestions are drawn from the experimental results.

3. BI-OBJECTIVE MODEL FOR THE LCDP WITH (UN)PACKING TIME UNCERTAINTY

In this section we present the details of how the previously studied deterministic LCDP model is modified into a bi-objective model to confront (un)packing time uncertain scenarios.

3.1. Cost-minimization objective

As shown in Section 2, the objective of deterministic LCDP models studied in previous papers is mostly to minimize the operational cost, and the main tradeoff is between the number of tractors deployed and the total travel distance. Since the problem we study is based on LCDP with tractor-trailer separable mode and empty container sharing strategy, in this paper, we refer to the LCDP model proposed by Xue *et al.* [23], and the (un)packing times are considered uncertain. In order to maintain the integrity of this paper, the critical modeling issues are briefly stated as follows. For more detailed modeling descriptions, we refer the readers to Xue *et al.* [23].

Container drayage operation considers the container transportation between a terminal and a set of customers around the local area of the terminal. The LCDP studied in this paper assumes that there are two types of customers: delivery customers and pickup customers. Delivery customers require cargos delivered from the terminal while pickup customers need an empty container to transport their cargos to the terminal.

We assume that there are a certain number of full-container-load pickup and delivery customers around a local terminal area. These tasks have to be fulfilled within a fixed planning horizon. We mathematically represent the problem as a graph $G = (N, A)$. Node set N and arc set A are described below.

We tactically split each customer node into two task nodes, which have the same location with the original customer node spatially. To be more specified, a pickup customer node is split into an empty pickup task node and a loaded pickup task node. An empty pickup task node requires an empty container. A loaded pickup task node requires a tractor to transport its loaded container back to the terminal. Similarly, a delivery customer node can be split into a loaded delivery task node and an empty delivery task node. A loaded delivery task node requires its loaded container transported from the terminal. An empty delivery task node requires a tractor to tow the emptied container away. It is noted that the two task nodes of the same customer can be served by different tractors. We suppose there are p pickup customers and d delivery customers. The total task number $n = 2(p + d)$.

The definitions of the node sets are:

C : represents the task node set of all the customers; each customer node is split into two task nodes: a stage one task node and a stage two task node

C_1 : the stage one task node set

C_2 : the stage two task node set

N : the entire node set

The relationships between the node sets are as follows:

$$N \begin{cases} 0 & \text{the terminal node} \\ C & \begin{cases} C_1 \begin{cases} P_1 : \{1, 2, \dots, p\} & \text{empty pickup task nodes} \\ D_1 : \{p+1, p+2, \dots, p+d\} & \text{loaded delivery task nodes} \end{cases} \\ C_2 \begin{cases} P_2 : \{1', 2', \dots, p'\} & \text{loaded pickup task nodes} \\ D_2 : \{(p+1)', (p+2)', \dots, (p+d)'\} & \text{empty delivery task nodes.} \end{cases} \end{cases} \end{cases}$$

Here we introduce the notations $\theta(i) = i'$ and $\theta(i') = i$ for $i \in C_1$ to represent the corresponding relationship between C_1 and C_2 .

The arc set is represented by $A = \{(i, j) | i, j \in N, i \neq j\}$. For each arc (i, j) , we use τ_{ij} to denote the direct vehicle travel time from the location of node i to the location of node j , and t_{ij} to denote the real travel time between two consecutive nodes served by the same tractor. Based on tractor-trailer separable mode and empty container sharing strategy, t_{ij} may have different relationships with τ_{ij} as is shown in Table 1:

ρ_1 : the fixed setup cost for a deployed tractor

ρ_2 : the truck/tractor travel cost per unit travel time

\tilde{p}_i : commitment (un)packing times for customer i , $\forall i \in C_1$, which is uncertain before the scenarios are realized

T : the planning horizon

K : a sufficiently large constant

Variables:

X_{ij} : equals 1 if node i and j are served consecutively by the same tractor, $\forall i, j \in N, i \neq j$; 0 otherwise

S_i : the service beginning time at node i , $i \in C$

Cost-minimization model:

$$\text{Min} \quad \rho_1 \cdot \sum_{j \in C} X_{0j} + \rho_2 \cdot \sum_{i \in N} \sum_{j \in N \setminus \{i\}} t_{ij} X_{ij} \quad (3.1)$$

$$\text{s.t.} \quad \sum_{i \in N \setminus \{j\}} X_{ij} = 1, \quad \forall j \in C \quad (3.2)$$

$$\sum_{i \in N \setminus \{j\}} X_{ij} - \sum_{i \in N \setminus \{j\}} X_{ji} = 0, \quad \forall j \in C \quad (3.3)$$

TABLE 1. Calculation of t_{ij} .

$i \setminus j$	0	P_1	D_1	P_2	D_2
0	—	τ_{0j}	τ_{0j}	τ_{0j}	τ_{0j}
P_1	τ_{i0}	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$	τ_{ij}	τ_{ij}
D_1	τ_{i0}	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$	τ_{ij}	τ_{ij}
P_2	τ_{i0}	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$
D_2	τ_{i0}	τ_{ij}	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$	$\tau_{i0} + \tau_{0j}$

$$S_i + t_{ij} \leq S_j + K(1 - X_{ij}), \quad \forall i, j \in C, i \neq j \quad (3.4)$$

$$S_i + \tilde{\mathbf{p}}_i \leq S_{\theta(i)}, \quad \forall i \in C_1 \quad (3.5)$$

$$S_i \geq t_{0i}, \quad \forall i \in C_1 \quad (3.6)$$

$$S_i + t_{i0} \leq T, \quad \forall i \in C_2 \quad (3.7)$$

$$X_{ij} \in \{0, 1\}, \quad \forall i, j \in N, i \neq j. \quad (3.8)$$

Objective function (3.1) minimizes the total cost, including the cost of deployed tractors and total travel cost. Constraints (3.2) and (3.3) are flow constraints. Constraints (3.4) specify the service beginning times of two consecutive task nodes served by the same tractor. Constraints (3.5) are the temporal dependence between the two stages of the same customer, which ensure that the (un)packing time of each customer is satisfied. Constraints (3.6) and (3.7) ensure that all tasks are finished within the planning horizon. Constraints (3.8) define the decision variables as binary ones.

3.2. Robustness-maximization objective

Figures 1 and 2 show a simple example to demonstrate how a robust-oriented schedule can be more tolerant to uncertainties than a cost-oriented schedule. The travel time and commitment (un)packing times are labeled in Figure 1, where D and D' represent the two time points of the delivery customer before and after unpacking, P and P' represent the two time points the pickup customer before and after packing. The planning horizon we consider here is 15. Figure 2 shows the comparison between a cost-oriented schedule and a robust-oriented schedule on a time axis. The cost-oriented schedule is operated in the stay-with mode and the robust-oriented is operated in tractor-trailer separable mode. The comparison shows that the cost-oriented schedule has a shorter total travel time, while the robust-oriented schedule is more tolerant to uncertainties. For example, when the realized unpacking time of the delivery customer is larger than its committed value, and supposing the realized unpacking time is 5, the robust-oriented schedule will still be feasible; however, the cost-oriented schedule will be infeasible under this situation.

As shown in the example in Figures 1 and 2, a baseline schedule plan generated from a traditional cost-minimization model will become infeasible or may cause severe losses under uncertainties. Robust solutions can perform better under such uncertain situations, although with a possible increase in cost.

While creating baseline schedules for tractors, the first issue is how to define and measure the robustness. There are several strategies to measure the robustness in practice, such as the expected objective values, the objective value of the worst case, the length of the slack time (buffer time), and so on. In this paper, we adopt a strategy based on buffer time, and robustness is defined as a weighted sum of buffer times [1], in which the weights are related to the relative punctuality levels of customers to fit practical situations.

The buffer time b_i is defined as the time period between the (un)packing work's finish time (which is $S_i + \tilde{\mathbf{p}}_i, \forall i \in C_1$) for customer node i and the service starting time of the customer's stage two task node (which is $S_{\theta(i)}$), which is expressed as $b_i = S_{\theta(i)} - S_i - \tilde{\mathbf{p}}_i$.

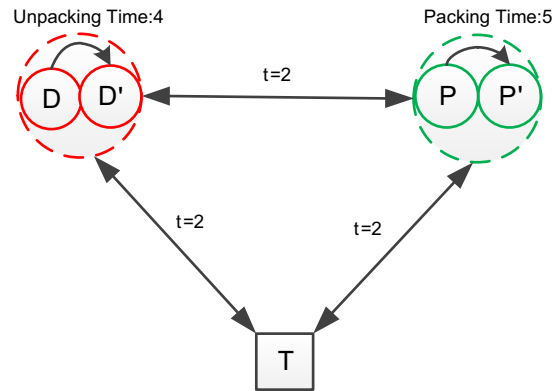


FIGURE 1. An illustrative network of one delivery customer and one pickup customer.

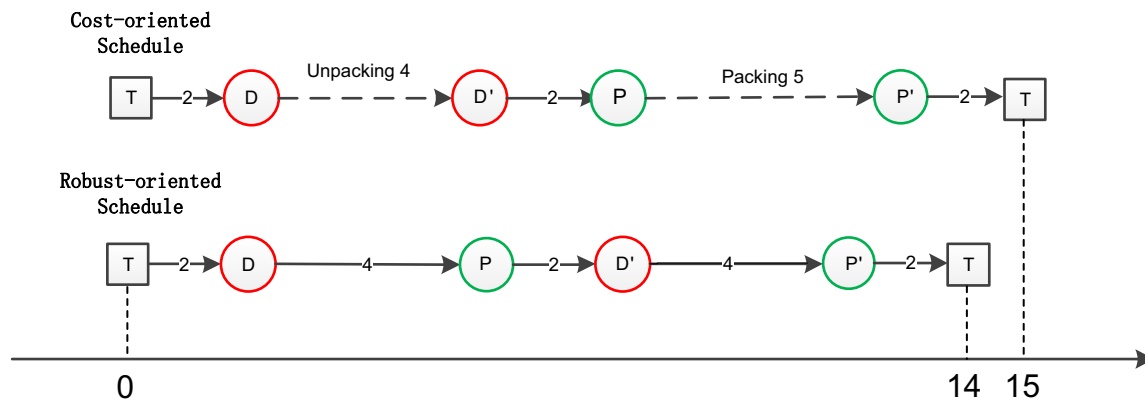


FIGURE 2. Min-cost schedule and max-robustness schedule for the illustrative network.

Intuitively, the time buffer for each customer needs to be arranged as long as possible. However, considering common practice and the studies in related areas, the robustness return per time unit for each customer is diminishing. It should be mentioned that the adoption of exponential decreasing function (rather than other expressions, such as linear expression, logarithmic function and so on) to express the diminishing trend of robustness return is a usual practice in robust project scheduling [9, 12], and [27] has applied this idea in the research on robust berth allocation scheduling.

The robustness factor R_i for each customer i is calculated as $R_i = \int_0^{b_i} e^{-x} dx = 1 - e^{-b_i}$. Based on the above explanation, the robustness of a given schedule is measured by the sum of robustness factor R_i . In the solution construction process, we want to maximize the robustness, and the robustness of a solution s is calculated as follows:

$$\text{Robust maximization: } R(s) = \sum_{i \in C_1} R_i = \sum_{i \in C_1} (w_i \cdot (1 - e^{-b_i})). \quad (3.9)$$

Here, w_i is the parameter that measures the punctuality level of customers based on the fleet planner's human experience.

3.3. Bi-objective optimization: min-cost and max-robustness

The purpose of this study is to obtain a robust schedule with a satisfactory cost. However, the improvement in robustness will cause an increase in transportation cost, and vice versa. In order to balance the tradeoff between the two objectives, a bi-objective optimization model is proposed based on the models described above. This model can help planners make proactive scheduling plans when facing (un)packing time uncertainties based on limited information (only the commitment (un)packing times are known in advance).

$$\text{Objective 1: Min } \rho_1 \cdot \sum_{j \in C} X_{0j} + \rho_2 \cdot \sum_{i \in N} \sum_{j \in N \setminus \{i\}} t_{ij} X_{ij} \quad (3.10)$$

$$\text{Objective 2: Max } \sum_{i \in C_1} (w_i \cdot (1 - e^{-b_i})) \quad (3.11)$$

$$\text{s.t. } (3.2)-(3.8)$$

$$b_i = S_{\theta(i)} - S_i - \tilde{\mathbf{p}}_i, \quad \forall i \in C_1 \quad (3.12)$$

$$b_i \geq 0. \quad (3.13)$$

Constraints (3.12) define the time buffer b_i for each customer i . Through the bi-objective optimization model, we are able to obtain a Pareto front with a given set of commitment values, and the details of calculation for obtaining the Pareto front are presented in Section 5.3.2. After the non-dominated solutions are obtained, their performances are further tested with a set of actual stochastic instances generated by a simulator, and the details of the evaluation strategies are presented in Section 5.

4. SOLUTION APPROACH

We introduce a coefficient λ to combine the above two objectives:

$$\text{Objective: Min } (1 - \lambda) \cdot \left(\rho_1 \cdot \sum_{j \in C} X_{0j} + \rho_2 \cdot \sum_{i \in N} \sum_{j \in N \setminus \{i\}} t_{ij} X_{ij} \right) - \lambda \cdot \sum_{i \in C_1} w_i \cdot (1 - e^{-b_i}). \quad (4.1)$$

In this paper, the coefficient λ is a parameter ranging from 0.1 to 0.9, and varied in steps of 0.1. By changing the value of λ , different sets of solutions can be obtained, where $\lambda = 0.1$ means the objective is cost-oriented, and $\lambda = 0.9$ tends to give a robust-oriented solution. The step in which λ is varied can be adjusted according to the requirement.

Though we have converted two objectives into one, it does not change the NP hard essence of this problem. Because the original deterministic LCDP is an extension of VRP, which is already NP hard, the tractor-trailer separable mode and empty container sharing strategy considered in this paper make the problem even harder. In addition, the nonlinear part of the objective function makes the problem more sophisticated. Therefore, we design a two stage heuristic to solve this problem. Firstly, we adopt the Ant Colony Optimization (ACO) scheme to guide the heuristic search process, and then a Zoutendijk feasible direction algorithm is applied to tackle the nonlinearity. A brief skeleton of the entire algorithm is described as follows. A feasible sequence of the task nodes is generated through ACO, and the value of objective 1 is obtained at the same time. After that, the optimal service time of this sequence is calculated using Zoutendijk feasible direction algorithm; simultaneously, we can obtain the value of objective 2. Then, the combined objective value is saved as heuristic information to guide further search of ACO. This search process continues iteratively until the stopping criteria are met. The following sections provide detailed descriptions of ACO and the Zoutendijk feasible direction algorithm.

4.1. Ant Colony Optimization

4.1.1. Preparation process

Before the heuristic process, some preparation work is done to narrow the search space by inspection [23]. The time windows of each task node can be reduced from $[0, T]$ to the following:

$$[a_i, b_i] = [t_{0i}, T - t_{\theta(i),0} - p_i], \quad \forall i \in C_1 \quad (4.2)$$

$$[a_i, b_i] = [t_{0,\theta(i)} + p_{\theta(i)}, T - t_{i0}], \quad \forall i \in C_2. \quad (4.3)$$

For a given sequence, the lower bounds of the service beginning times are updated as follows.

$$S_0 = 0, \quad (4.4)$$

$$S_i = S_j + t_{ji}, \quad \text{if } j \text{ precedes } i \text{ in the same route, } i \in C_1, j \in N \quad (4.5)$$

$$S_i = \max\{S_j + t_{ji}, S_{\theta(i)} + p_{\theta(i)}\}, \text{ if } j \text{ precedes } i \text{ in the same route, } i \in C_2, j \in N. \quad (4.6)$$

During the heuristic search process, customers' service beginning times are updated dynamically. In addition, they are checked with the time windows to ensure feasibility.

4.1.2. Ant Colony Optimization scheme

ACO is an algorithm in which a set of artificial ants are generated to iteratively construct candidate solutions. The ants' solution construction processes are guided by pheromone and problem-dependent heuristic information, as shown by Dorigo *et al.* [4]. In each construction process, every ant chooses unvisited task nodes based on certain probabilities. The probability of choosing node j after node i is given by

$$p_{ij} = \frac{\xi_{ij}^\alpha \eta_{ij}^\beta}{\sum_{(i,j) \in A} \xi_{ij}^\alpha \eta_{ij}^\beta}, \quad \forall (i, j) \in A, \quad (4.7)$$

where ξ is the pheromone and η is the heuristic information, α and β are the parameters that determine the relative importance of ξ and η . To be specific, the heuristic information factor η is highly problem dependent. When applying ACO to solve the TSP, the heuristic information is conventionally defined as $\eta_{ij} = 1/d_{ij}$, where d_{ij} is the length of arc (i, j) . In this paper, we define the heuristic information as follows:

$$\eta_{ij} = \begin{cases} \frac{1}{\rho_1 + \rho_2 \cdot t_{ij}}, & \text{if } i = 0 \\ \frac{1}{\rho_1 \cdot p_i / T}, & \text{if } i \in C_1 \text{ and } j = \theta(i) \\ \frac{1}{\rho_1 \cdot t_{ij} / T + \rho_2 \cdot t_{ij}}, & \text{otherwise.} \end{cases} \quad (4.8)$$

Due to the random data generation rules, t_{ij} can be very small, which will cause η_{ij} to be a very large number. In order to diversify the solution space, we introduce $\bar{\eta}_{ij}$ as an upper bound of η_{ij} .

To avoid early convergence, γ is introduced as the volatilization rate of the pheromone. In each iteration, only the elite ant is used to update the pheromone:

$$\xi_{ij}(t+1) = (1 - \gamma)\xi_{ij}(t) + \Delta\xi_{ij}, \quad (4.9)$$

where t is the iteration counter, and $\Delta\xi_{ij}$ is calculated in the following way:

$$\Delta\xi_{ij} = \begin{cases} Q/f(s), & \text{solution } S \text{ contains arc}(i, j) \\ 0, & \text{otherwise.} \end{cases} \quad (4.10)$$

Here, $f(s)$ is the objective value of solution s , and Q is the pheromone updating constant. In order to avoid situations in which some ants have much larger pheromone values than the others, $\xi_{ij}(t)$ is restricted within

the bounds ξ_{\min} and ξ_{\max} . According to Stutzle and Hoos [18], $\xi_{\min}(t)$ and $\xi_{\max}(t)$ are dynamically calculated as follows:

$$\lim_{t \rightarrow \infty} \xi_{ij}(t) \leq \frac{1}{\gamma} \frac{Q}{f(s^{\text{opt}})} = \xi_{\max}(t), \quad (4.11)$$

$$\xi_{\min}(t) = \xi_{\max}(t)/2n, \quad (4.12)$$

where s^{opt} is the current global best solution.

To initialize the pheromone, we set $\xi(0)$ as an arbitrary large number. Following the rules of equations (4.6) and (4.8), $\xi(1)$ will be forced to $\xi_{\max}(1)$ [18].

4.1.3. Route-Saving operator

The probabilistic searching essence of ACO can lead the search to slow convergence or a local minimum. Local search operators are often introduced to improve the solution quality. However, in this paper, due to the planning horizon and temporal constraints, common operators such as inter-route, intra-route and k -opt are highly likely to generate infeasible solutions. We designed a Route-Saving (RS) operator to avoid this problem. This operator can be more promising for finding feasible solutions, while maintain the robustness to some extent.

When a sequence is generated, we filter out the route with the least number of task nodes. The route removing process starts with the route having the least task node, and if its task node number is less than a small predefined number l , the task nodes on the selected route are sequentially removed and reinserted into other routes. The rest of routes are ranked in descending order with respect to their idle times. The insertion process starts with the route with largest idle time, and continues with respect to the ranked order to find feasible positions for the removed task nodes. If this operation is able to produce a feasible solution with one route less, the above removing process continues iteratively. The RS operation stops when the task node numbers of all routes are larger or equal to l , or the route selected in the previous processes cannot be removed. The reason for applying the RS operator is to avoid routes that are extremely tight, which will lead to solutions with low robustness. The efficiency of the RS operator is tested later in Section 5.3.1.

4.2. Zoutendijk feasible direction algorithm

Since we have introduced an exponential expression to represent the robustness in the objective function, Zoutendijk feasible direction algorithm is applied to tackle the nonlinearity during the search process of ACO. For a given task node sequence constructed by ACO, which is mathematically represented by matrix \mathbf{X} , we are able to calculate objective 1 and the initial service beginning times array \mathbf{S}^{init} (applying (4.1)–(4.3)). In order to calculate objective 2 and the exact service times array \mathbf{S} , we apply Zoutendijk feasible direction algorithm to the following model.

$$\text{Max} \quad \sum_{i \in C_1} w_i \cdot (1 - e^{-b_i}), \quad (4.13)$$

$$\begin{aligned} \text{s.t.} \quad & S_i + t_{ij} \leq S_j + K(1 - X_{ij}), \quad \forall i, j \in C, i \neq j, X_{ij} \text{ given} \\ & (3.5)–(3.7), (3.12). \end{aligned} \quad (4.14)$$

To simplify the description of the details of the algorithm, the above model is rewritten in matrix form.

$$\text{Max} \quad g(\mathbf{S}), \quad (4.15)$$

$$\text{s.t.} \quad \mathbf{AS} \geq \mathbf{b}. \quad (4.16)$$

The detailed Zoutendijk feasible direction algorithm is given in Algorithm 1.

Finally, with all the work described above, the solution algorithm scheme is concluded in Algorithm 2.

Algorithm 1 Zoutendijk Algorithm**function** ZOUTENDIJK($\mathbf{X}, \mathbf{S}^{\text{init}}$) Step 1 : initialize iteration number $k = 0$, $\mathbf{S}^k = \mathbf{S}^{\text{init}}$, tolerance ϵ ; Step 2 : split $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$, where $\mathbf{A}_1 \mathbf{S}^k = \mathbf{b}_1$, $\mathbf{A}_2 \mathbf{S}^k > \mathbf{b}_2$; Step 3 : solve the following LP model to optimality to get the feasible direction \mathbf{d} ;

$$\text{Min } \nabla g(\mathbf{S}^k)^T \mathbf{d} \quad (4.17)$$

$$\text{s.t. } \mathbf{A}_1 \mathbf{d} \geq 0 \quad (4.18)$$

$$-1 \leq d_j \leq 1, j \in C \quad (4.19)$$

 Step 4 : if $|\nabla g(\mathbf{S}^k)^T \mathbf{d}^k| < \epsilon$, output \mathbf{S}^k ; else, a feasible direction \mathbf{d}^k is obtained; Step 5 : calculate the optimal step size, $\mathbf{b}' = \mathbf{b}_2 - \mathbf{A}_2 \mathbf{S}^k$, $\mathbf{d}' = \mathbf{A}_2 \mathbf{d}^k$,

$$\varsigma_{\max} = \begin{cases} \min\{\frac{b'_i}{d'_i} | d'_i < 0\} & \text{when } \mathbf{d}' \not\geq 0 \\ +\infty, & \text{when } \mathbf{d}' \geq 0 \end{cases}, \text{ solve the problem } \min_{0 < \varsigma < \varsigma_{\max}} g(\mathbf{S}^k + \varsigma^k) \text{ to get optimal } \varsigma^k, \text{ set } \mathbf{S}^{k+1} = \mathbf{S}^k + \varsigma^k \mathbf{d}^k,$$

 $k := k + 1$, goto Step 2.**end function****Algorithm 2** Solution Algorithm Scheme**Initialization:** $iteration = 0$, $arrangedNum = 0$, $ITER_NUM$ **for** each edge **do** initialize *Pheromone*, *Heuristic*, *Probability***end for****while** $iteration < ITER_NUM$ **do** **for** each ant **do** choose an initial node from C_1 with *Probability* **while** $arrangedNum < N$ **do** **if** feasible task nodes can be found **then** choose next feasible task node j with *Probability* **else**

start a new route

end if $arrangedNum++$ **end while** **execute** Route-saving operator compute *objective 1*, apply **Zoutendijk algorithm** for *objective 2* **end for** update *Pheromone* and *Probability* $iteration++$ **end while**

5. NUMERICAL EXPERIMENTS AND ANALYSES

In this section, experiments are designed and instances are generated to evaluate and measure the proposed model and methods. The solution procedures described in Section 4 are implemented in Java on a laptop with 3.1 GHz CPU, 64 GB RAM.

5.1. Instance generation

Test instances are randomly generated, and for the generation rules, we refer to Xue *et al.* [23]. The test area under consideration is a square region with side length equal to 200 km, and the terminal is located at the center

TABLE 2. Parameter settings.

Parameter	Parameter description	Values
$ITER_NUM$	Iteration number	1000
m	Number of ants	20
Q	Pheromone updating constant	10
$\bar{\eta}_{ij}$	Upper bound of heuristic parameter	0.9
γ	Volatilization rate of pheromone	0.95
ϵ	Tolerance of Zoutendijk Algorithm	0.01
l	Bound value of RS operator	4

of the region. All the pickup and delivery customers are randomly distributed in this region, and the direct travel time between two customers is calculated by the Euclidean distance divided by the speed of the vehicles (70 km/h). The commitment (un)packing times of customers are generated by a uniform distribution of $U[3,5]$ hours. Considering both the practical and academic relative values of the truck deployment setup cost to travel cost, we set $\rho_1 = 10$ and $\rho_2 = 1$. Based on the previous researches of Xue *et al.* [23] and Xue *et al.* [24] on the deterministic LCDP with tractor-trailer separable mode, and to balance the tradeoff between computational cost and practical value, the test instance size we used here is 20/20, which means that there are 20 pickup customers and 20 delivery customers, the total task nodes that need to be served is 80.

5.2. Algorithm parameter settings

Based on preliminary researches and numerical experiments, the main parameters are presented in Table 2; however, the details are omitted considering the article length.

5.3. Experiment results and analyses

In this section, we describe the numerical experiments that were conducted based on the model, settings, and solution algorithms described in above sections.

5.3.1. Algorithm efficiency

Since ACO and Zoutendijk algorithm are both iterative algorithms, we examine the convergence properties to reveal their efficiencies. For simplicity of presentation, we use the 20/20 scenario with $\lambda = 0.5$ (λ values do not have significant influence on the calculation speed of algorithms) to show the convergence properties of the two proposed algorithms. Firstly, experiments are conducted to compare ACO and ACO with RS operator (ACORS); both the algorithms incorporate the Zoutendijk algorithm. Secondly, we test the convergence efficiency of Zoutendijk feasible direction method with a feasible sequence generated by ACORS. The results are displayed in Figure 3.

In Figure 3a, we display the changes of up-to-now best solution objectives of two algorithms with respect to the iteration numbers. Evidently, both the algorithms are able to converge, and the improvement ratios are more than 20%. In more detail, both the algorithms converge significantly in the first 200 iterations, and the convergence trend is much more gentle in the later 800 iterations. In comparison, ACORS can always find better solutions compared with the ACO without RS operator, and the gaps are between 10% and 25%. Moreover, ACORS converges faster than ACO without RS operator. In Figure 3b, we display the convergence of objective 2 with respect to the iteration number for a determined sequence (X_{ij} values given). The convergence curve shows that the Zoutendijk algorithm is able to gain optimal service times (S_i) with a given sequence (X_{ij}) within a predefined tolerance in less than 15 iterations, and the total improvement of objective 2 is up to 50%. In conclusion, the proposed ACO incorporated with Zoutendijk algorithm is able to efficiently solve the robust LCDP under the tractor-trailer separable mode.

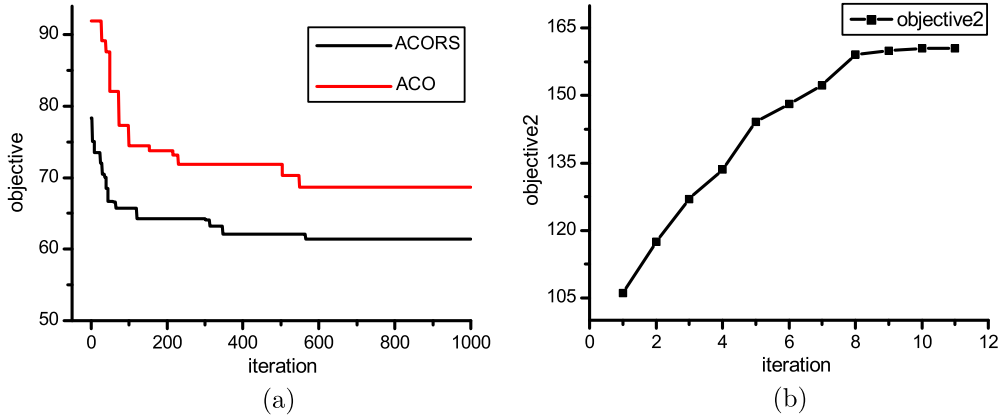


FIGURE 3. Convergence of proposed algorithms.

5.3.2. Approximation of Pareto front

In this paper, we study the tradeoff between two objectives: minimizing cost and maximizing robustness. However, there is no ultimate solution that can realize the lowest cost and the highest robustness simultaneously. In this section, we present an illustrative example of obtaining a set of non-dominated solutions for the LCDP. We conduct experiments by setting λ from 0.1 to 0.9 based on the 20/20 instance. λ approaching 0.1 means that the solutions are cost-oriented, and λ approaching 0.9 means that the solutions are robust-oriented. For each λ , we apply the solution algorithms described in Section 4 to obtain a set of solutions. Here, we illustrate the top 30 best solutions for each λ in Figure 4, where each circle denotes a solution, and the non-dominated solutions are filtered out from the 270 solutions and represented by red solid circles. The Pareto front is approximated by a curve in the figure.

As mentioned above, there is no solution with the highest robustness and lowest cost at the same time among these non-dominated solutions. Which solution or schedule should be chosen is determined by the preference of the planners and the uncertainties of the practical environment.

5.3.3. Robustness evaluation by simulation

In order to further evaluate the performance of different non-dominated schedules, we generated a series of scenarios based on the 20/20 instance to simulate practical situations. When generating the scenarios, we assume that the realized (un)packing times follow a normal distribution. For customer i , the mean value of its realized (un)packing time is its committed (un)packing time \tilde{p}_i , and the variance of its realized (un)packing time is σ^2 . Here, we only consider situations in which the realized (un)packing times are longer than their commitment values, because if the realized (un)packing time for one customer is shorter than its commitment value, the baseline schedule will always be feasible for this customer as it will have a larger feasible time window as discussed in Section 4.1.1.

Besides the uncertainties in variances, we also consider the proportion in which the realized (un)packing times vary from the commitment values, because in practical situations, not all customers will violate their commitments simultaneously. We define Pr to denote the percentage of varied customers among the entire population C_1 in one scheduling operation. For each generated scenario, $Pr \cdot |C_1|$ customers are chosen randomly, and their realized (un)packing times are generated by the normal distribution $N(p_i, \sigma^2)$. Except for these $Pr \cdot |C_1|$ customers, the realized (un)packing times of the remaining customers are the same as their commitment values. In order to ensure the reliability of the experiments, we generated 100 instances for each scenario to calculate the average values.

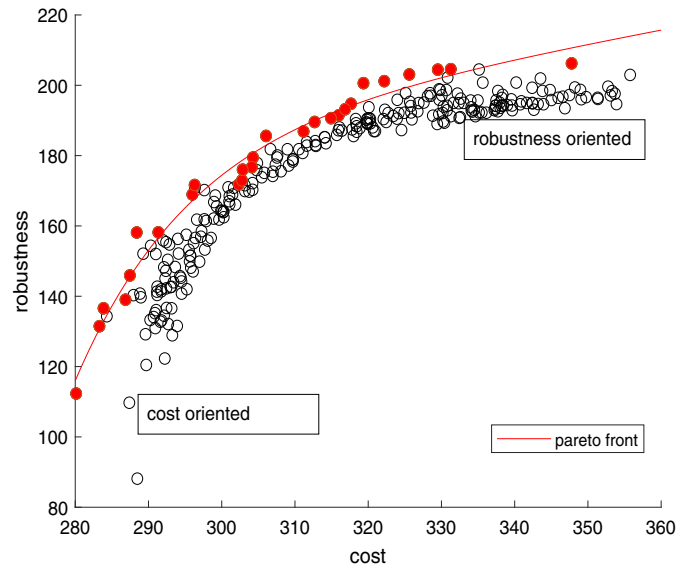


FIGURE 4. Pareto front approximation.

In this paper, we mainly use three measurements to evaluate the robustness performances of the non-dominated baseline schedules:

- (1) The number of violated (N_v) routes. Feasible baseline schedules are generated according to the commitment (un)packing times, but the realized (un)packing times may be longer than commitment values, which will cause the length of certain routes to be longer than expected or even lead to violations. Here one violation is recognized if the final realized makespan of one route exceeds the planning horizon T . N_v is the summation of all violations in one schedule.
- (2) The quantity of violations (Q_v). Q_v measures the degree of violations. For each route, when the actual makespan exceeds the planning horizon, we calculate the difference between the actual makespan of this route and the planning horizon T . The sum of the violation times of all routes is Q_v .
- (3) The rearrangement and penalty cost (C_a). Although we have proposed a bi-objective model to generate schedules to balance the cost and robustness, objective 2 has no practical meaning in practical operations, and the entire planning work is unfinished till this stage. Therefore, we have defined C_a to represent the actual cost if reactive methods are applied to deal with the violations. There are two ways to deal with violated tasks: (a) extra tractors and drivers are dispatched to deal with the violated tasks; (b) tractors and drivers continue to work with the violated tasks, but penalties will be levied for the extra work, which may cause some legitimacy issues in some countries due to the long working hours of drivers. C_a is calculated as follows: $C_a = \text{objective 1} + P_1 \cdot N_v + P_2 \cdot Q_v$. Here, P_1 and P_2 vary in different practical situations, where P_1 represents the rearrangement penalty of tractors and drivers, and P_2 is the penalty cost for extra working hours of tractors and drivers, and they are decided based on practical situations.

The performance of different baseline schedules under different scenarios are presented in Table 3. Baseline schedules are obtained based on the solution algorithms proposed above with different λ values. Different scenario instances are generated with combinations of $Pr = 0.2, 0.4, 0.6, 0.8$ and $\sigma^2 = 2, 4, 6, 8$. To ensure reliability, for each pair of Pr and σ^2 , we generate 100 different instances and calculate the average values of N_v , Q_v , and C_a .

The following aspects can be observed from Table 3:

TABLE 3. The calculation of N_v and Q_v according to Pr and σ^2 .

		λ																	
		0.1		0.2		0.3		0.4		0.5		0.6		0.7		0.8		0.9	
Pr	σ^2	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v	N_v	Q_v
0.2	2	0.52	0.38	0.79	0.67	1.17	0.97	1.06	0.96	1.98	1.99	1.53	1.62	1.96	2.07	1.45	1.17	1.72	1.68
0.2	4	1.11	1.35	1.19	1.64	1.86	2.85	2.11	3.09	2.69	4.33	2.26	3.53	2.82	4.28	2.33	3.26	2.73	4.26
0.2	6	0.95	1.52	1.60	2.61	2.06	3.22	2.17	3.28	2.98	5.50	2.43	4.54	2.87	5.24	2.49	3.94	2.83	4.77
0.2	8	1.46	2.50	2.05	3.59	2.62	4.67	3.05	5.52	3.55	7.60	2.95	6.72	3.53	7.69	2.94	6.15	3.30	6.72
0.4	2	0.95	0.76	1.73	1.76	2.50	2.48	2.32	2.60	3.83	4.49	2.99	3.58	3.94	4.75	2.94	3.50	3.50	4.15
0.4	4	1.67	2.32	2.83	4.69	3.62	5.70	3.70	5.82	5.24	9.47	4.36	8.31	5.20	9.61	4.07	6.80	4.70	8.50
0.4	6	2.42	3.95	3.58	6.52	4.39	8.68	4.73	10.08	5.63	13.67	5.17	11.89	5.85	14.13	4.99	10.51	5.78	12.59
0.4	8	3.23	6.68	4.41	10.48	5.01	13.11	5.62	14.74	6.28	20.39	5.49	17.37	6.31	18.68	5.76	15.17	6.06	18.88
0.6	2	1.33	0.99	2.35	2.88	3.77	3.86	3.56	4.01	5.17	7.31	4.44	5.81	5.29	7.19	4.32	5.22	5.15	6.96
0.6	4	2.66	3.26	3.78	6.33	5.11	8.82	5.38	9.65	6.71	15.20	5.75	12.05	6.86	14.59	5.93	10.82	6.66	13.42
0.6	6	3.57	6.48	5.06	10.23	5.79	13.85	6.53	15.98	7.14	20.48	6.53	18.11	7.85	21.72	7.12	16.85	7.57	21.37
0.6	8	4.56	9.05	6.21	14.77	6.65	17.47	7.19	21.64	7.95	28.78	7.32	23.46	8.63	28.89	7.21	21.75	8.36	27.29
0.8	2	2.00	1.71	3.40	4.21	4.64	5.03	5.03	6.23	6.79	10.44	5.71	8.41	6.96	9.96	5.65	7.74	6.59	10.01
0.8	4	3.89	5.61	5.51	9.05	6.39	12.07	6.99	14.71	8.09	21.13	7.27	17.10	8.74	21.10	7.58	15.74	8.73	20.04
0.8	6	4.78	8.41	7.16	16.01	7.31	18.85	8.00	22.68	8.82	31.29	8.28	24.26	9.60	29.31	8.47	23.89	9.50	30.04
0.8	8	5.82	13.33	8.28	22.71	8.08	26.05	8.79	32.64	9.58	41.77	9.35	34.76	10.27	41.84	9.39	33.31	10.40	41.56

Notes. Pr : the probability of varied customers. σ^2 : variances of varied customers. N_v : the total number of violated tractor routes. Q_v : the total quality of violations. λ : the weights for robustness.

- (1) Along the vertical direction, for each baseline schedule, both N_v and Q_v are increasing with respect to σ^2 for a determined Pr , while the trend is the same with respect to Pr for a determined σ^2 . These trends are obvious because a high level of uncertainties will lead to more violations.
- (2) Along the horizontal direction, for each pair of Pr and σ^2 , both N_v and Q_v show an increasing trend with respect to λ , which means that the cost-oriented schedules are more sensitive to uncertainties.

Figure 5 presents the actual cost for reactive strategies, where $P_1 = 10$ and $P_2 = 5$, $\lambda = 0.1$ stands for the most cost-oriented schedule, and $\lambda = 0.9$ represents the most robust-oriented schedule. From this figure, we conclude that robust-oriented schedules are more stable to uncertainties with the given P_1 and P_2 values; that is to say, cost-oriented schedules tend to have a lower cost when the uncertainty levels (Pr and σ^2) are low. However, with the increase in uncertainty levels, their costs increase much faster than the robust-oriented schedules.

5.3.4. Parameter analyses of P_1 and P_2

In previous subsections, we have mentioned that P_1 is the rearrangement costs for tractors and drivers and P_2 is the penalty cost for tractors and drivers. Since in practical situations, the strategies to deal with violations may vary due to planners' preferences and legitimacy issues, P_1 and P_2 can also have different values. Besides the results shown in Figure 5, we conducted experiments on different P_1 and P_2 combinations as shown in Figure 6, where $\lambda = 0.1$ represents the most cost-oriented schedule and $\lambda = 0.9$ represents the most robust-oriented schedule. The main observations in Figure 6 can be summarized as follows. In Figure 6a and b, where the rearrangement costs and penalty costs are relatively low, cost-oriented schedules have slightly dominant advantages over robust-oriented schedules when the uncertainty levels (Pr and σ^2) are low; however, as the uncertainty levels rise, some robust-oriented schedules start showing their advantages. On the other hand, as the rearrangement and penalty costs are relatively higher as shown in Figure 6c and d, with the increase in uncertainty levels, robust-oriented schedules trend to outperform cost-oriented schedules. In conclusion, when

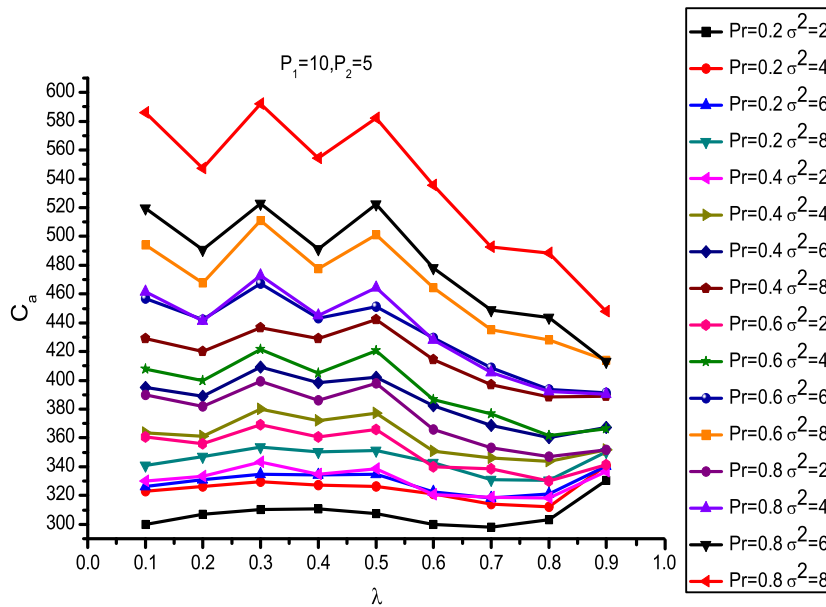


FIGURE 5. Actual cost C_a according to Pr and σ^2 for different λ .

the rearrangement and penalty costs are relatively low, the cost-oriented schedules and robust-oriented schedules have relatively fair results. On the other hand, the rearrangement and penalty costs are high, robust-oriented schedules have more advantages than cost-oriented schedules.

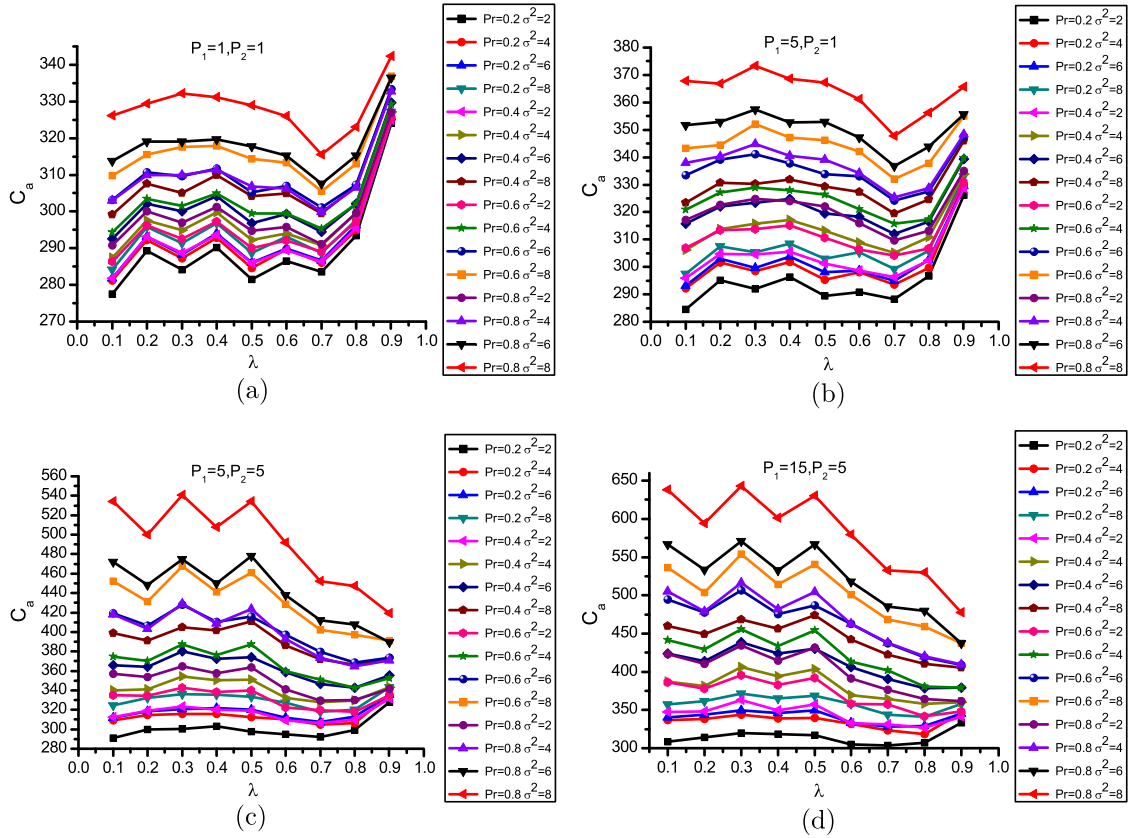
5.3.5. Schedule selection preferences

In order to provide some management insights to the planners, we extend the experiments to show how to adopt proper schedules for different scenarios. In Section 5.3.2, we presented how to approximate the Pareto front of the solution space and obtain non-dominated solutions. Each solution corresponds to a λ , which reflects the robustness of this solution. As the value of λ changes from 0.1 to 0.9, it means that the solutions are changing from cost-oriented to robust-oriented, and they are denoted by $S1, S2, \dots, S9$ in the following parts. Experiments are conducted with respect to the changes in uncertainty levels (Pr and σ^2). To explore the influence of uncertainty levels, experiments are conducted in the following two aspects using a control variable method:

- (1) for the first two groups of experiments, Pr is set as: 0.2 and 0.8, respectively, and instances are generated with σ^2 changing from 0 to 8;
- (2) for the following two groups of experiments, σ^2 is set to 2 and 8, respectively, and instances are generated with Pr varying from 0 to 0.8.

All the instances are repeated 100 times to achieve average values. The results are illustrated in Figure 7, and the best schedule suggestions are shown in the upper part of each subfigure. The lower part of each subfigure shows the evolution trends of the actual cost C_a for different schedules, which explains the choices of the upper parts of the subfigures, and in order to make the figure clean and clear, we only present the curves of a few representative schedules.

The vertical axis of the upper parts of each subfigure represents different baseline schedules, where $S9$ stands for the most robust-oriented baseline schedule, and $S1$ stands for the most cost-oriented schedule. The horizontal axes represent the changes in λ and σ^2 . Figure 7a and b show that with the increase in σ^2 , the schedule choices

FIGURE 6. Actual costs C_a for schedules with different P_1 and P_2 values.

will change from cost-oriented to robust-oriented, but Figure 7b shows a much more obvious trend to choose robust-oriented schedules than Figure 7 since it has a larger Pr value. Figure 7c and d show that with the increase in Pr , schedule choices also have a tendency to change from cost-oriented to robust-oriented, Figure 7d shows a more obvious trend of choosing robust-oriented schedules than Figure 7c. Besides, Pr seems to have a more obvious influence on the choice of schedules than σ^2 .

5.3.6. Human experience of fleet planners

Since, in this paper, we study the proactive strategy in which the plans are made by fleet planners before complete information is obtained, the accumulated experience of planners is important to make baseline schedules in the long-term orientation. Based on the history of cooperations, fleet planners may have anticipations for each customer of being punctual or late. If the anticipation information is considered in the generation process of baseline schedules, it will possibly improve the robustness of the obtained schedules.

Experiments are conducted for different anticipation information based on varied human experience of fleet planners. We assume that 20 out of 40 customers have longer (un)packing times than their commitment (un)packing values, and fleet planners can predict the actual scenarios in different levels. For ease of presentation, two different representative cases are considered in the experiments with different anticipation levels:

Anti5: 5 out of 20 uncertain customers are predicted correctly and are assigned with larger w_i values as described in Section 3.2.

Anti20: 20 out of 20 uncertain customers are predicted correctly and are assigned with larger w_i values.

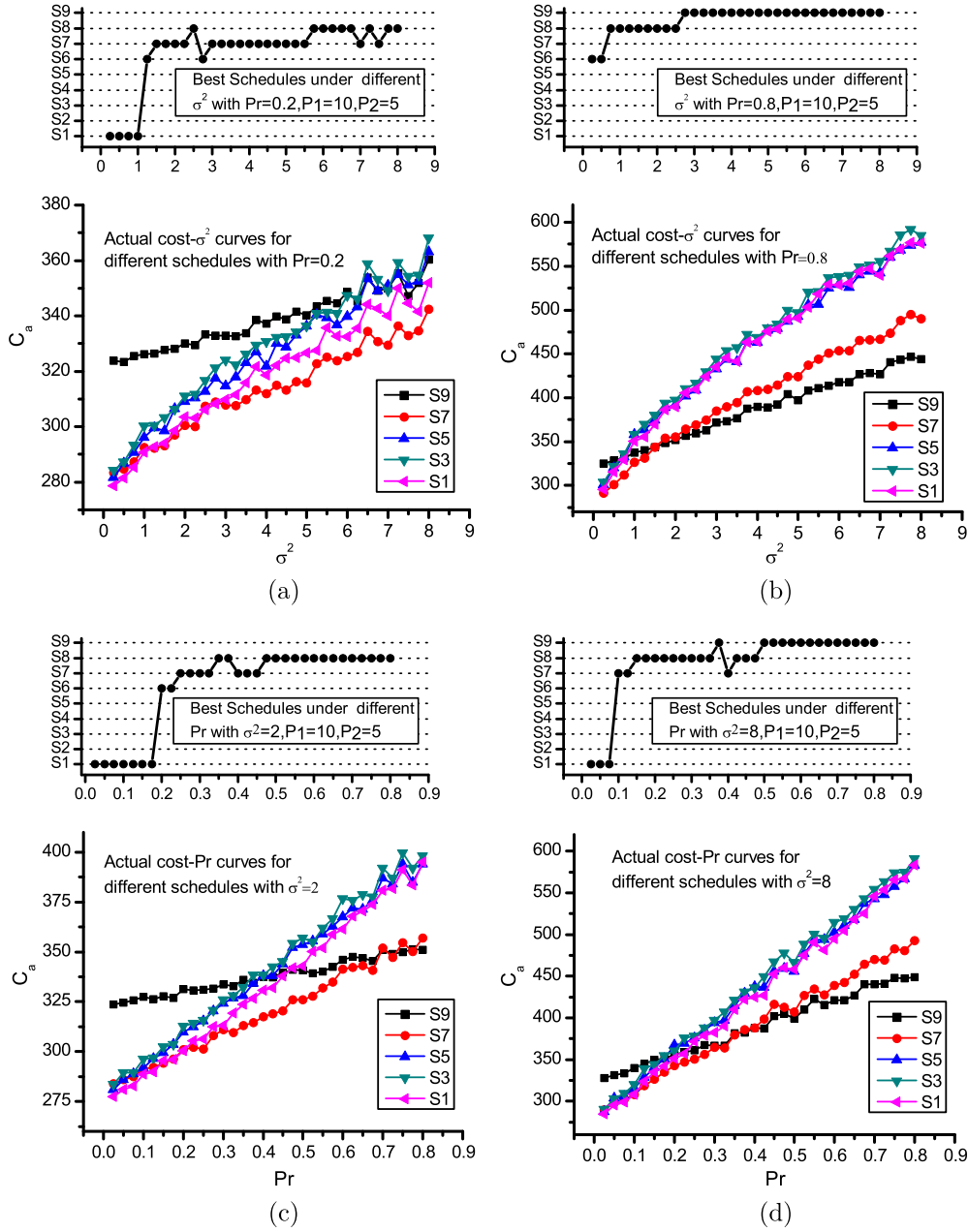


FIGURE 7. Examples of choices and details for planners.

We solve the two models with varied w_i values and obtain two groups of schedules. Each schedule is tested through the generated instances with variances ranging from 2 to 16 to calculate C_a values, and experiments are conducted 100 times to obtain average values. For better illustration, parts of the results are presented in Figure 8a and b, from which we find that more anticipation information leads to more cost savings. Furthermore, Figure 8c shows the minimum costs among all schedules under the two different levels of anticipations. When the variance is low, which means that the uncertainty levels are insignificant, the costs under different anticipation

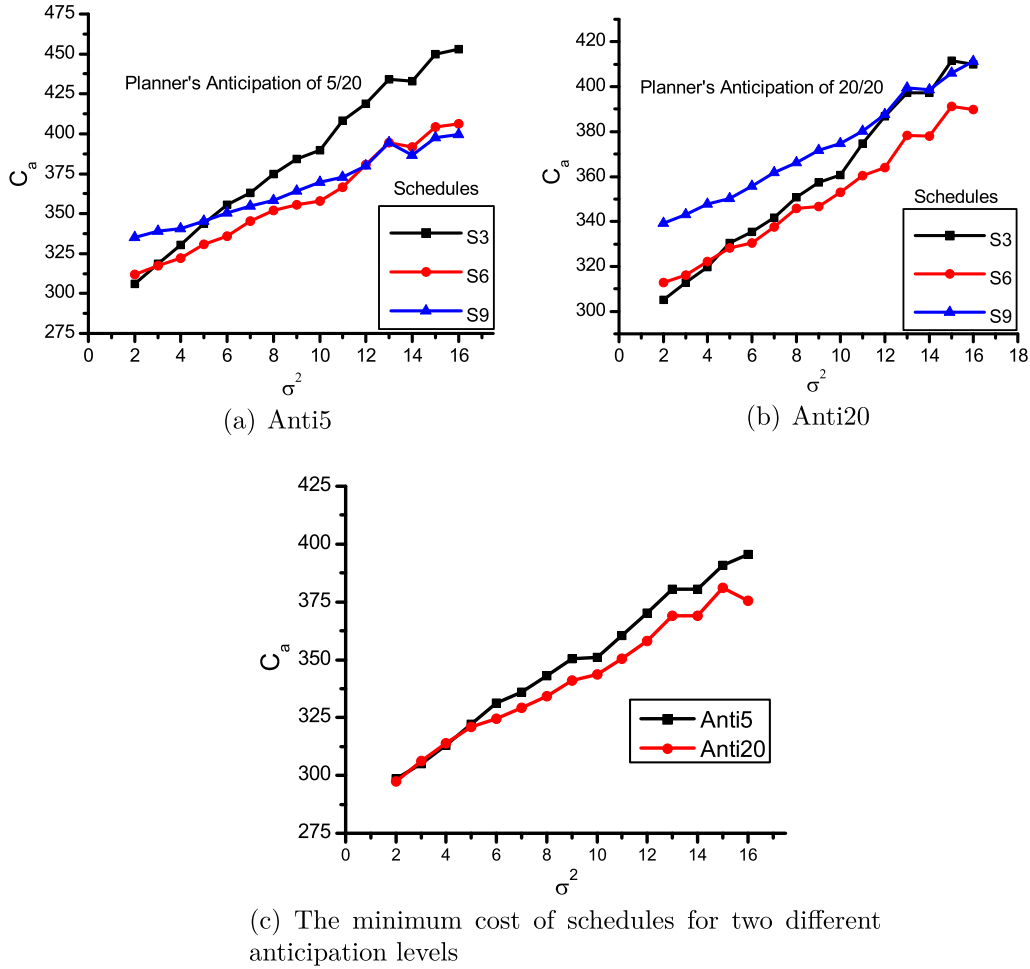


FIGURE 8. Performance comparisons of different levels of anticipation based on planner's experience.

information levels are very close; however, as the uncertainty level increases, the importance of anticipation information becomes increasingly evident. The experiments in this subsection validate the importance of correct anticipation for uncertainties in practical situations.

5.3.7. Comparisons among different planning strategies

Though we studied the importance of anticipation information in Section 5.3.6, in realistic environments, fleet planners need to make plans without any prior information; therefore, heuristic strategies other than the bi-objective model proposed in this paper are often adopted. One of the most commonly applied heuristic strategy is to simply extend the committed (un)packing times in the planning process (greedy extension strategy) to generate robust baseline schedules. In order to evaluate and compare the performances of strategies, we conducted experiments on three different scheduling strategies:

- (1) Greedy extension strategy (Greedy 1): the simplest heuristic strategy for fleet planners is to extend all customers' (un)packing times from the commitment values with certain extend levels (which means the (un)packing times used to generate baseline schedules are multiplied by the extend levels). Here we

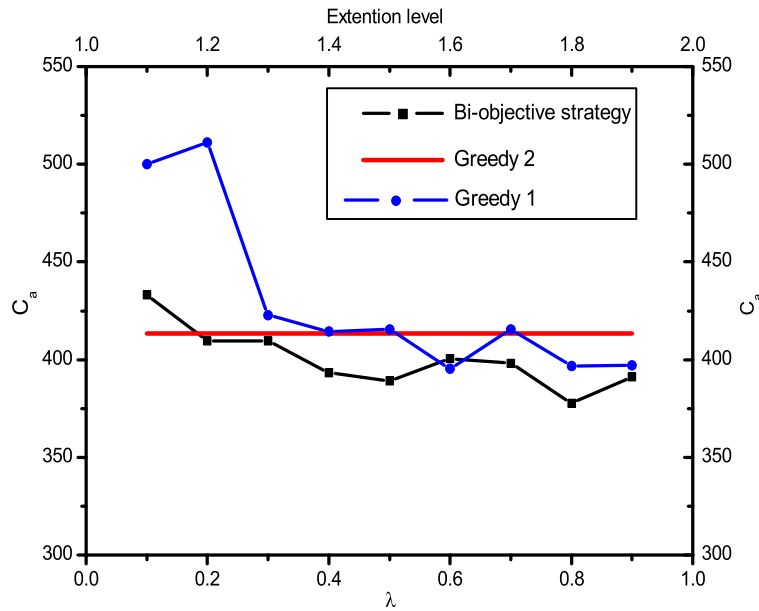


FIGURE 9. Comparisons among different planning strategies.

conducted experiments with extension levels ranging from 1.1 to 1.9 with in steps of 0.1; thus 9 schedules were generated.

- (2) Greedy extension strategy with complete information (Greedy 2): we consider this strategy as a baseline strategy of this subsection, which means that all the uncertain information are predicted accurately before the planning process, and the solution process is the same as the greedy extension strategy with the extension levels (ranging from 1.1 to 1.9) for different customers being related to their varied levels (σ^2 ranging from 2 to 8) from the committed values.
- (3) Bi-objective strategy: a group of nine schedules were generated by the model and algorithms proposed in this paper without any information given, except for the commitment (un)packing times.

In order to evaluate the schedules generated by the three strategies above, 20 out of 40 customers were set as uncertain and instance generation processes, and experiments were repeated 100 times for calculating the average C_a values. The results are presented in Figure 9, where three different strategies are presented in one figure to make a visual presentation; however, there is no physical meaning in comparing the different strategies directly. The upper x-axis stands for the extend levels of the heuristic extension strategy, the lower x-axis stands for λ of the bi-objective model, and the y-axis stands for the C_a values of different schedules. The following aspects are observed in Figure 9:

- (1) C_a tends to decrease as the extension level of the heuristic extension strategy and λ of the bi-objective strategy increase.
- (2) We expect the bi-objective strategy schedules to have lower C_a than the heuristic extension strategy schedules and complete human experience strategy schedules in most of the cases.
- (3) Complete human experience strategy has compatible C_a with the other two strategies when their robustness levels are low, but as their robustness levels increase, complete human experience strategy starts being overtaken by the other two strategies.

The bi-objective strategy proposed in this paper outperforms the greedy extension strategy and greedy extension strategy with complete information in most of the cases, and it can generate robust schedules that have lower costs for most of the situations compared with the other two strategies.

6. CONCLUSION

This paper studies proactive strategies for making robust schedule plans for the Local Container Drayage Problem with Separable tractors and trailers, and additionally with consideration of (un)packing time Uncertainties (LCDPSU), and proposes a bi-objective model and algorithms to generate robust baseline schedules based on committed (un)packing times of customers. The contributions of this study mainly include three aspects:

- (1) A “model robust” bi-objective model is proposed to balance the trade-off between operational cost and robustness for the routing and scheduling of tractors, containers, and drivers in LCDPSU. The first objective of the proposed model is the same as most of the traditional deterministic LCDP researches on this topic, which basically minimizes the truck deployment setup cost and travel cost. The second objective is to combine a time buffer insertion strategy, which is used to incorporate model robustness, with the deterministic LCDP model and increase the robustness of baseline schedules.
- (2) An ACO scheme based algorithm incorporated with the Zoutendijk feasible direction algorithm is proposed to solve this model. Candidate service sequences are generated by ACORS, and exact optimal schedules are calculated by the Zoutendijk feasible direction algorithm. Unlike the existing researches, this study attempted to combine an exact method with a meta-heuristic algorithm to solve a practical problem in LCDPSU.
- (3) Numerical experiments and parameter analyses are conducted, and management suggestions are given to fleet planners of container drayage operations so that they can make proactive schedule plans to face uncertainties. The proposed model and methods are able to generate schedules that consider the costs and uncertainties at the same time, and the solutions outperform some basic heuristic strategies.

In the future, we will try to analyze the structure of this problem further, and apply exact algorithms for the model. At the same time, we will also consider other uncertain factors such as travel time uncertainties.

FUNDING STATEMENT

The work is supported by the National Natural Science Foundation of China under Grant No. 71502112, 71872092 and 71472108, Shenzhen Municipal Science and Technology Innovation Committee under Grant No. JCYJ20170302144108956, and Natural Science Foundation of SZU under Grant No. 000154.

Acknowledgements. The authors would like to thank the two anonymous referees for their valuable comments and suggestions which improved the quality of this paper.

REFERENCES

- [1] M.A. Al-Fawzan and M. Haouari, A bi-objective model for robust resource-constrained project scheduling. *Int. J. Prod. Econ.* **96** (2005) 175–187.
- [2] K. Braekers, A. Caris and G.K. Janssens, Bi-objective optimization of drayage operations in the service area of intermodal terminals. *Transp. Res. Part E-logist. Transp. Rev.* **65** (2014) 50–69.
- [3] L. Coslovich, R. Pesenti and W. Ukovich, Minimizing fleet operating costs for a container transportation company. *Eur. J. Oper. Res.* **171** (2006) 776–786.
- [4] M. Dorigo, V. Maniezzo and A. Colorni, Ant system: Optimization by a colony of cooperating agents. *Syst. Man Cybern.* **26** (1996) 29–41.
- [5] A. Ghezelsoflu, M.D. Francesco, A. Frangioni and P. Zuddas, A set-covering formulation for a drayage problem with single and double container loads. *J. Ind. Eng. Int.* **14** (2018) 665–676.
- [6] C.E. Gounaris, W. Wiesemann and C.A. Floudas, The robust capacitated vehicle routing problem under demand uncertainty. *Oper. Res.* **61** (2013) 677–693.

- [7] J. Han, C. Lee and S. Park, A robust scenario approach for the vehicle routing problem with uncertain travel times. *Transp. Sci.* **48** (2014) 373–390.
- [8] X. Han, Z. Lu and L. Xi, A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *Eur. J. Oper. Res.* **207** (2010) 1327–1340.
- [9] W. Herroelen and R. Leus, Project scheduling under uncertainty: Survey and research potentials. *Eur. J. Oper. Res.* **165** (2005) 289–306.
- [10] A. Imai, E. Nishimura and J.R. Current, A lagrangian relaxation-based heuristic for the vehicle routing with full container load. *Eur. J. Oper. Res.* **176** (2007) 87–105.
- [11] H. Julia, M.M. Dessouky, P.A. Ioannou and A. Chassiakos, Container movement by trucks in metropolitan networks: Modeling and optimization. *Transp. Res. Part E-logist. Transp. Rev.* **41** (2005) 235–259.
- [12] O. Lambrechts, E. Demeulemeester and W. Herroelen, Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *J. Sched.* **11** (2008) 121–136.
- [13] C. Lee, K. Lee and S. Park, Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *J. Oper. Res. Soc.* **63** (2012) 1294–1306.
- [14] C. Liu, C. Zhang and L. Zheng, A bi-objective model for robust yard allocation scheduling for outbound containers. *Eng. Optim.* **49** (2017) 113–135.
- [15] J.M. Mulvey, R.J. Vanderbei and S.A. Zenios, Robust optimization of large-scale systems. *Oper. Res.* **43** (1995) 264–281.
- [16] S. Shiri and N. Huynh, Optimization of drayage operations with time-window constraints. *Int. J. Prod. Econ.* **176** (2016) 7–20.
- [17] Y. Song, Z. Jiantong, L. Zhe and Y. Chunming, An exact algorithm for the container drayage problem under a separation mode. *Transp. Res. Part E-logist. Transp. Rev.* **106** (2017) 231–254.
- [18] T. Stutzle and H.H. Hoos, Max-min ant system. *Future Gener. Comput. Syst.* **16** (2000) 889–914.
- [19] I. Sungur, F. Ordonez and M.M. Dessouky, A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *Iie Trans.* **40** (2008) 509–523.
- [20] D. Tjokroamidjojo, E. Kutanoglu and G.D. Taylor, Quantifying the value of advance load information in truckload trucking. *Transp. Res. Part E-logist. Transp. Rev.* **42** 340–357.
- [21] S. Wang and Q. Meng, Robust schedule design for liner shipping services. *Transp. Res. Part E-logist. Transp. Rev.* **48** (2012) 1093–1106.
- [22] X. Wang and A.C. Regan, Local truckload pickup and delivery with hard time window constraints. *Transp. Res. Part B-methodol.* **36** (2002) 97–112.
- [23] 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. Part E-logist. Transp. Rev.* **62** 136–150.
- [24] Z. Xue, C. Zhang, P. Yang and L. Miao, A combinatorial benders' cuts algorithm for the local container drayage problem. *Math. Probl. Eng.* **2015** (2015) 1–7.
- [25] R. Zhang, W.Y. Yun and H. Kopfer, Heuristic-based truck scheduling for inland container transportation. *OR Spectr.* **32** (2010) 787–808.
- [26] R. Zhang, W.Y. Yun and K.I. Moon, Modeling and optimization of a container drayage problem with resource constraints. *Int. J. Prod. Econ.* **133** (2011) 351–359.
- [27] L. Zhen and D.F. Chang, A bi-objective model for robust berth allocation scheduling. *Comput. Ind. Eng.* **63** (2012) 262–273.