WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 31 (2024) 167–191
DOI: 10.1111/itor.13154

# Robust traveling salesman problem with drone: balancing risk and makespan in contactless delivery

Lei Zhao[a], Xinhua Bi[a,*], Zhaohui Dong[a], Ni Xiao[b] and Anni Zhao[a]

[a]*School of Business and Management, Jilin University, Changchun 130012, China*
[b]*College of Economics and Management, Jilin Agricultural University, Changchun 130118, China*
*E-mail: zhaolei17@mails.jlu.edu.cn [Zhao]; bixh@jlu.edu.cn [Bi]; dzh20@mails.jlu.edu.cn [Dong]; xiaoni@jlau.edu.cn [Xiao]; zhaoan15@mails.jlu.edu.cn [Zhao]*

Received 16 April 2021; received in revised form 5 April 2022; accepted 19 April 2022

**Abstract**

The spread of COVID-19 outbreak has promoted truck-drone delivery from trials to commercial applications in end-to-end contactless solutions. To fully integrate truck-drone delivery in contactless solutions, we introduce the robust traveling salesman problem with a drone, in which a drone makes deliveries and returns to the truck that is moving on its route under uncertainty. The challenge is to find, for each customer location in truck-drone routing, an assignment to minimize the expected makespan. Apart from the complexity of this problem, the risk of synchronization failure associated with uncertain travel time should be also considered. The problem is first formulated as a robust model, and a novel efficient frontier heuristic is proposed to solve this model. By coupling the implicit adaptive weighting with epsilon-constraint methods, the heuristic generates a series of scalarized single-objective problems, where the goal is to minimize expected makespan under the constraint of synchronization risk. The experiment results show that the robust (near-)optimal solutions offer a considerable reduction in risk, yet only hint at a small increase in makespan. The heuristic in the present study is effective to construct approximations of Pareto frontier and allows for assignment decisions in *a priori* or *a posteriori* manner.

*Keywords:* truck-drone; traveling salesman problem; synchronization; contactless delivery; efficient frontier; robust optimization

## 1. Introduction

The spread of COVID-19 outbreak has accelerated the use of drone delivery. At the initial stage, online sales have gathered steam and, due to strict lockdown, logistics activities have been suspended (Singh et al., 2021). Facing these challenges, a China's e-commerce mega player JD.com as well as many other e-commercial enterprises have presented with end-to-end contactless solutions, which

*Corresponding author.


Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

use drones to provide logistics service in restricted regions to keep their workers and customers safe (Wolf, 2020). Even though the policy is starting to catch up with drone technology and lowering the bar of business applications in many countries, only some e-commercial enterprises will continue to operate drone fleets when the immediate crisis ends. The specific use of drones to ensure the supply of essential products (e.g., foods and medicines) while maintaining social distancing seems to more relief logistics fighting crisis at any cost, rather than commercial applications seeking the most profit and balancing risk. This learning motivates us to focus on cost-effective ways to integrate drones into supply chains.

Fully integrating drones into supply chains requires synchronizations, a lot more than drone technology. From an integration perspective, Murray and Chu (2015) first demonstrated that truck-drone, as a delivery system, could play their respective advantages. Due to short-range, limited capacity and battery endurance, the drone has to return to a truck (as mobile depot and recharging platform) for supplement after each delivery. The truck has long-range and large capacity, but it is often slower than the drone because of traffic congestions or restrictions in inner cities. Carlsson and Song (2018) elaborated that the improvement of efficiency for truck-drone coordination was proportional to the square root of the ratio of both vehicles' speeds. Some researchers, for example, Wang et al. (2017) and Poikonen and Golden (2020a), get similar conclusions by theoretical analysis based on the determined conditions (e.g., truck and drone take the fixed maximum speed). However, a few scholars are aware of uncertainties in practice. Raj and Murray (2020) found that flying at high speeds could dramatically reduce flight ranges, thus limiting the effectiveness of the truck-drone delivery system. This finding also indicates that significant time or cost savings can be achieved by operating drones at variable speeds according to actual traffic conditions. Although considering the traffic uncertainty, these researches focus on synchronizations efficiency rather than synchronizations risk. Because traffic conditions around customers are real-time changing, the truck travel time is also changing in these regions. To ensure that the drone can be effective to make deliveries and successfully retrieved by the truck, a reliable schedule should be well considered under traffic uncertainty.

There are three levels of impaction on the truck-drone delivery networks if truck travel time is perturbated under uncertain traffic conditions. (1) Normal stop. The flight time of drone is not more than its endurance and it can hover in the air automatically until the truck arrives rendezvous point, or fly to the truck directly. In this case, the current delivery operations of air-borne drone and ground-based truck are slightly unsynchronized, but the air-borne drone can adjust its flying model or rendezvous point automatically to compensate for the perturbations of truck travel time. The consequence is that the follow-up delivery operations would be postponed, and therefore, the makespan of the truck-drone delivery system is longer. (2) Operational stop. The flight time of drone is more than its endurance and it can land at a safety point nearby until the truck arrives. The consequence is that current delivery operations would be stopped and follow-up delivery operations had to be postponed, and therefore, the schedule of truck-drone delivery system is inefficient. (3) Emergency stop. The drone flight time is more than its endurance and it cannot land on a safety point nearby or even crash after running out of battery. The consequence is that both current and follow-up delivery operations had to cease, and therefore, the schedule of truck-drone delivery system is infeasible.

The uncertainty of traffic conditions in big cities is a common and important context for a combined truck-drone delivery system. The risk for a slightly unsynchronized network with normal

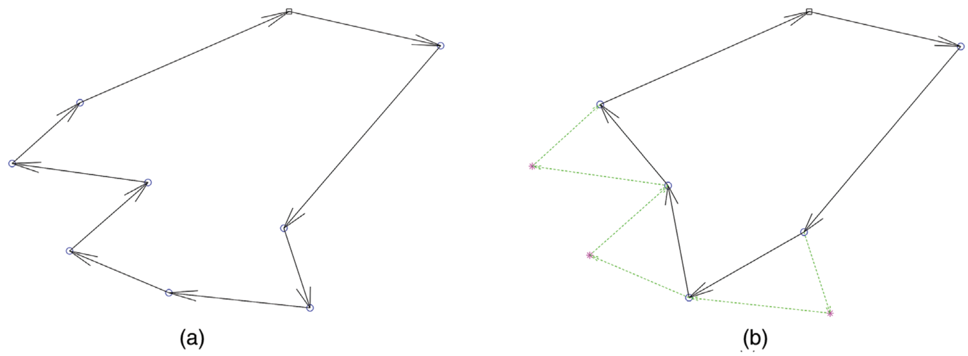


Fig. 1. An illustration of solutions for (a) TSP and (b) TSP-D. *Notes*: The paths of truck and drone are indicated by solid and dashed lines, respectively. The depot, truck customers, and drone customers are indicated by the square, circles, and stars, respectively.

stops may be a small efficiency loss, however, the risk for that with operational stops or emergency stops would be a chain reaction. In this paper, the operational stop and emergency stop are defined as synchronization failure, considering that the air-borne drone cannot compensate for the travel time perturbations of a ground-based truck (e.g., the drone runs out of the battery). Additionally, if traffic conditions are uncertain, the truck-drone schedule that balances the risk of synchronization failure and expected makespan is a challenging task. The goal of this paper is to model the truck-drone schedule problem and develop a solution method to balance the risk and makespan.

From the transportation planning perspective, the truck-drone schedule is a three-level traveling salesman problem with a drone (TSP-D) under traffic uncertainty. At the first level, routing decisions determine the predetermined sequence in which the customers are serviced orderly, which is a traveling salesman problem (TSP), as shown in Fig. 1a. At the second level, assignment decisions determine which vehicle, drone, or truck will serve which customer. At the third level, if a drone is assigned to visit some customers, location decisions determine where to launch and retrieve the drone.

If truck routing decisions are first, we can constrain the launch and retrieve locations that are all on the predetermined truck route, see Fig. 1b. One common assumption in the literature is that the drone and truck can rendezvous at any customer node (Murray and Chu, 2015; Wang et al., 2017; Agatz et al., 2018; Es Yurek and Ozmutlu, 2018; Ha et al., 2018; Mbiadou Saleu et al., 2018; Freitas and Penna, 2020; Murray and Raj, 2020; Raj and Murray, 2020; Boccia et al., 2021; Dell'Amico et al., 2021). Another assumption in the literature is that the drone and truck can rendezvous at any point of the truck route (Carlsson and Song, 2018; Schermer et al., 2019; Schermer et al., 2020; Poikonen and Golden, 2020a; Poikonen and Golden, 2020b). Without considering the conditions for launching or retrieving a drone, these assumptions may be appropriate to reduce computational complexity. However, in other cases, for example, under traffic congestions shown in Fig. 2a, these assumptions do neglect the risk of synchronization failure when the drone and truck rendezvous at a customer location in the congested area.

If drone location decisions are first, we can insert the launch and retrieve locations (truck stops) into truck route, see Fig. 2b. Truck stops are similar to intermediate nodes (noncustomer) en route,

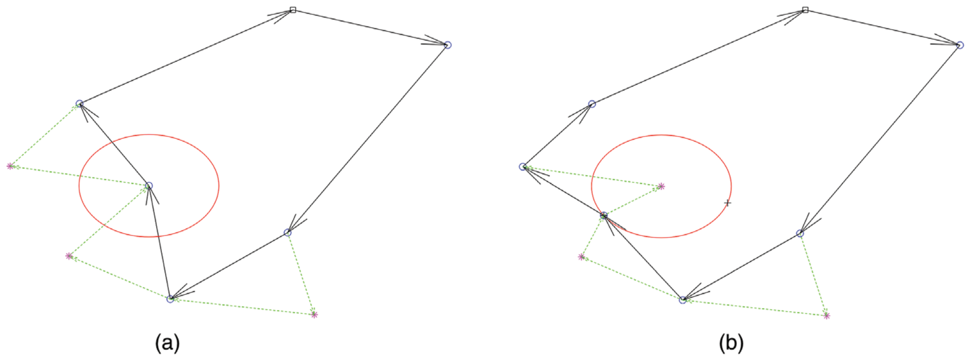


Fig. 2. An illustration of solutions under uncertainty for (a) TSP-D and (b) RTSP-D. *Notes*: The paths of truck and drone are indicated by solid and dashed lines, respectively. The depot, truck customers, and drone customers are indicated by the square, circles, and stars, respectively. The range of the traffic congestions is indicated by the red circle. Candidate truck stops are indicated by the plus signs where a drone can launch from or return to the truck. The left plus sign in panel (b) is chosen as a truck stop that is linked by two new solid arcs. The length of these arcs is described as uncertain travel time calculated by Equations (11) and (12) in Section 4.1.

which is not related to providing service but is necessary to keep the vehicle operational naturally (Martins et al., 2021). The difference is that truck stops are moving points, which are not on the predetermined truck route. Inserting truck stops can extend the range of drone delivery. Of course, the argument can be made that inserting truck stops could be an attractive alternative to enhance the robustness of schedules.

To enhance the robustness of schedules, this paper introduces the RTSP-D as an extension of the TSP-D defined by Agatz et al. (2018) under traffic uncertainty. In our problem, a drone makes deliveries and returns to a truck that is moving on its route with uncertain travel times. The objectives are to minimize the expected makespan, meanwhile to reduce risk of synchronization failure that the drone has not been retrieved by the truck within its maximum flight time. Compared to the TSP-D, RTSP-D is a more complex bi-objective problem in which truck stops are additional decision variables and travel times are uncertain parameters that affect the tractability of the problem. Since the TSP-D generalizes the NP-hard Hamiltonian cycle problem (Agatz et al., 2018), the RTSP-D is also NP-hard.

To the best of our knowledge, the present study related to RTSP-D has never been reported. The main contributions are as follows: (1) We develop a robust model of RTSP-D, which can minimize the risk of synchronization failure and expected makespan simultaneously. (2) We propose a state-of-the-art heuristic to approximate the efficient frontier, which can visualize trade-offs between the risk and makespan, allowing for assignment decisions in *a priori* or *a posteriori* manner. (3) Our experiment results show that the robust (near-)optimal solutions offer a small increase in expected makespan against a large reduction in risk. The results also indicate that e-commerce enterprises can take a robust approach to fully integrate truck-drone delivery in end-to-end contactless solutions.

The remainder of this paper is organized as follows. In Section 2, we present a literature review. In Section 3, we introduce the TSP-D and illustrate that the solutions of solely minimizing expected makespan are vulnerable to uncertain travel times. In Section 4, we formulate the

RTSP-D to balance synchronization risk and expected makespan. In Section 5, we propose the novel efficient frontier heuristic. In Section 6, we provide a detailed analysis of the heuristic performance. In Section 7, we finally conclude with some future research directions.

## 2. Related Literature

There are two literature streams related to our work to be reviewed: the truck-drone routing problem and the robust optimization of vehicle routing problem (VRP).

### *2.1. Truck-drone routing problem*

The RTSP-D is conceptually related to the flying sidekick traveling salesman problem (FSTSP), which was first introduced by Murray and Chu (2015). Since then, several variants or special cases of the FSTSP have been proposed. Carlsson and Song (2018) analyzed the benefits of coordination between one truck and one drone and they demonstrated that the efficiency improvement is proportional to the square root of the speed ratios. Considering different objectives, Ha et al. (2018) gave a variant to minimize operational cost. Moshref-Javadi et al. (2020) developed a multi-trip delivery model with one truck and several drones to minimize the customer waiting times. Murray and Raj (2020) proposed the multiple flying sidekicks traveling salesman problem (mFSTSP), and they revealed that the potential time savings decline with adding of employing drones. Raj and Murray (2020) extended the mFSTSP, in which the drone speeds were decision variables to optimize the trade-offs between the speed and the range. A comprehensive review of these problems is presented in Li et al. (2021), Rojas Viloria et al. (2021), and Macrina et al. (2020).

As a generalization of the FSTSP, Agatz et al. (2018) proposed the TSP-D, in which the drone could launch from or land at the same customer node repeatedly. Following the method of route-first and cluster-second, they provided two heuristics based on local search and dynamic programming to solve the TSP-D. Considering the NP-hard nature of truck-drone routing problems, many scholars are also committed to developing algorithms of the exact or heuristic methods, for example, Ferrandez et al. (2016), Bouman et al. (2018), Es Yurek and Ozmutlu (2018), Kitjacharoenchai et al. (2019), Poikonen et al. (2019), Freitas and Penna (2020), Schermer et al. (2020), and Boccia et al. (2021).

Although the FSTSP, TSP-D, and RTSP-D are very similar, there are three key differences in the RTSP-D. First, the travel time is not a fixed parameter between two predetermined locations, which means that the time and distance are not directly proportional in real transportation networks (e.g., traffic congestions, restrictions in inner cities, or wind weather). Second, the truck may visit noncustomer locations that are not in the predetermined route, as this could be beneficial for operating the drone, see Fig. 2b. Third, the objective function should not only minimize the expected makespan but also minimize synchronization risk.

Considering the location decisions of launching or retrieving a drone, the RTSP-D is philosophically similar to the mothership (or carrier-vehicle) and drone routing problem (MDRP), which was introduced by Poikonen and Golden (2020a). In their model, a series of take-off and landing points were chosen from continuous Euclidean space. Different from the MDRP, our problem of interest is focused on inserting truck stops to serve the customers in congested areas, and the truck still reserves the option to visit these customers even in the worst case of traffic uncertainty. In this sense, we consider a truck-drone routing problem with satellites that the truck can stop at either customer location or noncustomer location. The truck-drone routing can be viewed as a two-echelon delivery network (Li et al., 2020), in which the locations of the satellites are not fixed. We intend to demonstrate that inserting truck stops offers several practical advantages (e.g., risk reduction) over the settings studied so far in the literature.

Considering the synchronization constraints, we extend the fixed rendezvous point to a rendezvous area where the drone can return to the truck within its maximum flight time. A common assumption in the literature is that both the truck and the drone are required to arrive at some fixed rendezvous points, where the vehicle arriving faster should wait for the other, such as Murray and Chu (2015), Murray and Raj (2020), Poikonen and Golden (2020b), Boccia et al. (2021), and Vu et al. (2021). This assumption implies that the truck should stop at the rendezvous point and the truck driver manually launches or retrieves the drone. When the drone handling is automatic and the drone can easily find the moving truck with GPS or other communication technologies, this assumption is no more required. Thus, we assume that the truck and the drone rendezvous at a moving point, where the drone can return to the truck within its maximum flight time. The obvious benefit of such a setting is the waiting time saving.

In the RTSP-D, the waiting time of the drone is calculated into its flight time. This is the same setting as the high synchronization requirements proposed by Dell'Amico et al. (2021). They considered two settings of the FSTSP: low synchronization requirements that the drone was allowed to land and wait at customer locations generally, and high synchronization requirements that the drone was allowed to wait only in flying mode specifically. Their experiments showed that the latter was harder to solve even for small instances (e.g., 10 customers), because drone battery endurance constraints were $n^3$ "big-M" constraints. Based on this, Dell'Amico et al. (2022) proposed three MILP formulations that decrease the "big-M" constraints to $n^2$. Different from the MILP formulations with the "big-M" constraints, we propose a second-order cone program model to calculate the drone flight time and its battery endurance constraints.

The RTSP-D is essentially a bi-objective problem. There are several variants with two objectives that have been studied in truck-drone routing problems. For example, Wang et al. (2020) addressed a variant of the bi-objective TSP-D to attain a compromise between operational cost and completion time. Salama and Srinivas (2020) proposed a variant of the bi-objective problem with one truck and multiple drones to minimize the total cost and the completion time. Shavarani et al. (2021) developed a bi-objective UAV-supported delivery model to minimize the total travel cost (distance) and lost demand simultaneously. Instead of the trade-off between the total cost and completion time or lost demand, our paper aims to obtain robust solutions that balance the risk and makespan.

In addition, the efficient frontier heuristic we proposed is similar to the hybrid method (combining the adaptive weighting and epsilon-constraint methods), which is commonly used to solve the bi-objective or multi-objective problems, for example, Demir et al. (2014), Eskandarpour et al. (2019), Wang et al. (2020), Zhen et al. (2020), and Abdullahi et al. (2021). Our solution method includes an implicit adaptive weighting method based on theoretical insights (Section 5.3.2). Our solution method also includes an epsilon-constraint method to generate a series of scalarized single-objective problems, in which the goal is to minimize expected makespan under the constraint of the synchronization risk. The difference is that our solution method allows solving the problem in *a priori* or *a posteriori* manner (Section 5.3.3).

### 2.2. Robust optimization of vehicle routing problem

In the literature on robust optimization of vehicle routing problem (VRP), a natural but cost-ineffective strategy is to use a deterministic model to generate reliable solutions in the worst case. The RTSP-D is therefore philosophically similar to the vehicle routing problem with drones (VRP-D) proposed by Wang et al. (2017), in which maximum savings were analytically determined on worst case. However, worst-case analysis hardly determines the trade-offs between the risk and makespan. The worst-case solutions may be deeply conservative in practice. Different from Wang et al. (2017), we use robust approaches to model the RTSP-D that balances the risk and makespan. Indeed, we present in Section 4.1 that the objective function of the RTSP-D favors minimized makespan while penalizing risk simultaneously.

Compared to worst-case analysis, an attractive option is to exploit additional knowledge of the uncertainty. There are three main uncertain phenomena considered in the VRP literature, including customer demands, service requests, and travel time. Among these uncertain phenomena, in the form of representative scenarios, probability distributions, or uncertainty set, routing optimization problems with uncertain travel time has been well studied (Laporte et al., 1992; Lee et al., 2012; Jaillet et al., 2016; Munari et al., 2019; Vural et al., 2019). Based on these researches, we adopt a commonly applied type, the simple ellipsoidal uncertainty set, to describe the travel time perturbation. We develop distributional robust models of the RTSP-D partially characterized through descriptive statistics such as moments, which are meaningful in practice and remain tractable (Ben-Tal et al., 2009; Bertsimas et al., 2011).

As a special variant of the VRP, the RTSP-D also involves aerial delivery with uncertainties, such as different weather conditions (Vural et al., 2019), uncertain navigation environments (Xiang et al., 2021), urban soundscapes perception (Torija et al., 2020), multiple-UAV coordination, and communication (Tortonesi et al., 2012). Considering the improved drone technology and the short flight time for each delivery, the drone speed is less uncertain than that of the truck in crowded urban areas. Thus, our study focuses on the uncertain truck travel time rather than the uncertain drone flight time.

## 3. Optimizing truck-drone schedule to minimize makespan

### 3.1. Traveling salesman problem with a drone

In TSP-D, a drone makes deliveries and returns to a truck that is moving on its route. Both vehicles start at a depot, denoted $v_0 \in V^0$, and deliver parcels to $n$ customer locations, denoted $v_1, \ldots, v_n \in V^n$. For each drone customer location $v_i \in V^d$, $i = 1, 2, \ldots, n$, the drone departs from the truck, flies to $v_i$, and then returns to the truck. Each customer location is assigned to the drone for contactless delivery, that is, $V^d = V^n$. For each truck customer location $v_i \in V^t$, $i = 1, 2, \ldots, n$, we require that the truck visits $v_i$. However, whether the truck visits a customer location also depends on the actual traffic conditions or restrictions, that is, $V^t \subseteq V^n$. If the truck arrives at $v_i$, the drone is launched and retrieved at $v_i$ to provide service to $v_i$. If the truck cannot arrive at $v_i$, two truck stops will be selected and inserted into the truck route to launch and retrieve the drone to provide service to $v_i$. When all customers have been visited, both vehicles return to the depot, denoted $v_{n+1} \in V^0$, $v_{n+1} = v_0$. Let $c^t(v_i, v_j)$ and $c^d(v_i, v_j)$ be the travel times from $v_i$ to $v_j$ for the truck and the drone, respectively. The TSP-D can be modeled in a graph $G(V,\ E)$, where $V = V^0 \cup V^n$, $E = \{(v_i, v_j) : \forall v_i \in V, v_j \in V\}$.

Suppose that each truck stop, at a customer location or noncustomer location, can be chosen from the Euclidean plane for launching or retrieving a drone. Times when a drone launches, lands, and flight durations are continuous variables, not constrained to discrete time intervals. The drone has a unit capacity and must return to the truck after visiting a customer.

Considering the synchronization requirements, the moving truck should appear at the rendezvous area where the drone can return to the truck within its maximum flight time. Therefore, the rendezvous point (as a truck stop in our paper) should be selected in the rendezvous area and the synchronization constraints are also needed. Because of limited battery capacity and waiting only in flying mode, in all cases, the drone must not be separated from the truck for more than $tmax$ consecutive time units. Otherwise, the failure of synchronization will happen.

The drone has an expected speed of $\alpha$ and the truck has an expected speed $\beta$. Note that defining network distances as vehicle travel times implies that the triangle inequality holds for both $c^t$ and $c^d$. To simplify notation, we assume that the service time and replenish time of both vehicles can be neglected.

The objective of TSP-D is to minimize the makespan of a truck-drone delivery system that a truck pairing with a drone starts at a depot, making parcel delivery for all customers until both vehicles are returned to the depot. The makespan is the sum of combined time and separated time for a truck and a drone in all delivery operations. The combined time in a delivery operation equals the travel time of a truck with a drone staying on from the current landing point (depot) to the next launch point. The separated time in a delivery operation is at least as long as the travel time of a truck from the current launch point to the next landing point, and at least as long as the flight time of a drone from the current launch point to a target customer, and then to the next landing point.

Given a solution of TSP-D that a truck route $\mathcal{R} = (r_0 = v_0, r_1, \ldots, r_k = v_{n+1})$ together with a drone route $\mathcal{D} = (d_0 = v_0, d_1, \ldots, d_m = v_{n+1})$, all customer nodes need to be contained in $(\mathcal{R},\ \mathcal{D})$. Note that the drone route $\mathcal{D}$ describes the full path of the drone, including all customer locations that are visited by both truck and drone.

We define binary decision variables $X = (\mathfrak{x}_0, \mathfrak{x}_1, \ldots, \mathfrak{x}_n, \mathfrak{x}_{n+1})$ as customer locations assignment, where $\mathfrak{x}_0 = \mathfrak{x}_{n+1} = 1$ is the depot that must be visited by the truck, and $\mathfrak{x}_i$ is equal to 1 if a customer location $v_i$ is visited by the truck and 0 otherwise. Thus, the truck customer node set $V^t = V\ (X) \backslash V^0$. Please note that every customer location is assigned to the drone for contactless delivery. In this case, it is assumed that all customer nodes are drone customer nodes. The reasons are as follows: (1) The drone can fly over the road or obstacle to serve truck-unreachable customers. (2) The drone has a much lower transportation cost. (3) The drone can be automatically launched and retrieved while the truck is moving. (4) The drone can provide logistics service in restricted regions to keep the truck driver and customers safe.

We define continuous decision variables $Z = (z_0^l,\ z_0^r,\ z_1^l,\ z_1^r, \ldots, z_n^l,\ z_n^r,\ z_{n+1}^l, z_{n+1}^r)$, where $z_i^l,\ z_i^r$ are truck stops for launching and retrieving a drone, respectively. For a predetermined sequence $\mathcal{R}^* = (v_0,\ v_1, \ldots, v_n, v_{\mathrm{n+1}})$, to determine the optimal assignment and where to launch and retrieve a drone, this subproblem can be formulated as follows:

$$\text{TSP-D-LR}\,(\mathcal{R}^*, X)$$

$$\min_X \sum_{i=0}^{n} (ctruck\,(i) + cdrone\,(i)), \tag{1}$$

subject to:

$$\frac{\|\ z_{i+1}^l - \ z_i^r\ \|}{\beta} \le ctruck\,(i)\,, \quad i = 0, 1, \ldots, n, \tag{2}$$

$$\frac{\|\ z_i^l - \ z_i^r\ \|}{\beta} \le cdrone\,(i)\,, \quad i = 0, 1, \ldots, n, \tag{3}$$

$$\|\ \mathcal{R}^*\,(i) - \ z_i^l\ \| \le oDist\,(i)\,, \quad i = 0, 1, \ldots, n, \tag{4}$$

$$\|\ \mathcal{R}^*\,(i) - \ z_i^r\ \| \le iDist\,(i)\,, \quad i = 0, 1, \ldots, n, \tag{5}$$

$$\frac{oDist\,(i) + iDist\,(i)}{\alpha} \le cdrone\,(i)\,, \quad i = 0, 1, \ldots, n, \tag{6}$$

$$cdrone\,(i) \le tmax, \quad i = 0, 1, \ldots, n, \tag{7}$$

$$z_i^l = \mathcal{R}^*\,(i)\,, \quad \forall \mathcal{R}^*\,(i) \in V\ \ (X)\,, \quad i = 0, 1, \ldots, n, n+1, \tag{8}$$

$$z_i^r = \mathcal{R}^*\,(i)\,, \quad \forall \mathcal{R}^*\,(i) \in V\ \ (X)\,, \quad i = 0, 1, \ldots, n, n+1. \tag{9}$$

The objective function (1) sets the makespan of a truck-drone delivery system as the sum of combined times and separated times. Constraint (2) ensures that combined time is at least as large as the truck travel time from $z_i^r$ to $z_{i+1}^l$. Constraint (3) ensures that separated time is at least as large as the truck travel time from $z_i^l$ to $z_i^r$. Together, constraints (4)–(7) ensure that separated time is at least as large as the sum of the drone's outbound flight duration and inbound flight duration, but not more than the drone's maximum flight time. Constraints (8) and (9) set the origin, truck customer locations, and destination. Table 1 provides a list of the decision variables and parameters of the TSP-D-LR($\mathcal{R}^*$, $X$).

The above second-order cone program defined by TSP-D-LR($\mathcal{R}^*$, $X$) is a subproblem of the TSP-D. Given a predetermined sequence $\mathcal{R}^*$ and an assignment $X$, TSP-D-LR($\mathcal{R}^*$, $X$) can be quickly solved to determine the optimal schedule of truck stops to lunch and retrieve a drone. We can easily verify that the optimal sequence of TSP may not be the optimal sequence of the TSP-D. To get the optimal solution of the TSP-D, it is useless to restrict the predetermined sequence to the optimal solution of the TSP. The TSP-D-LR($\mathcal{R}^*$, $X$) does not choose the sequence in which order the customer locations are visited. Similar to the carrier-drone routing problem proposed by Savuran and Karakaya (2016), they assumed that the carrier moved on a fixed route at a constant speed to predict drone landing points. Although the predetermined sequence $\mathcal{R}^*$ is already fixed in our subproblem, it is possible to adjust the truck route by inserting truck stops. We focus on the

Table 1
Decision variables and parameters of the TSP-D-LR($\mathcal{R}^*$, $X$)

| Name | Var./Fixed | Domain | Meaning |
|---|---|---|---|
| $n$ | Fixed | $\mathbb{N}$ | Number of customer locations |
| $\mathcal{R}^*(i)$ | Fixed | $\mathbb{R}^2$ | The $i$th customer location of a fixed sequence $\mathcal{R}^*$ |
| $\alpha$ | Fixed | $(1, \infty)$ | Expected speed of a drone |
| $tmax$ | Fixed | $(0, \infty)$ | Maximum flight time of a drone |
| $\beta$ | Fixed | 1 | Expected speed of a truck |
| $X$ | Var. | $\{0, 1\}$ | Every customer visited by the truck or not, $x_i \in \{0, 1\}$ |
| $ctruck(i)$ | Var. | $[0, \infty)$ | Combined time between $\mathcal{R}^*(i)$ landing and $\mathcal{R}^*(i+1)$ launch |
| $cdrone(i)$ | Var. | $[0, \infty)$ | Separated time between $\mathcal{R}^*(i)$ launch and $\mathcal{R}^*(i)$ landing |
| $z_i^l$ | Var. | $\mathbb{R}^2$ | The location where drone launches before flying to $\mathcal{R}^*(i)$ |
| $z_i^r$ | Var. | $\mathbb{R}^2$ | The location where drone lands after flying to $\mathcal{R}^*(i)$ |
| $oDist(i)$ | Var. | $[0, \infty)$ | Distance of the outbound flight from $z_i^l$ to $\mathcal{R}^*(i)$ |
| $iDist(i)$ | Var. | $[0, \infty)$ | Distance of the inbound flight from $\mathcal{R}^*(i)$ to $z_i^r$ |

Table 2
An example of the TSP-D-LR($\mathcal{R}^*$, $X$)

| Assignment ($X$) | Objective | Risk | Parameters |
|---|---|---|---|
| [1,1,1,1,1,0,0,0,1,1] | 235.31 | 50.1% | |
| [1,1,1,1,0,1,0,0,1,1] | 242.64 | 10.0% | $n = 8$ |
| [1,1,1,1,0,0,1,0,1,1] | 234.43 | 74.1% | $\mathcal{R}^*$=[66 84; 95 75; 65 27; 70 6; 43 10; 24 21; 39 39; 13 44; 26 60; 66 84] |
| [1,1,1,1,0,0,0,1,1,1] | 244.32 | 0.0% | $\alpha = 2$ |
| [1,1,1,0,1,1,0,0,1,1] | 224.08 | 54.3% | $tmax = 20$ |
| <u>[1,1,1,0,1,0,1,0,1,1]</u> | 219.48 | 87.9% | $\beta = 1$ |
| [1,1,1,0,1,0,0,1,1,1] | 229.04 | 67.3% | $\text{Risk} = Prob\left\{\| \frac{q_i cdrone(i)}{tmax} \|_\infty > 1\right\},\ i = 0, 1, \ldots, n$ |
| [1,1,1,0,0,1,1,0,1,1] | 234.15 | 76.6% | |
| [1,1,1,0,0,1,0,1,1,1] | 234.12 | 74.4% | |
| [1,1,1,0,0,0,1,1,1,1] | 234.39 | 74.6% | |

decisions of the assignment $X$ whether each customer location is visited by a truck or not. However, it is still an exponential number of assignment operations ($2^n$) even if the sequence is fixed in truck-drone schedules.

### 3.2. Risk sensitivity of optimal schedules

Under traffic uncertainty, travel times of both vehicles in the TSP-D-LR($\mathcal{R}^*$, $X$) may be subject to variables. A naive approach to deal with uncertain travel time is to use expected speed values in the optimization. The problem with this approach is that optimal schedules become infeasible under small travel time perturbations. Now assume that *a priori* knowledge of the travel time perturbations (percentage) is given by a column vector $Q = (q_0, \ldots, q_n)^T,\ \forall q_i \in [0.95, 1.05],\ i = 0, 1, \ldots, n$, in which the travel-time perturbations surround its expected value with a uniform distribution. As an example (shown in Fig. 2), the optimal solution is risk-sensitive to travel-time perturbations even at a level of 5%. This example is described in Table 2. $\mathcal{R}^*(0)$, $\mathcal{R}^*(9)$ is the origin and destination, respectively. The truck must visit five customers, in which $\mathcal{R}^*(1)$, $\mathcal{R}^*(2)$, and $\mathcal{R}^*(8)$ are assigned to the truck already. With these two additional restrictions, this example can be solved by complete enumeration quickly. Solving large-sized problems that full enumeration is infeasible will be discussed in Section 5.

The formulation of the overall risk is provided in Table 2. The synchronization failure for each flight is described by the ratio of separated time to the maximum flight time of the drone. If the ratio is more than 1, the solution is infeasible that synchronization failure occurs. The risk of each solution is described by the probability of synchronization failure occurring, which is evaluated by Monte Carlo analysis in this paper (see Supplementary materials A).

The nominal optimal assignment underlined in Table 2 has a large risk. This implies that, even though the nominal makespan is minimized, there is a lot of uncertainty in the actual makespan realization. Compared to the nominal optimal assignment, some suboptimal assignments are associated with significantly lower risk while only entailing a small increase in makespan. This illustrates the need to construct schedules that balance the risk and makespan.

## 4. Optimizing truck-drone schedule to balance risk and makespan

### 4.1. Robust traveling salesman problem with drone

To balance the risk and makespan, we reformulate the TSP-D as its robust counterpart. Note that the truck is always on the path to the next location when the drone is visiting a customer. Let $\tilde{\beta}_i$ be the uncertain truck speed when traveling to $v_i$, for example, the shortest path of $(v_j, v_i)$ is crowded and $\tilde{\beta}_i$ is smaller than expected speed, thus the travel time becomes longer. To determine the optimal assignment and where to launch and retrieve a drone, we define $U$ as the uncertainty region in which the true truck speeds lie, and the robust counterpart of this subproblem can be formulated as follows:

TSP-D-LR$(\mathcal{R}^*, X)_{robust}$

$$\min_X \left\{ \min_{\tilde{\beta} \in U} \sum_{i=0}^{n} (ctruck\,(i) + cdrone\,(i)) \right\}, \tag{10}$$

subject to:

$$\frac{\| z_{i+1}^l - z_i^r \|}{\tilde{\beta}_{i+1}} \leq ctruck\,(i), \quad i = 0, 1, \ldots, n, \tag{11}$$

$$\frac{\| z_i^l - z_i^r \|}{\tilde{\beta}_{i+1}} \leq cdrone\,(i), \quad i = 0, 1, \ldots, n, \tag{12}$$

(4)–(9).

The objective of TSP-D-LR$(\mathcal{R}^*, X)_{robust}$ is to minimize the makespan with intractable constraints (11) and (12) to limit the synchronization risk, using the worst possible truck speeds among parameters $\tilde{\beta}$ in the uncertainty region $U$. Imagine a natural perturbation mechanism that produces random perturbation of parameter $\tilde{\beta}_i$ when a truck visits $v_i$, but the additional travel time is not more than a known bound $c(i) \geq 0$, which can be considered the maximal extra time to take a detour. If the truck does not visit $v_i$, it inevitably takes another path to visit the next customer, which can be considered to pass by a noncustomer $v_i'$ and generate an additional travel time bounded by $c'(i) \geq 0$. We adopt a commonly used type of $U$ in the robust optimization literature, that is, the simple ellipsoidal uncertainty set. This uncertainty set allows for a flexible approximation of various uncertain parameters, and crucially, it results in tractable robust counterpart typically (Ben-Tal et al., 2009). For describing truck speed perturbation, the simple ellipsoidal uncertainty set is defined as

$$Z = \left\{ \zeta \in \mathbb{R}^{n+1} : \ \|\zeta\|_\infty \leq 1, \ \|\zeta\|_2 \leq \Omega \right\}, \tag{13}$$

where $\zeta$ is a random variable and $\Omega$ is the radius centered at the origin. According to perturbation set $Z$, constrains (11) and (12) can be rewritten as

$$\frac{\| z_{i+1}^l - z_i^r \|}{\beta} + x_i \zeta_{i+1} c' (i+1) \neg x_{i+1} + x_i \zeta_{i+1} c (i+1) x_{i+1} \leq ctruck (i), \tag{14}$$

$$\frac{\| z_i^l - z_i^r \|}{\beta} + \neg x_i \zeta_{i+1} c' (i+1) \neg x_{i+1} + \neg x_i \zeta_{i+1} c (i+1) x_{i+1} \leq cdrone (i). \tag{15}$$

Based on Ben-Tal et al. (2009), we can recast the TSP-D-LR$(\mathcal{R}^*, X)_{robust}$ as a tractable model as follows (see Supplementary materials B):

RTSP-D-LR $(\mathcal{R}^*, X)$

$$\min_X \left\{ \sum_{i=0}^{n} (ctruck (i) + cdrone (i)) + \Omega \sqrt{\sum_{i=0}^{n} (c' (i+1) \neg x_{i+1} + c (i+1) x_{i+1})^2} \right\}, \tag{16}$$

subject to:

(2)–(9).

The objectives of RTSP-D-LR$(\mathcal{R}^*, X)$ are to minimize expected makespan and synchronization risk simultaneously. Denote that $SD = \sqrt{\sum_{i=0}^{n} (c'(i+1)\neg x_{i+1} + c(i+1)x_{i+1})^2}$ is the standard deviation of additional travel time. Note that the parameter $\Omega$ is defined as the radius centered at the origin in a simple ellipsoidal uncertainty set. If $\Omega = 0$, the function (16) will degenerate into a single objective to minimize the expected makespan. If $\Omega > 0$, the function (16) will minimize the expected makespan and the synchronization risk simultaneously. The parameter setting of $\Omega$ involves the risk aversion of a decision-maker, whose preference is nonlinear in probabilities (Rabin and Thaler, 2001). Thus, the $\Omega$ is not only a changing weight of the objective, but also a risk penalty coefficient associated with the uncertainty in travel time.

To describe the risk aversion of a decision-maker, we can properly set $\Omega$ by introducing probabilistic guarantees. This specific scalarizing approach is commonly used in robust optimization, for

Table 3
Robust probabilistic guarantees

| | | | | | |
|---|---|---|---|---|---|
| $\Omega$ | 2.15 | 2.45 | 3.03 | 3.26 | 3.72 |
| $\epsilon$ | 10% | 5% | 1% | 0.5% | 0.1% |

example, Ben-Tal et al. (2009), Rooderkerk and van Heerde (2016), Bertsimas and Youssef (2020), and Hong et al. (2021). We construct a simple ellipsoidal uncertainty set by assuming a multivariate normal distribution for the random variables $\zeta$, that is, $\zeta \sim \mathcal{N}(0, 1)$. Consequently, the total additional travel time associated with an assignment and a fixed time bound follows a normal distribution,for example, $\sum_{i=0}^{n}( \zeta_{i+1}(c'(i+1)\neg x_{i+1} + c(i+1)x_{i+1}) ) \sim \mathcal{N}(0,\ SD^2)$. Based on Ben-Tal et al. (2009), for every $\Omega \geq 0$, it holds that

$$Prob\left\{\zeta : \sum_{i=0}^{n} \left( \zeta_{i+1}\left(c'(i+1)\neg x_{i+1} + c(i+1)x_{i+1}\right) \right) > \Omega * SD \right\} \leq exp\left\{-\Omega^2/2\right\}. \tag{17}$$

The probability of (17), in which the true additional travel time is larger than robust additional travel time, is not more than the specified confidence level. Given a confidence level $\epsilon = exp\{-\Omega^2/2\}$, we get $\Omega = \sqrt{2\,ln(1/\epsilon)}$ that provides robust probabilistic guarantees (shown in Table 3). Thus, the objective function (16) can balance the risk and makespan by setting a confidence level.

The next section illustrates the differences between the TSP-D-LR($\mathcal{R}^*, X$) and RTSP-D-LR($\mathcal{R}^*, X$). For a detailed analysis, we now denote the objective values of the TSP-D-LR($\mathcal{R}^*, X$) and RTSP-D-LR($\mathcal{R}^*, X$) as nominal makespan ($NM$) and robust makespan ($RM$), respectively.

$$NM = \sum_{i=0}^{n}(ctruck(i) + cdrone(i)), \tag{18}$$

$$RM = \sum_{i=0}^{n}(ctruck(i) + cdrone(i)) + \Omega \cdot SD. \tag{19}$$

### 4.2. Robust optimization example

Taking the example presented earlier, Table 4 shows that the nominal optimal assignment (marked with a straight line) is not the robust optimal assignment (marked with a straight line) at three confidence levels commonly used. Since customer location $\mathcal{R}^*(6)$ is in the traffic congestion area (see Fig. 2) and $\mathcal{R}^*(5)$ is also severely affected, the assignment $X$ including any one or two of these locations provides a high $SD$ for the truck-drone schedule (e.g., [1,1,1,0,0,1,1,0,1,1] with $SD = 20.45$). In addition, the robust optimal assignment to each of these three confidence levels is the same. This is due to the lower $SD$ associated with this solution. Lower $SD$ means that the uncertainty in travel time can be reduced substantially by choosing an assignment that only slightly increases the expected makespan. The $NM$ (229.04) of the robust optimal assignment only increases 4.4%, but its $RM$ (244.05, $\epsilon = 1\%$) reduces 11.4% than that of the nominal optimal solution. More importantly, the risk of the robust optimal assignment (67.3%) is much smaller than that of the nominal

Table 4
Nominal versus robust makespan of the feasible assignments

| | | | *RM* | | | |
|---|---|---|---|---|---|---|
| Assignment ($X$) | $NM$ | Risk | $\epsilon = 5\%$ | $\epsilon = 1\%$ | $\epsilon = 0.5\%$ | $SD$ |
| [1,1,1,1,1,0,0,0,1,1] | 235.31 | 50.1% | 243.15 | 245.03 | 245.73 | 3.20 |
| [1,1,1,1,0,1,0,0,1,1] | 242.64 | 10.0% | 265.42 | 270.88 | 272.93 | 9.31 |
| [1,1,1,1,0,0,1,0,1,1] | 234.43 | 74.1% | 281.10 | 292.29 | 296.49 | 19.07 |
| [1,1,1,1,0,0,0,1,1,1] | 244.32 | 0.0% | 260.94 | 264.93 | 266.42 | 6.79 |
| [1,1,1,0,1,1,0,0,1,1] | 224.08 | 54.3% | 243.82 | 248.55 | 250.33 | 8.06 |
| [1,1,1,0,1,0,1,0,1,1] | 219.48 | 87.9% | 264.74 | 275.60 | 279.67 | 18.49 |
| [1,1,1,0,1,0,0,1,1,1] | 229.04 | 67.3% | 241.15 | 244.05 | 245.14 | 4.95 |
| [1,1,1,0,0,1,1,0,1,1] | 234.15 | 76.6% | 284.21 | 296.21 | 300.72 | 20.45 |
| [1,1,1,0,0,1,0,1,1,1] | 234.12 | 74.4% | 258.70 | 264.59 | 266.81 | 10.04 |
| [1,1,1,0,0,0,1,1,1,1] | 234.39 | 74.6% | 281.97 | 293.38 | 297.66 | 19.44 |

optimal assignment (87.9%). This instance shows the essentials of robust optimization: balancing a relatively small increase in expected makespan against a substantial reduction in risk surrounding the realization of a truck-drone schedule.

In this regard, we would like to note another approach based on sensitivity analysis to deal with parameter perturbations after the problem has been optimized, for example, Dorling et al. (2017) and Wang and Sheu (2019). These studies still essentially optimize the deterministic model without considering any uncertainty during optimization. Our robust approach is fundamentally different from this approach because it balances the risk and makespan during the optimization. Hence, in our study, the risk and makespan co-determine the compositions of the optimal solution.

## 5. Solution method for optimizing truck-drone schedule

### 5.1. Motivation for solution method

We select *a posteriori* efficient frontier heuristic to solve large instances of the RTSP-D. The subproblem RTSP-D-LR($\mathcal{R}^*$, $X$) is essentially a bi-objective second-order cone program, which minimizes expected makespan and synchronization risk simultaneously. If the value of risk penalty coefficient $\Omega$ is fixed, it will lead to a single-objective optimization problem to minimize the makespan. And then it may be quickly solved to determine the optimal assignment and where to launch and retrieve a drone associated with a predetermined sequence $\mathcal{R}^*$.

For the predetermined sequence $\mathcal{R}^*$, as discussed in Section 3.1, it does not need to be an optimal solution of the TSP. This justifies that using any TSP heuristic to construct the $\mathcal{R}^*$ during our experiments. More importantly, with a state-of-the-art TSP solver such as Concorde, the instances of up to several hundred nodes can be solved quickly (Agatz et al., 2018). Therefore, we suppose that the $\mathcal{R}^*$ is given in our subproblem. This also indicates that we adopt a route-first cluster-second method to solve the RTSP-D.

For every assignment, it will generate a truck route $\mathcal{R}$ and a relative drone route $\mathcal{D}$, which combine a solution ($\mathcal{R}$, $\mathcal{D}$) of the RTSP-D. For small instances, such as the example discussed in

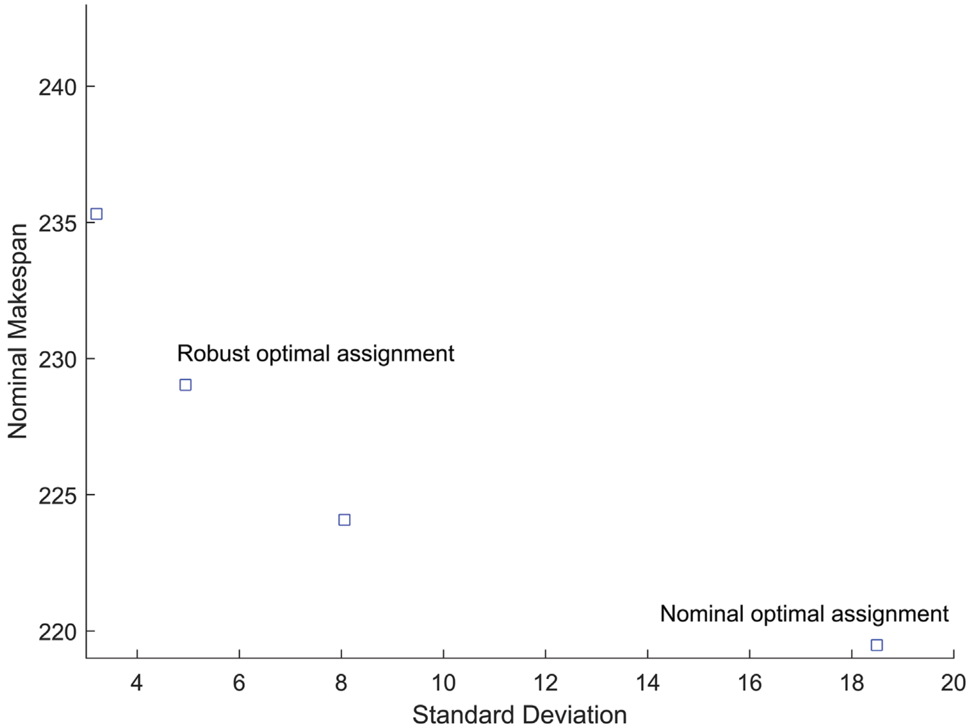


Fig. 3. The efficient frontier for the example in Table 4.

Section 3.2, we can solve this problem using full enumeration. In practice, large-scale problems preclude the use of full enumeration. Therefore, we resort to heuristics to solve these problems.

For a given value of Ω, heuristics that aim to find the optimal solution of the RTSP-D are referred to as *a priori* methods. If we choose a larger Ω (e.g., $\Omega = 3.72$ in Table 3), the robust optimal solution in Table 4 will change. Hence, a decision-maker may find that it is very difficult to announce the amount of risk aversion (setting Ω value) before seeing the trade-offs in certain real-life scenarios, such as in contactless delivery. Thus, we propose an efficient frontier heuristic to construct approximations of the whole Pareto frontier, which belongs to *a posteriori* methods. To reduce computational effort, the heuristic also allows for selecting a (near-)optimal solution based on the priori information.

### 5.2. *Efficient frontier example*

Taking the example discussed in Section 4.2, Fig. 3 shows the efficient frontier that provides a visual tool for a decision-maker to adapt his/her preference. Note that the efficient frontier excludes some assignments (e.g., [1,1,1,0,0,1,1,0,1,1]) that do not satisfy the efficiency principle: bring a bigger expected makespan than nominal optimal assignment, meanwhile leading to a larger *SD*. The visualizations of the efficient frontier, in which the *SD* associated with every assignment is declining gradually, support a decision-maker to find other assignments that provide better trade-offs between the risk and makespan. This also indicates that a decision-maker can visually adjust the value of Ω, considering the quantities of uncertainties that refer to the willingness.

In Fig. 3, there is no efficient assignment with an *SD* exceeding that of the nominal optimal assignment. As an immediate corollary from the definition of the efficient frontier, the nominal optimal assignment sets an upper bound of the *SD* for all efficient assignments. By systematically lowering this upper bound, we can quickly construct a subset of efficient assignments to approximate the efficient frontier. In the next section, we propose an efficient frontier heuristic based on problem features of the RTSP-D.

### 5.3. Efficient frontier heuristic method

#### 5.3.1. Minimizing makespan while limiting standard deviation

Limiting the *SD* introduces a nonlinear restriction to the TSP-D-LR($\mathcal{R}^*$, $X$) formulated in Section 3.1. The expression of the bounded *SD* is formulated as follows:

$$\sqrt{\sum_{i=0}^{n} \left(c'(i+1)\neg x_{i+1} + c(i+1)\,x_{i+1}\right)^2} + h = \overline{SD}, \tag{20}$$

where $\overline{SD}$ is the upper bound of the actual *SD*, and a slack variable $h \geq 0$ is the difference between the imposed upper bound and the actual *SD*. With this restriction, we denote the reformulated version as

$$\text{TSP-D-LR}(\mathcal{R}^*, X)_{limited}$$

$$\min_{X} \left\{ \sum_{i=0}^{n} (ctruck(i) + cdrone(i)) - mh \right\}, \tag{21}$$

subject to:

(2)–(9) and (20).

The objective function (21) consists of two parts: the left part is $NM$, the right part is that slack variable $h$ times a small enough positive value $m$. By adding the right part, we enforce the optimal assignment is efficient while obeying the imposed upper bound. That is, first finding the assignment to minimize $NM$, and if some assignments with the same minimized $NM$, the TSP-D-LR($\mathcal{R}^*$, $X$)$_{limited}$ will select the assignment with the minimum *SD*.

Suppose that $\delta > 0$ is the minimum makespan difference between any two efficient assignments. Then $m = \delta/\overline{SD}$ is a theoretically optimal value that we select, that is, $0 \leq mh \leq (\delta/\overline{SD})\overline{SD} = \delta$. For practical reasons, we choose a small enough value of $\delta = 0.0001$ in the empirical application (Rooderkerk and van Heerde, 2016). This value is sufficiently small in our experiment that it does not affect the solutions. Consequently, every possible decrease in expected makespan will always be preferred to a smaller *SD*.

*5.3.2. Theoretical insights*

Our solution method is predicated on the following observations.

**Observation 1.** Let $obj(TSP)$, $obj(TSP\text{-}D)$, and $obj(RTSP\text{-}D)$ denote the optimal objective value for the TSP, TSP-D, and RTSP-D, respectively, for the same target locations in graph $G(V, E)$. Then $obj(TSP)/(\alpha + 1) \leq obj(TSP\text{-}D) \leq obj(RTSP\text{-}D) \leq obj(TSP)$.

This is a generalization that an optimal solution of the TSP is a $(\alpha + 1)$ approximation to the TSP-D proposed by Agatz et al. (2018). The lower bound in Observation 1 can be demonstrated by noting that the truck-drone must travel the same minimum distance of the Euclidean TSP among the locations in $V$ with combined speed $(\alpha + 1)$ in the best case. The upper bound in Observation 1 is valid because the truck may travel to each target location in $V$ in the worst case. The *obj(RTSP-D)* is not less than the *obj(TSP-D)* because of the objective function (16). Based on Observation 1, we can get the upper and lower bound of objective values of the RTSP-D.

**Observation 2.** Let $X1$ be a sub-assignment of $X2$ for a given sequence $\mathcal{R}^*$ of target locations in graph $G(V, E)$, and $NM(\mathcal{R}^*, X1)$, $NM(\mathcal{R}^*, X2)$ is the nominal optimal makespan of *TSP-D-LR*$(\mathcal{R}^*, X1)$ *and TSP-D-LR*$(\mathcal{R}^*, X2)$, respectively. Then $NM(\mathcal{R}^*, X1) \leq NM(\mathcal{R}^*, X2)$.

Observation 2 can be demonstrated that any feasible solution of the TSP-D-LR$(\mathcal{R}^*, X1)$ must be a feasible solution to the TSP-D-LR$(\mathcal{R}^*, X2)$ with loose constraints (8) and (9). Thus, the nominal optimal makespan of the TSP-D-LR$(\mathcal{R}^*, X1)$ is at most as the TSP-D-LR$(\mathcal{R}^*, X2)$.

If $X1$ is a sub-assignment of $X2$ and $X1$ is not promising (i.e., $NM(\mathcal{R}^*, X1)$ is large), then $X2$ will not be given a high priority in our local search because that $NM(\mathcal{R}^*, X2)$ is at least as large as $NM(\mathcal{R}^*, X1)$. Thus, our local search is step by step for $|V^t| = 1, 2, \ldots, n-1$. If we set $|V^t| = 0$, then $X^0 = [1, 0,\ldots,0,1]$ is a sub-assignment for all assignments. As an immediate corollary of Observation 2, $NM(\mathcal{R}^*, X^0)$ is the nominal optimal makespan. Since there is no efficient assignment with an *SD* that exceeds the *SD* of the nominal optimal assignment, $X^0$ sets an upper bound of the *SD* for all efficient assignments. With this upper bound of the *SD*, we develop an efficient frontier heuristic method to solve the RTSP-D.

*5.3.3. Efficient frontier heuristic*

Figure 4 shows the steps of the efficient frontier heuristic method. For $|V^t| = 1, 2, \ldots, n-1$, the heuristic will terminate until the maximum $|V^t|$ is reached or early terminate at Step 7. The Early Termination Criterion is based on Observations 1 and 2. If we only want to find a (near-)optimal solution rather than the whole Pareto frontier, the heuristic can be early terminated.

**Early Termination Criterion.** If the most recently obtained $NM_{recent}$ of the efficient assignment is not less than the $RM_{best}$ found so far, it is certainly no better robust solution will be found.

Figure 5 illustrates the heuristic to construct the efficient frontier for the example discussed in Section 4.2. Given $|V^t| = 5$, $\Omega = 3.03$, $s = 5$, the *SD* upper bound is lowered to zero step by step. At each step, the TSP-D-LR$(\mathcal{R}^*, X)_{limited}$ is solved. Finally, the efficient frontier heuristic terminates after five sub-iterations. The efficient assignments found by the heuristic (shown as red dots in Fig. 5) are the same as those found by full enumeration. The *NM* values corresponding to these assignments are shown in Fig. 5a. The *RM* values corresponding to these assignments are shown in Fig. 5b.

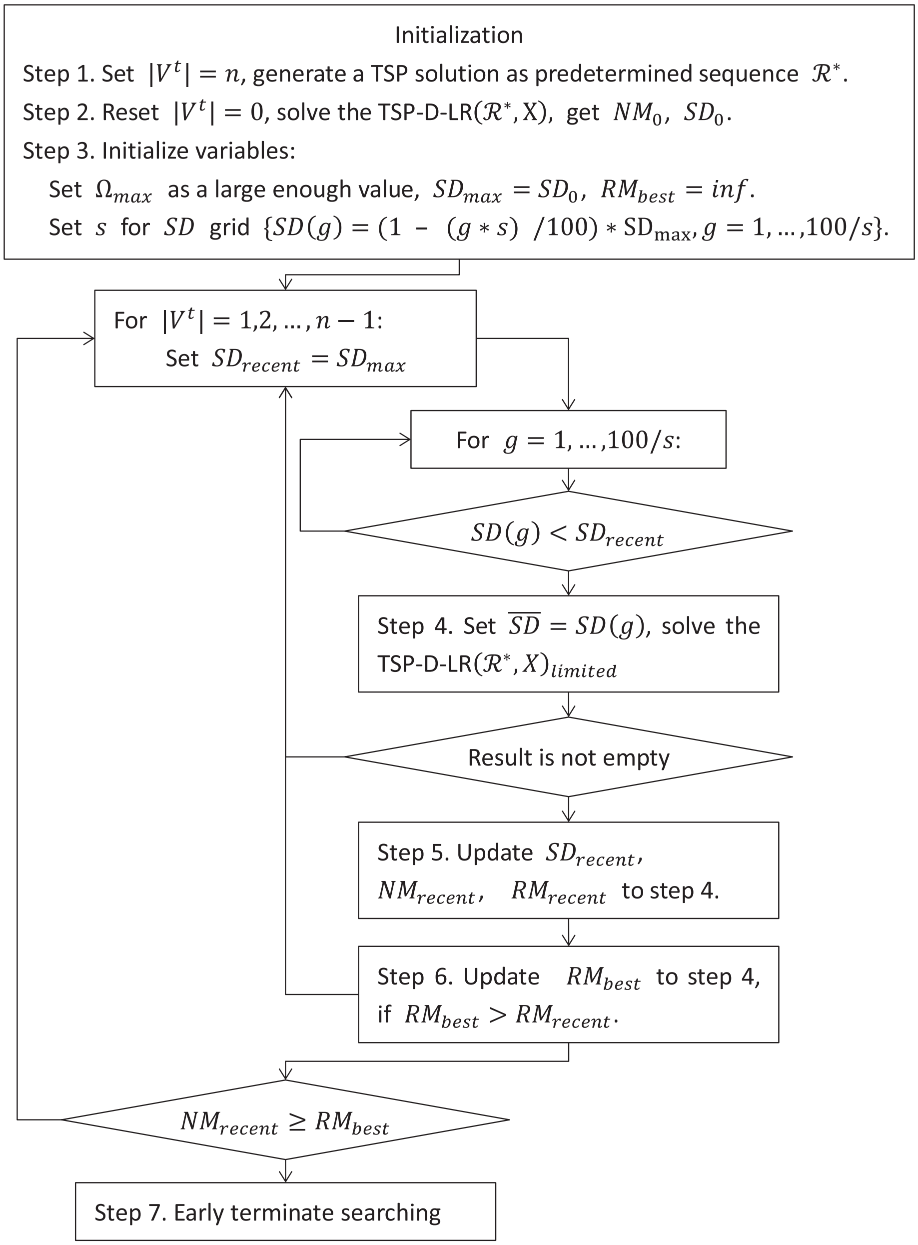


Fig. 4. Flowchart of efficient frontier heuristic.

One advantage of the heuristic is that it allows choosing a robust solution for any given value of $\Omega \leq \Omega_{\max}$ after it is terminated. This is convenient for a decision-maker to decide how much risk aversion to take, based on the risk-makespan trade-offs on hand. Another advantage is that it can obtain a subPareto frontier by early termination, which depends on the priori information

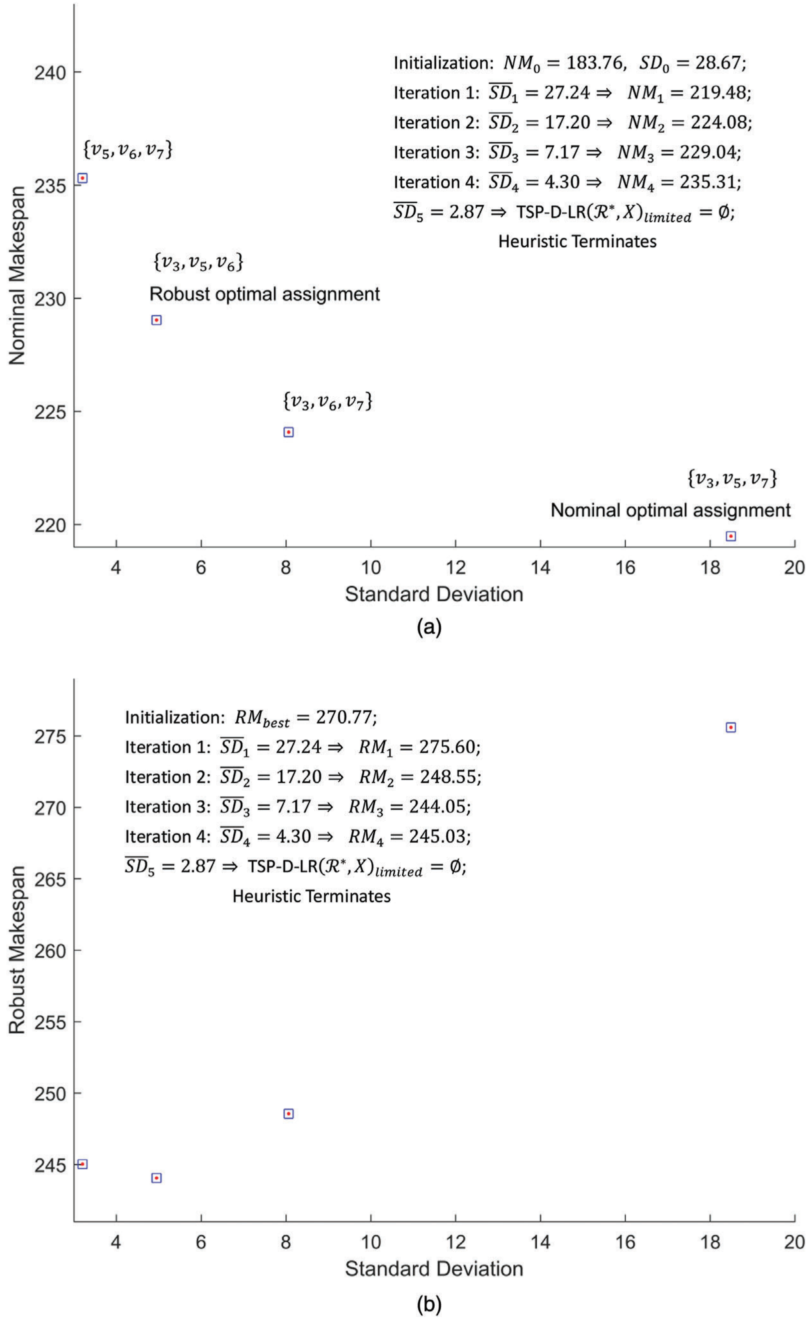


Fig. 5. An illustration of the $NM$ (a) and $RM$ (b) obtained by the heuristic. *Note*: Set $|V^t| = 5$, $\Omega = 3.03$, $s = 5$. The blue squares and red dots are found by full enumeration and the heuristic, respectively. The nodes between the brackets in panel (a) are customer nodes not visited by the truck in corresponding efficient assignments.

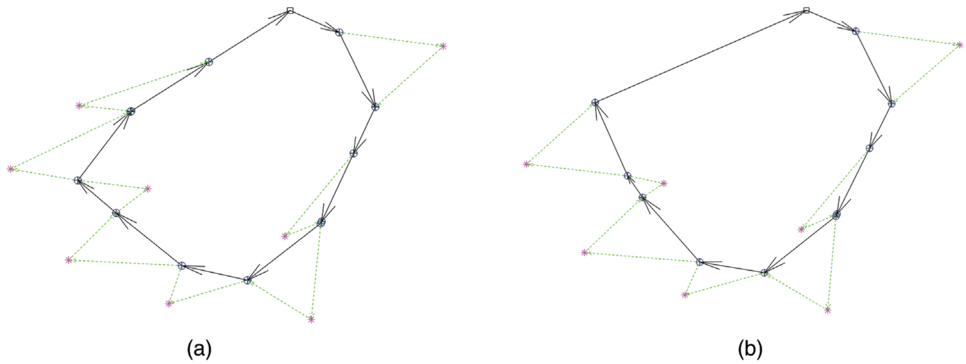


Fig. 6. Nominal optimal solution (a) versus robust optimal solution (b) for extended example. *Note*: Set $\Omega = 3.03$. The paths of the truck and drone are indicated by solid and dashed lines, respectively.

from a decision-maker. When $\Omega_{\max}$ is completely unknown, by setting a large enough number, the heuristic can approximate the whole Pareto Frontier, in which every robust solution presents a balance between the risk and makespan. But approximations of the whole Pareto Frontier will take more time to run the heuristic. To reduce computational effort, by setting an expected value of $\Omega_{\max}$, the heuristic can provide a subset of the whole Pareto Frontier. If a robust solution of good enough is found in the subset, the heuristic can be early terminated.

## 6. Minimize the robust makespan for extended example

To explore the potential benefits of risk-makespan trade-offs, we remove the additional restrictions of the example shown in Table 2. For this extended example, the number of assignments ($X$) has piped up to $2^8$ for every given sequence. The dimensionality of this example is not excessively large solved by full enumeration, which allows us to verify the performance of the efficient frontier heuristic. All optimization problems, including the efficient frontier heuristic and the full enumeration, are coded in Matlab R2016b and run on a laptop PC equipped with AMD A12-9700P 2.5 GHz processor and 16 GB memory capacity. The software code is available on request.

### *6.1. Nominal versus robust assignment optimization*

To assess the potential of the robust approach, we first solve the extended example using full enumeration. For every predetermined sequence, we randomly generate the additional travel time parameters based on the normal distribution and optimize the assignments separately. We perform the robust optimization problems at three confidence levels: 5%, 1%, 0.5%, corresponding to the values of $\Omega$: 2.45, 3.03, 3.26, respectively. The average computation time per problem is in seconds: 449.03, 451.67, and 456.14, respectively.

Figure 6 shows the nominal and robust optimal solutions for the extended example. Both solutions demonstrate that inserting truck stops is an attractive alternative to balance the risk and

Table 5
Nominal versus robust optimal assignment

| | Constrained example | | Extended example | |
|---|---|---|---|---|
| | *NM* | *RM* | *NM* | *RM* |
| $\Omega = 2.45$ | | | | |
| Nominal optimal assignment | 219.48 | 264.74 | 183.76 | 254.29 |
| Robust optimal assignment | 229.04(4.4%) | 241.15(−8.9%) | 187.34(1.9%) | 217.23(−14.6%) |
| $\Omega = 3.03$ | | | | |
| Nominal optimal assignment | 219.48s | 275.60 | 183.76 | 270.77 |
| Robust optimal assignment | 229.04(4.4%) | 244.05(−11.4%) | 187.34(1.9%) | 224.40(−17.1%) |
| $\Omega = 3.26$ | | | | |
| Nominal optimal assignment | 219.48 | 279.67 | 183.76 | 277.23 |
| Robust optimal assignment | 229.04(4.4%) | 245.14(−12.3%) | 187.34(1.9%) | 227.09(−18.1%) |

*Note*. The percentage in the brackets is the relative change of objective values.

makespan. When moving from the nominal optimal solution to a robust one, a natural experience is that an increase in nominal makespan comes along with a decrease in robust makespan. The problem is how large do they change? The answer is shown in Table 5. These comparisons are between the nominal optimal assignment (100%) and the robust optimal assignment by the relative changes of objective values.

Table 5 shows that balancing a large gain (risk reduction) only needs a small price (makespan increase). For the constrained example and the extended example, the relative increase in $NM$ is quite small, while the relative decrease in $RM$ is significant. In addition, when moving $\Omega$ from 2.45 to 3.26, the relative changes in RM become larger for the extended example. Based on these results, we compare them to the results obtained by the efficient frontier heuristic in the next section.

### *6.2. Application of the efficient frontier heuristic*

In an automated fashion, the extended example is solved by the efficient frontier heuristic sequentially for different values of $\Omega$ and $s$. The performance of the efficient frontier heuristic is listed in Table 6.

Table 6 shows that the efficient frontier heuristic performs as well as full enumeration. First, the heuristic obtains the robust optimal assignment for almost all sequences (between 26 and 29 out of 29). As expected, when the values of $\Omega$ and $s$ are moved from 2.45 and 1% to 3.26 and 5%, respectively, the quantities of the robust optimal assignment found by the heuristic are slightly decreased. Even for the worst case ($\Omega = 3.26$ and $s = 5\%$), the average percentage of resulting assignments is very near to the optimum (100.05% of the optimum). Second, the average number of iterations in the heuristic is significantly smaller than the maximum iterations ($100/s$). This proves that the heuristic can be early terminated, which depends on the level of risk aversion and interaction information in real-time by using the Early Termination Criterion. Third, the running time of the heuristic is very short, even at the worst case ($\Omega = 3.26$ and $s = 1\%$). A smaller step size will result in a more complete search and more likely find the robust optimal assignment. However, this also results in more time consumption, for example, the average running time from 9.97 seconds

Table 6
Performance of the efficient frontier heuristic

| | Step size ($s$) | | |
|---|---|---|---|
| | 1.0% | 2.5% | 5.0% |
| Number of sequences with the robust optimal assignment (of 29) | | | |
| $\Omega = 2.45$ | 29 | 29 | 28 |
| $\Omega = 3.03$ | 29 | 28 | 27 |
| $\Omega = 3.26$ | 27 | 27 | 26 |
| Average percentage of the optimal robust makespan | | | |
| $\Omega = 2.45$ | 100.00(100.00) | 100.00(100.00) | 100.04(101.04) |
| $\Omega = 3.03$ | 100.00(100.00) | 100.05(101.58) | 100.09(101.58) |
| $\Omega = 3.26$ | 100.01(100.23) | 100.02(100.34) | 100.05(100.81) |
| Average number of iterations in the heuristic | | | |
| $\Omega = 2.45$ | 9.07(17) | 8.90(15) | 8.45(11) |
| $\Omega = 3.03$ | 10.48(17) | 10.10(15) | 9.86(13) |
| $\Omega = 3.26$ | 11.17(18) | 11.07(16) | 10.41(14) |
| Average running time (in seconds) | | | |
| $\Omega = 2.45$ | 9.78(15.22) | 9.61(13.19) | 9.29(12.25) |
| $\Omega = 3.03$ | 12.10(18.72) | 10.73(14.78) | 9.90(13.25) |
| $\Omega = 3.26$ | 12.80(19.23) | 11.35(18.97) | 9.97(16.57) |
| Average reduction in running time compared to full enumeration (in percent) | | | |
| $\Omega = 2.45$ | 97.82(96.57) | 97.86(97.03) | 97.93(97.24) |
| $\Omega = 3.03$ | 97.32(95.80) | 97.63(96.68) | 97.81(97.02) |
| $\Omega = 3.26$ | 97.19(95.74) | 97.51(95.80) | 97.81(96.33) |

*Note*. The values in brackets are the worst-case values, that is, the maximum percentage of the optimal robust makespan, the maximum number of iterations, the maximum running time, and the minimum reduction in running time.

($s = 5\%$) is up to12.80 seconds ($s = 1\%$) for $\Omega = 3.26$. More importantly, compared with the full enumeration, the average running time of the heuristic algorithm is reduced by 95.74% or more in all cases. Finally, these comparisons indicate the great potential of the heuristic to solve large instances when the full enumeration is infeasible.

## 7. Conclusions and directions for future research

In this paper, we systemically investigate the robust schedule of a truck and a drone in the contactless delivery setting. Our first study presents that the nominal assignment of minimum makespan is sensitive to uncertain travel time, and a robust assignment is required to be constructed to reduce the risk of synchronization failure. Therefore, this paper proposes a robust model of RTSP-D to minimize the expected makespan and synchronization risk simultaneously.

We extend route-first cluster-second methods to solve the robust model of RTSP-D, in which the assignment decisions and location decisions are extracted from three-level truck-drone routing problem. At the first level, a TSP solver is used to generate a sequence (not necessarily optimal) as the predetermined sequence of RTSP-D. At the second level, we design an efficient frontier heuristic to find (near-)optimal assignments of RTSP-D. The heuristic combines implicit adaptive weighting and epsilon-constraint methods to generate a series of scalarized single-objective problems, in which the goal is to minimize expected makespan and the synchronization risk is a constraint. The heuristic constructs approximations of Pareto frontier, which allows the problem to be solved in *a priori* or *a posteriori* manner. At the third level, for the given sequence and assignment, we can quickly solve the subproblem (second-order cone program) to determine the locations to launch and retrieve a drone.

The experiment results show that the efficient frontier heuristic performs as well as full enumeration. Moreover, the efficient frontier heuristic is much faster than full enumeration. The results also show that inserting truck stops is an attractive alternative to balance the risk and makespan. And the robust (near-)optimal assignments offer a relatively small increase in expected makespan against a large reduction in synchronization risk. These results provide managerial insights for e-commerce enterprises to fully integrate truck-drone delivery networks under uncertainty. For example, for the prescheduled delivery of fresh foods or medicines, whose delivery is time-sensitive under uncertain traffic conditions in big cities, our robust approach can provide a solution to ensure that an airborne drone compensates for the travel time perturbations of a ground-based truck cost-effectively.

Finally, it would be interesting to apply the robust approach we developed to other routing problems with more drones. One example is the routing problem with one truck and several drones to minimize the expected makespan and synchronization risk simultaneously. If several drones return to a truck at the same time and the truck only can retrieve the drone one-by-one, it will increase the potential of operation conflicts between drones. Considering potential operation conflicts, it is challenging to design a robust truck-drone schedule under traffic uncertainty. Additionally, the sequence is assumed to be predetermined in our model. Developing exact methods to obtain the optimal sequence is another interesting direction. The efficient frontier heuristic is time-consuming to solve the large-scale problems in practice. It is a limiting factor for applications and may introduce an opportunity to develop more efficient solution methods in future studies. As an exploratory study of the RTSP-D first, we hope that this paper will stimulate more research on the robust optimization of truck-drone routing problems.

## *Data Availability Statement*

The data presented in this study are available in supplementary materials here.

## References

Abdullahi, H., Reyes-Rubiano, L., Ouelhadj, D., Faulin, J., Juan, A.A., 2021. Modelling and multi-criteria analysis of the sustainability dimensions for the green vehicle routing problem. *European Journal of Operational Research* 292, 143–154.

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. *Transportation Science* 52, 965–981.

Ben-Tal, A., Ghaoui, L.E., Nemirovski, A., 2009. *Robust Optimization.*, Princeton University Press, Princeton, NJ.

Bertsimas, D., Brown, D.B., Caramanis, C., 2011. Theory and applications of robust optimization. *SIAM Review* 53, 464–501.

Bertsimas, D., Youssef, N., 2020. Stochastic optimization in supply chain networks: averaging robust solutions. *Optimization Letters* 14, 839–855.

Boccia, M., Masone, A., Sforza, A., Sterle, C., 2021. A column-and-row generation approach for the flying sidekick travelling salesman problem. *Transportation Research Part C: Emerging Technologies* 124, 102913.

Bouman, P., Agatz, N., Schmidt, M., 2018. Dynamic programming approaches for the traveling salesman problem with drone. *Networks* 72, 528–542.

Carlsson, J.G., Song, S., 2018. Coordinated logistics with a truck and a drone. *Management Science* 64, 4052–4069.

Dell'amico, M., Montemanni, R., Novellani, S., 2021. Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. *Optimization Letters* 15, 1617–1648.

Dell'amico, M., Montemanni, R., Novellani, S., 2022. Exact models for the flying sidekick traveling salesman problem. *International Transactions in Operational Research* 29, 1360–1393.

Demir, E., Bektas, T., Laporte, G., 2014. The Bi-objective pollution-routing problem. *European Journal of Operational Research* 232, 464–478.

Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 70–85.

Es Yurek, E., Ozmutlu, H.C., 2018. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies* 91, 249–262.

Eskandarpour, M., Ouelhadj, D., Hatami, S., Juan, A.A., Khosravi, B., 2019. Enhanced multi-directional local search for the Bi-objective heterogeneous vehicle routing problem with multiple driving ranges. *European Journal of Operational Research* 277, 479–491.

Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using K-means and genetic algorithm. *Journal of Industrial Engineering and Management* 9, 374–388.

Freitas, J.C., Penna, P.H.V., 2020. A Variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research* 27, 267–290.

Ha, Q.M., Deville, Y., Pham, Q.D., Hà, M.H., 2018. On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies* 86, 597–621.

Hong, L.J., Huang, Z., Lam, H., 2021. Learning-based robust optimization: procedures and statistical guarantees. *Management Science* 67, 3447–3467.

Jaillet, P., Qi, J., Sim, M., 2016. Routing optimization under uncertainty. *Operations Research* 64, 186–200.

Kitjacharoenchai, P., Ventresca, M., Moshref-Javadi, M., Lee, S., Tanchoco, J.M.A., Brunese, P.A., 2019. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers and Industrial Engineering* 129, 14–30.

Laporte, G., Louveaux, F., Mercure, H., 1992. The vehicle routing problem with stochastic travel times. *Transportation Science* 26, 161–170.

Lee, C., Lee, K., Park, S., 2012. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society* 63, 1294–1306.

Li, H., Wang, H., Chen, J., Bai, M., 2020. Two-echelon vehicle routing problem with time windows and mobile satellites. *Transportation Research Part B: Methodological* 138, 179–201.

Li, H.Q., Chen, J., Wang, F.L., Bai, M., 2021. Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: a review. *European Journal of Operational Research* 294, 1078–1095.

Macrina, G., Pugliese, L.D., Guerriero, F., Laporte, G., 2020. Drone-aided routing: a literature review. *Transportation Research Part C-Emerging Technologies*, 120, 102762.

Martins, L.C., Hirsch, P., Juan, A.A., 2021. Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *International Transactions in Operational Research* 28, 201–221.

Mbiadou Saleu, R.G., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A., 2018. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks* 72, 459–474.

Moshref-Javadi, M., Lee, S., Winkenbach, M., 2020. Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E: Logistics and Transportation Review* 136, 101887.

Munari, P., Moreno, A., De La Vega, J., Alem, D., Gondzio, J., Morabito, R., 2019. The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method. *Transportation Science* 53, 1043–1066.

Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* 54, 86–109.

Murray, C.C., Raj, R., 2020. The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies* 110, 368–398.

Poikonen, S., Golden, B., 2020a. The mothership and drone routing problem. *INFORMS Journal on Computing* 32, 249–262.

Poikonen, S., Golden, B., 2020b. Multi-visit drone routing problem. *Computers & Operations Research* 113, 104802.

Poikonen, S., Golden, B., Wasil, E.A., 2019. A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing*, 31, 335–346.

Rabin, M., Thaler, R.H., 2001. Anomalies: risk aversion. *Journal of Economic Perspectives* 15, 219–232.

Raj, R., Murray, C. 2020 The multiple flying sidekicks traveling salesman problem with variable drone speeds. *Transportation Research Part C: Emerging Technologies*, 120, 102813.

Rojas Viloria, D., Solano-Charris, E.L., Muñoz-Villamizar, A., Montoya-Torres, J.R., 2021. Unmanned aerial vehicles/drones in vehicle routing problems: a literature review. *International Transactions in Operational Research* 28, 1626–1657.

Rooderkerk, R.P., Van Heerde, H.J., 2016. Robust optimization of the 0–1 knapsack problem: Balancing risk and return in assortment optimization. *European Journal of Operational Research* 250, 842–854.

Salama, M., Srinivas, S., 2020. Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies* 114, 620–642.

Savuran, H., Karakaya, M., 2016. Efficient route planning for an unmanned air vehicle deployed on a moving carrier. *Soft Computing* 20, 2905–2920.

Schermer, D., Moeini, M., Wendt, O., 2019. A hybrid Vns/Tabu search algorithm for solving the vehicle routing problem with drones and En route operations. *Computers & Operations Research* 109, 134–158.

Schermer, D., Moeini, M., Wendt, O., 2020. A branch-and-cut approach and alternative formulations for the traveling salesman problem with drone. *Networks* 76, 164–186.

Shavarani, S.M., Golabi, M., Izbirak, G., 2021. A capacitated biobjective location problem with uniformly distributed demands in the UAV-supported delivery operation. *International Transactions in Operational Research* 28, 3220–3243.

Singh, S., Kumar, R., Panchal, R., Tiwari, M.K., 2021. Impact of COVID-19 on logistics systems and disruptions in food supply chain. *International Journal of Production Research* 59, 1993–2008.

Torija, A.J., Li, Z., Self, R.H., 2020. Effects of a hovering unmanned aerial vehicle on urban soundscapes perception. *Transportation Research Part D-Transport and Environment* 78, 102195.

Tortonesi, M., Stefanelli, C., Benvegnu, E., Ford, K., Suri, N., Linderman, M., 2012. Multiple-UAV coordination and communications in tactical edge networks. *IEEE Communications Magazine* 50, 48–55.

Vu, L., Vu, D.M., Hà, M.H., Nguyen, V.-P., 2021. The two-echelon routing problem with truck and drones. *International Transactions in Operational Research* 29, 2968–2994.

Vural, D., Dell, R.F., Kose, E., 2019. Locating unmanned aircraft systems for multiple missions under different weather conditions. *Operational Research* 21, 725–744.

Wang, K., Yuan, B., Zhao, M., Lu, Y., 2020. Cooperative route planning for the drone and truck in delivery services: a Bi-objective optimisation approach. *Journal of the Operational Research Society* 71, 1657–1674.

Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: several worst-case results. *Optimization Letters* 11, 679–697.

Wang, Z., Sheu, J.B., 2019. Vehicle routing problem with drones. *Transportation Research Part B: Methodological* 122, 350–364.

Wolf, H., 2020. We're about to see the golden age of drone delivery – here's why [Online]. World Economic Forum. Available: https://www.weforum.org/agenda/2020/07/golden-age-drone-delivery-covid-19-coronavirus-pandemic-technology/ (accessed 1 October 2020).

Xiang, S., Wang, L., Xing, L.N., Du, Y.H., 2021. An effective memetic algorithm for UAV routing and orientation under uncertain navigation environments. *Memetic Computing* 13, 169–183.

Zhen, L., Hu, Z., Yan, R., Dan, Z.G., Wang, S.A., 2020. Route and speed optimization for liner ships under emission control policies. *Transportation Research Part C-Emerging Technologies* 110, 330–345.