Submitted to

manuscript

Robust Drone Delivery with Weather Information

Chun Cheng
School of Economics and Management, Dalian University of Technology, Dalian, 116024, China
chun.cheng@polymtl.ca

Yossiri Adulyasak
GERAD and Department of Logistics and Operations Management, HEC Montréal, Montréal, Quebec, H3T 2A7, Canada
yossiri.adulyasak@hec.ca

Louis-Martin Rousseau
Polytechnique Montréal and CIRRELT, Montréal, H3C 3A7, Canada
louis-martin.rousseau@cirrelt.net

Problem definition: Drone delivery has recently garnered significant attention due to its potential for faster delivery at a lower cost than other delivery options. When scheduling drones from a depot for delivery to various destinations, the dispatcher must take into account the uncertain wind conditions, which affect the delivery times of drones to their destinations, leading to late deliveries. **Methodology:** To mitigate the risk of delivery delays caused by wind uncertainty, we propose a two-period drone scheduling model to robustly optimize the delivery schedule. In this framework, the scheduling decisions are made in the morning, with the provision for different delivery schedules in the afternoon that adapt to updated weather information available by midday. Our approach minimizes the essential riskiness index, which can simultaneously account for the probability of tardy delivery and the magnitude of lateness. Using wind observation data, we characterize the uncertain flight times via a cluster-wise ambiguity set, which has the benefit of tractability while avoiding overfitting the empirical distribution. A branch-and-cut (B&C) algorithm is developed for this adaptive distributionally framework to improve its scalability. **Results:** Our adaptive distributionally robust model can effectively reduce lateness in out-of-sample tests compared to other classical models. The proposed B&C algorithm can solve instances to optimality within a shorter time frame than a general modeling toolbox. **Managerial implications:** Decision-makers can use the adaptive robust model together with the cluster-wise ambiguity set to effectively reduce service lateness at customers for drone delivery systems.

Key words: drone delivery, uncertain flight time, uncertain wind conditions, cluster-wise ambiguity set, distributionally robust optimization.

1. Introduction

Drone delivery is expected to redefine the traditional shipping market, as it is faster, cheaper, and not restricted to congested roads, compared to truck delivery (Agatz et al. 2018). This innovative transportation mode has gained increasing interest in recent years, especially since 2013 when Amazon CEO Jeff Bezos announced plans for Amazon Prime Air in an interview (Rose 2013). Major competitors in the delivery service industry, including DHL, Google, and JD.com, have also been

developing and testing their drone models in recent years. On October 18, 2019, Alphabet's drone delivery company, Wing, launched the first U.S. commercial drone delivery flight (Doherty 2019). Startups like Flirtey and Drone Delivery Canada have focused on designing and implementing commercially viable drone delivery systems. Aside from their commercial use, drones can also be applied in humanitarian logistics to deliver relief items (Rabta et al. 2018) and in healthcare projects to refill medicine and pick up test kits (Kim et al. 2017).

A well-designed drone delivery system mitigates the risks of weather uncertainty. Uncertain wind conditions, *i.e.*, speeds and directions of the wind, can impact the transit times of the drones to their destinations, leading to late deliveries or even cancellations of service (Walker 2014). Enderle (2019) suggests that delays caused by weather conditions should be overcome before drone-based order delivery becomes a reality. Black (2017) further states that the future of drone delivery depends on its capability to adapt to different scenarios of weather conditions, and drone delivery companies, such as Zipline, are actively working to build models that allow drones to safely and efficiently operate under different weather scenarios.

In order to tackle this important issue, we explore how the availability of weather data, such as wind observations, can be used to improve scheduling decisions in a drone delivery system, where a fleet of identical drones is dispatched to visit a set of N geographically dispersed customers. Each drone can make multiple round trips, where each round trip is a flight from the depot to the customer and back to the depot for preparation for the next delivery, which may include tasks such as mounting a new payload and/or swapping out a depleted battery. The transit times of the drones between the depot and their assigned customers are affected by wind conditions, *i.e.*, the wind may increase or decrease their flight times depending on its speed and direction. Figure 1 illustrates the relationship between the wind conditions and flight times. The depot and customers are distributed in a two-dimensional space and, without any loss of generality, the coordinate of the depot is situated at $(0, 0)$. The wind vector is represented by the polar coordinate (r, θ), where r denotes the speed of the wind, and θ is the angle between the direction of the wind and the x-axis. The location of customer i, $i \in [N]$ is represented by the polar coordinate (d_i, ϕ_i), where d_i is the distance from the depot and ϕ_i is the angle between the arc connecting the depot and customer i and the x-axis.

The flight time of a drone depends on the ground speed, *i.e.*, the speed relative to the ground. The ground speed is the vector sum of the wind speed and the airspeed. The airspeed is the relative velocity between a drone and the air, which generates the lift for the drone to move (Bocewicz et al. 2020, Radzki et al. 2021). The airspeed of drones in the forward direction (*i.e.*, from the depot to customers) for the ith customer, depending on the weight and size of the customer's payload, is denoted by $\bar{r}_i$. In the absence of a payload, the airspeed of drones in the backward direction is

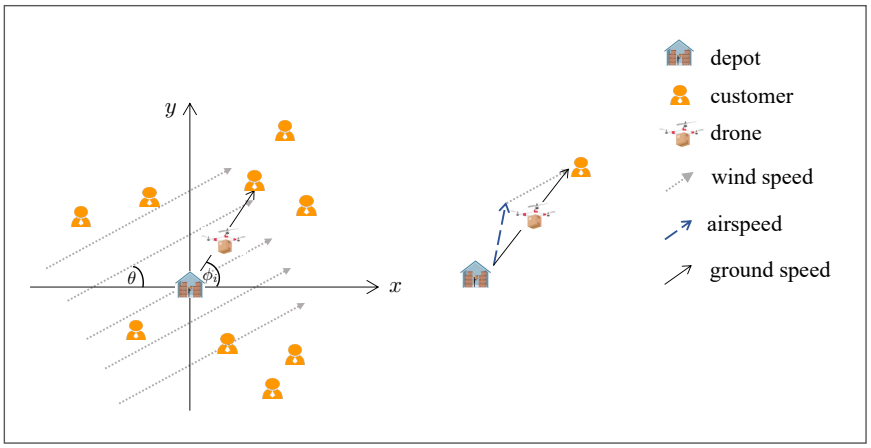


Figure 1 Illustration for calculating flight times. The right part is based on vector addition.

$\bar{r}_0$, $\bar{r}_0 \geq \bar{r}_i$, $i \in [N]$. We assume $r < \bar{r}_i$, $i \in [N]$ to forbid delivery under very windy conditions. The automatic navigation system within a delivery drone would adapt according to the wind conditions, to offset the wind's influence on flight direction, such that the drone can fly straightly from the depot to customers. Based on the definition and the calculation equation of the ground speed in the three studies above, we can derive the nominal forward flight time u_i and the backward flight time v_i as

$$\begin{cases} u_i(r,\theta) \triangleq \dfrac{d_i}{\sqrt{(\bar{r}_i)^2 - r^2\sin^2(\theta-\phi_i)} + r\cos(\theta-\phi_i)}, \\ v_i(r,\theta) \triangleq \dfrac{d_i}{\sqrt{(\bar{r}_0)^2 - r^2\sin^2(\theta-\phi_i)} - r\cos(\theta-\phi_i)}, \end{cases} \tag{1}$$

where the denominators are the forward and backward ground speeds, respectively. Note that since $r < \min_{i\in[N]} \bar{r}_i$, the airspeed of the drone has a positive influence on flight duration. Meanwhile, a drone could reduce its speed or even delay its flight, which may be necessary to avoid an early arrival at the customer's location. For instance, if a drone departs from the depot at τ and the customer can only be served after $\bar{\tau}^1$, then the arrival time of the drone at the customer's location would be $\max\{\tau + u_i, \bar{\tau}^1\}$.

In practice, the wind conditions cannot be perfectly predicted, and the actual flight times may deviate from the nominal values. We denote the uncertain forward and backward flight times resulting from the uncertain wind vector as $\tilde{u}_i$ and $\tilde{v}_i$, $i \in [N]$, respectively. Fortunately, weather data, such as historical wind observations, are commonly available from the national or local meteorological information center and can be used to predict flight times. Figure 2 is a wind rose diagram of a province in China during a particular time interval. Wind information is reported in terms of speed and direction. Each piece of information can be recognized as a wind observation sample, based on which we can calculate the flight times between the depot and customers, thus generating samples of flight times. Specifically, given H samples of the observed wind speed and

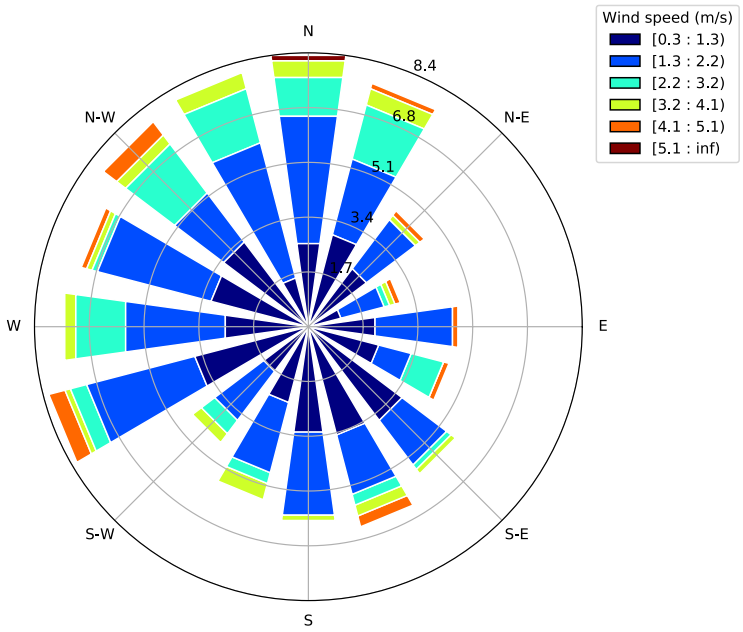


Figure 2 Wind information collected from 145 subregions of Sichuan Province, China, ranging from the time interval 00:00 to 04:00 on September 14$^{\text{th}}$, 2019. Data are downloaded from the China National Meteorological Information Center (http://data.cma.cn/site/index.html). The color-coded bands represent wind speed ranges, and the circles denote different frequencies (from 0 to 8.4%).

direction, (r_h, θ_h), $h \in [H]$, the corresponding forward and backward flight times of a drone to the ith customer, $i \in [N]$ are given by $u_i(r_h, \theta_h)$ and $v_i(r_h, \theta_h)$, respectively.

Contributions

We propose a drone delivery system to fulfill customers' requests for their delivery to be made in either the morning or in the afternoon. Given the weather data, our goal is to robustly optimize the schedules of the drone delivery system to mitigate the risks of delivery delays due to uncertainty in wind conditions. Our contributions are summarized as follows.

1. We propose a novel two-period data-driven adaptive distributionally robust optimization (DRO) model that permits the modeler to use wind observation data to improve scheduling decisions in a drone delivery system. The scheduling decisions for the fleet of drones are made in the morning, with the provision for different delivery schedules in the afternoon that adapt to updated weather information available by midday.
2. We show how to construct the ambiguity set characterizing the uncertain flight times of the drones using wind observation data. We propose a cluster-wise ambiguity set, which has the benefits of tractability while avoiding overfitting the empirical distribution. We incorporate other weather information in the form of discrete scenarios to enhance the ambiguity set.
3. Inspired by the *essential riskiness index* (ERI) proposed by Zhang et al. (2019), we propose a new decision criterion based on the characteristics of our problem. The new criterion can

simultaneously account for the probability of tardy delivery and the magnitude of lateness as the ERI does. More importantly, it leads to a more straightforward and computationally scalable formulation for our problem.

4. For greater computational scalability, we propose a branch-and-cut (B&C) approach to solve the adaptive DRO model.
5. We show in our computational studies that a small number of clusters obtained via K-means can achieve high-quality, robust drone delivery schedules. We validate that the adaptive DRO model can effectively reduce lateness in out-of-sample tests compared to other classical models.

Literature Review

This section reviews related works on drone delivery problems and data-driven robust optimization (RO). We also discuss the differences between our problem and other scheduling problems. For more details about drones' applications, see the review paper by Moshref-Javadi and Winkenbach (2021). For RO methods, interested readers can refer to the review paper by Yanıkoğlu et al. (2019).

Drone delivery problems. Most studies in drone delivery consider a truck-drone tandem system, where one truck carries one drone to deliver packages. When the truck performs deliveries along a traveling salesman problem (TSP) route, the drone can be launched from the truck at a customer location to make one delivery. Then, the truck continues its route and retrieves the drone at a new customer location. If the drone is not operational, it is carried by the truck. This problem is termed the flying sidekick TSP (FSTSP) by Murray and Chu (2015). Agatz et al. (2018) extend the FSTSP by allowing the truck to wait at the customer site for the drone to return. They denote the problem as the TSP with drone. Many studies focus on developing solution methods for the two problems above, *e.g.*, simulated annealing heuristic (Ponza 2016), dynamic programming (Bouman et al. 2018), and branch-and-price (Roberti and Ruthmair 2021). Other variants are also studied. For example, Sacramento et al. (2019) address a vehicle routing problem with drones, where multiple trucks, each equipped with one drone, are considered. There are other drone-related delivery problems where trucks and drones do not work in tandem. Murray and Chu (2015) introduce a parallel drone scheduling TSP, where drones and trucks work independently. Dorling et al. (2016) study a multi-trip drone routing problem where only drones are used. Recently, Gao et al. (2023) explore an ambulance-drone system in emergency medical services, where drones are dispatched to provide the first responses for some requests with bystander assistance, followed by subsequent ambulance deployment. The authors consider the randomness of request arrival, bystander availability, and travel times. They formulate the problem as a Markov decision process and develop an approximate dynamic programming solution method. Unlike our work, they assume known distributions of random variables. Gupta et al. (2022) propose a new estimator of out-of-sample performance of a policy for data-driven optimization problems, particularly suitable

for scarce data settings. They evaluate the estimator using a drone-assisted emergency medical response problem with uncertain response times. They estimate drones' response times mainly based on the straight-line travel distances between nodes. We note that their work focuses on providing a general framework to mitigate the *optimizer's curse* (Smith and Winkler 2006) by using limited data. However, our study explicitly considers the impact of uncertain weather conditions on drones' travel times and aims to mitigate the risks of service delay. Additionally, to improve the out-of-sample performance, we adopt an adaptive policy by utilizing observed weather information.

One significant aspect that makes drone delivery problems particularly challenging, as opposed to the typical vehicle delivery problems, is the influence of weather conditions on drones' operational efficiency. Some recent studies have tackled this issue by directly or indirectly incorporating weather conditions into the optimization models. Radzki et al. (2019) study a drone routing problem considering the impact of wind on energy consumption. They assume that all the influencing factors, like ground speeds, wind conditions, and drone weight during a trip, are given parameters, resulting in constant energy consumption values for each arc. Thibbotuwawa et al. (2019) design a decision support tool for the multi-trip drone fleet mission planning problem under different weather conditions. They decompose the problem into subproblems, dividing the time period into time slots with analogous weather conditions. They then decide the customer clusters for each slot and the routes for each cluster. The sequence of routes within each cluster is finally decided. Kim et al. (2017) consider the influence of wind on flight time by setting asymmetric flight distances.

Vural et al. (2019) study a drone location routing problem and state that weather such as snow, fog, and heavy rain would limit drone operations in some regions. In their study, weather scenarios are known with an occurrence probability, and each scenario indicates whether a specific region can operate drones. They formulate the problem as a two-stage stochastic programming model. Kim et al. (2019) study a stochastic drone facility location problem. They state that some weather conditions can negatively affect energy consumption, affecting drones' flight ranges. A chance constraint is used to ensure that the probability that a drone's flight range is larger than or equal to the travel distance between a facility and a customer is acceptable. They assume the flight range follows an exponential distribution and reformulate the resulting model as a mixed-integer linear programming model. Kim et al. (2018) consider a drone scheduling problem with uncertain flight ranges, resulting from uncertain air temperature on battery duration. They use an RO method and compare the performance of three uncertainty sets—polyhedral, box, and ellipsoidal. They use historical and forecast data to estimate the hourly temperature (a span of 12 hours) over an area and then calculate the flight range deviations via a regression function. These deviations are fixed values and are treated as different scenarios with weights. Then, their uncertainty sets are constructed to denote the random weights.

The works above either ignore the impact of weather completely or consider it in a simplified way. Specifically, future weather conditions (or flight range/time) are assumed to be deterministic, follow a known distribution, or belong to some scenarios with known probabilities. Only Kim et al. (2018) use a distribution-free method; however, they consider limited scenarios and need to specify the budget or radius of the uncertainty sets. Unlike nearly all reviewed papers, we use the DRO approach with a cluster-wise ambiguity set, where the distribution of random variables is assumed to lie in an ambiguity set rather than being known perfectly in advance.

Data-driven RO. Our paper falls within the realm of data-driven RO, where the key step is constructing ambiguity sets from historical data. Details on the construction of these sets can be found in Delage and Ye (2010) and Bertsimas et al. (2018). Recently, Chen et al. (2020) propose an event-wise ambiguity set, which is rich enough to capture a wide range of ambiguity sets, such as statistic-based and machine-learning-based ones. The works of Hao et al. (2020) and Perakis et al. (2023) are novel applications of the framework proposed in Chen et al. (2020). Specifically, Hao et al. (2020) address a single-period vehicle allocation problem with uncertain demand, which is related to weather conditions (rainy or sunny). They use a multivariate regression tree to construct the ambiguity set. Perakis et al. (2023) study a two-period, multi-item joint pricing and production problem, where the K-means clustering algorithm is utilized to partition the demand residuals. Then, a cluster-wise ambiguity set is constructed.

Differences between our problem and other scheduling problems. We first discuss the differences between our problem and the stochastic appointment sequencing and scheduling problem (ASSP). In the ASSP, a set of jobs with stochastic durations is to be performed by a single server. The planner makes the job sequencing decisions, *i.e.*, determining the orders in which the jobs will be performed, and the appointment scheduling decisions, *i.e.*, determining the starting times for jobs with a given sequence. The objective is to minimize the weighted costs arising from job waiting times, server idle time, and overtime (Denton et al. 2007, Mancilla and Storer 2012, Mak et al. 2014, 2015, de Kemp et al. 2021). Our problem mainly distinguishes the ASSP in two aspects. First, the ASSP often assumes only a single server; thus, no job-server assignment decisions need to be made. In contrast, such assignments are part of the decisions in our problem, as multiple servers are considered. Second, the ASSP is a single-period problem, while our problem involves two periods. Thus, for the jobs that can be performed in either period, we decide both the job-server assignments and the time periods to perform these jobs. Moreover, our second-period assignment and sequencing decisions can adapt to realized covariate information (the weather conditions in our context) in the first period. However, the decisions of the ASSP are static. Namely, once made, they are fixed regardless of the uncertainty realization, because jobs' scheduled appointment times are communicated to customers once they are decided. Thus, although we can utilize covariate

information, *e.g.*, the types of the server and patients, to construct a scenario-wise ambiguity set for the distributionally robust version of the ASSP, adapting the decisions is impossible in the ASSP. In a nutshell, the involvement of multiple servers and the two-period assumption make our problem distinct from the stochastic ASSP in terms of decision variables and modeling frameworks.

Our work is also related to Zhang et al. (2019) and Zhang et al. (2021), where they study a TSP with time windows and a vehicle routing problem with time window (VRPTW) under travel time uncertainty, respectively. They minimize the ERI, a convex decision criterion that simultaneously considers the probability of tardy delivery and the magnitude of lateness. To characterize the uncertainty, the former explores a moment-based ambiguity set, and the latter considers a Wasserstein distance-based ambiguity set. Unlike their static decision models, we consider an adaptive model, which has the benefit of alleviating the conservatism of robust solutions. We do so by developing a cluster-wise ambiguity set that is compatible with our two-period adaptive model. Moreover, to reduce the computational burden resulting from adaptability, we introduce a new decision criterion based on the structure of our problem, which keeps the desirable properties of the ERI, and also leads to a more computationally scalable model.

The rest of the paper is organized as follows. At the end of this section, we summarize the notation used throughout the paper. Section 2 constructs the ambiguity set from weather data. Section 3 presents the decision criterion, builds the DRO model, and introduces the adaptive policy. Section 4 reformulates the DRO model and presents the solution method. Section 5 reports numerical results. Section 6 concludes this paper. All the proofs appear in Appendix C.

Notation. We denote by $[N] \triangleq \{1, \ldots, N\}$ the set of positive running indices up to N. Boldface lowercase and uppercase characters represent vectors and matrices with appropriate dimensions, respectively. $\boldsymbol{a}'$ is the transpose of $\boldsymbol{a}$. We use $\mathcal{P}_0(\mathbb{R}^I)$ to represent the set of all distributions on $\mathbb{R}^I$. A random variable, $\tilde{\boldsymbol{z}}$ is denoted with a tilde sign and we use $\tilde{\boldsymbol{z}} \sim \mathbb{P}$, $\mathbb{P} \in \mathcal{P}_0(\mathcal{W})$, $\mathcal{W} \subseteq \mathbb{R}^I$ to define $\tilde{\boldsymbol{z}}$ as an I-dimensional random variable with distribution $\mathbb{P}$ over the support $\mathcal{W}$. For a vector $\boldsymbol{u} \in \mathbb{R}^I$, the expression $|\boldsymbol{u}|$ denotes the vector of absolute values of the components of $\boldsymbol{u}$. $\boldsymbol{e}$ corresponds to the vector of 1s with an approximate dimension. $\boldsymbol{e}_i$ is the ith standard basis vector.

2. Weather Uncertainty and Influence on Flight

We model a drone delivery system comprising a fleet of D identical drones that are dispatched to visit a set of N, $N > D$ geographically dispersed customers. Each drone can make multiple round trips, where each round trip is a flight from the depot to the customer and back to the depot for preparation for the next delivery. We assume that the drone has sufficient energy to perform a round trip since a depleted battery can quickly be swapped with a fully charged one. Hence, with a sufficient supply of charged batteries, we do not need to incorporate charging decisions

into our model. If a fully charged battery is insufficient for a round trip between the depot and a customer site, then that customer would be deemed a drone-ineligible customer. We characterize the uncertainty of flight times as the result of wind conditions. Without loss of generality, we assume that the time for mounting new payloads and swapping batteries at the depot is negligible, as it is not influenced by wind conditions and can be incorporated into the model easily (Ham 2018). The service time at customer locations is also neglected here because, whenever necessary, we can add it to the forward or backward flight time as a constant number.

We consider a two-period model, where Period 1 and Period 2 represent the morning and the afternoon, respectively. Period 1 starts at time 0 and ends at the midday time, $\bar{\tau}^1$, while Period 2 starts at $\bar{\tau}^1$ and ends at $\bar{\tau}^2$. At the beginning of Period 1, the model optimizes the scheduling policy, which determines the drones' delivery schedules in the morning, while the schedules in the afternoon can flexibly adapt to the observed wind vector, (r^1, θ^1), in the morning, and/or other weather information that would be known by midday. We assume that the modeler has H samples of wind observation data at the beginning of the time period. Under sample h, $h \in [H]$, we define (r_h^1, θ_h^1) and (r_h^2, θ_h^2) to be the observed wind vectors in Period 1 and Period 2, respectively.

Ambiguity Set for Uncertain Flight Times

The forward and backward flight times for serving customer i, $i \in [N]$ in Period 1 are uncertain, and we denote them by the random variables $\tilde{u}_i^1$ and $\tilde{v}_i^1$, which are functions of the wind vectors from Equation (1). Likewise, we also denote by the random variables $\tilde{u}_i^2$ and $\tilde{v}_i^2$ the forward and backward flight times for serving customer i, $i \in [N]$ in Period 2. For convenience, we define the vectorial notation $\tilde{\boldsymbol{u}}^t \triangleq (\tilde{u}_1^t, \ldots, \tilde{u}_N^t)'$ and $\tilde{\boldsymbol{v}}^t \triangleq (\tilde{v}_1^t, \ldots, \tilde{v}_N^t)'$, $t \in [2]$. Based on Equation (1), we also define $\boldsymbol{u}(r, \theta) \triangleq (u_1(r, \theta), \ldots, u_N(r, \theta))'$ and $\boldsymbol{v}(r, \theta) \triangleq (v_1(r, \theta), \ldots, v_N(r, \theta))'$.

We first characterize the ambiguity set in the first period using a cluster-wise ambiguity set introduced by Chen et al. (2020). We partition the wind vector chart into K^1 non-overlapping clusters $\mathcal{U}_k^1$, $k \in [K^1]$ so that the index set of the subsamples

$$\mathcal{L}_k = \{h \in [H] \mid (r_h^1, \theta_h^1) \in \mathcal{U}_k^1\}, \tag{2}$$

are each associated with a region to which the morning wind vectors belong (see Figure 3).

We introduce the random variable $\tilde{\kappa}^1$ taking discrete values in $[K^1]$ to represent the scenario $\tilde{\kappa}^1 = k$ associated with the cluster $\mathcal{U}_k^1$ that would contain the realized wind vector observed in

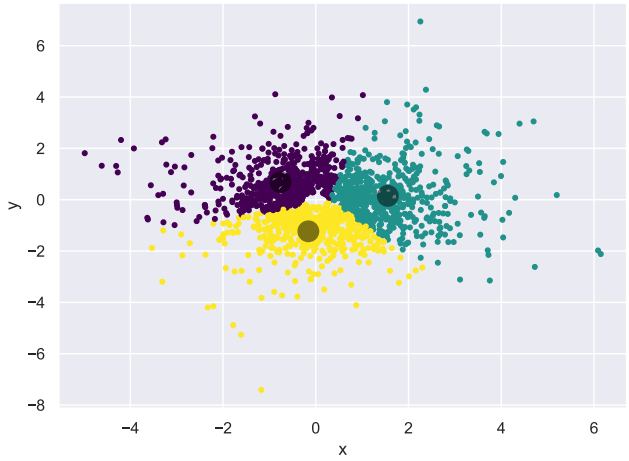


Figure 3 Partitioning the wind vector chart into clusters using the K-means clustering algorithm. Note that wind vectors are converted from polar coordinates to Cartesian coordinates to guarantee that coordinates are in the same scales before performing the clustering. The dark points in each cluster are the centroids.

Period 1. Accordingly, using wind observational data, we construct the cluster-wise ambiguity set $\mathcal{G}^1 \subseteq \mathcal{P}(\mathbb{R}^{2N} \times [K^1])$ associated with the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1)$ as follows:

$$\mathcal{G}^1 = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{2N} \times [K^1]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\mu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\nu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^1 - \boldsymbol{\mu}_k^1| \mid \tilde{\kappa}^1 = k\right] \leq \boldsymbol{\sigma}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^1 - \boldsymbol{\nu}_k^1| \mid \tilde{\kappa}^1 = k\right] \leq \boldsymbol{\varsigma}_k^1 & \forall k \in [K^1] \\ \mathbb{P}\left[\begin{array}{l} \underline{\boldsymbol{u}}_k^1 \leq \tilde{\boldsymbol{u}}^1 \leq \bar{\boldsymbol{u}}_k^1 \\ \underline{\boldsymbol{v}}_k^1 \leq \tilde{\boldsymbol{v}}^1 \leq \bar{\boldsymbol{v}}_k^1 \end{array} \middle| \tilde{\kappa}^1 = k \right] = 1 & \forall k \in [K^1] \\ \mathbb{P}\left[\tilde{\kappa}^1 = k\right] = |\mathcal{L}_k|/H & \forall k \in [K^1] \end{array} \right\}, \tag{3}$$

where the parameters of the forward flight times associated with the cluster k, $k \in [K^1]$, are characterized as follows:

$$\boldsymbol{\mu}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \boldsymbol{u}(r_h^1, \theta_h^1), \qquad \boldsymbol{\sigma}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \left| \boldsymbol{u}(r_h^1, \theta_h^1) - \boldsymbol{\mu}_k^1 \right|,$$
$$\left[\underline{\boldsymbol{u}}_k^1\right]_i = \min_{h \in \mathcal{L}_k} u_i(r_h^1, \theta_h^1), \qquad \left[\bar{\boldsymbol{u}}_k^1\right]_i = \max_{h \in \mathcal{L}_k} u_i(r_h^1, \theta_h^1), \qquad \forall i \in [N].$$

The parameters of the backward flight times can be obtained in a similar way.

Note that the ambiguity set does not explicitly express the relationship between the wind conditions and the flight times, *i.e.*, Equation (1). Instead, wind conditions and flight times are implicitly linked by the clusters. Specifically, after clustering the wind samples, we calculate the historical flight times associated with each sample in each cluster using Equation (1). In this way, we obtain the samples of flight times in each cluster. Then, we extract the statistical information on flight times and construct the cluster-wise ambiguity set.

In RO, the ambiguity set is often designed to encompass distributions beyond the empirical distribution to mitigate the risk of overfitting and improve out-of-sample performance. When $K^1 = 1$, $\mathcal{G}^1$ reduces to a marginal moment ambiguity set which, analogous to the *box uncertainty set*, can be too conservative when applied to solve DRO problems. As the number of clusters increases, the level of conservativeness reduces, although the computational complexity increases. In the extreme case of one sample per cluster, the ambiguity set only contains the empirical distribution due to the support and deviation constraints. The empirically optimized solution may suffer from the issue of overfitting, leading to poor out-of-sample performance, a phenomenon known as the *optimizer's curse* (Smith and Winkler 2006). In practice, the number of clusters is often determined empirically in validation tests to achieve a tradeoff between conservativeness and computational complexity.

By midday, in preparation for Period 2, the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1)$ would be realized. Apart from the wind data, we assume that there is other weather information like atmospheric pressures, weather forecasts, and other covariates, which could be used to predict the wind conditions in the afternoon. To incorporate this information into our ambiguity set, we discretize the weather information into S finite scenarios, which can be done using standard classification approaches in machine learning. Accordingly, we use $\tilde{s}$ to denote the random weather scenarios, which would be realized by midday. We assume that the weather data contains historical information of the weather scenarios, which we denote by s_h, $h \in [H]$. We define the index set of the subsamples associated with the different weather scenarios as $\mathcal{V}_s = \{h \in [H] \mid s_h = s\}$.

In the absence of other weather information beyond wind observation data, we can use the random scenario $\tilde{\kappa}^1$ defined for the ambiguity set $\mathcal{G}^1$ as the random scenario for predicting wind conditions in the afternoon, in which case, $S = K^1$ and $\mathcal{V}_s = \mathcal{L}_s$ for all $s \in [S]$.

Given a realized scenario $\tilde{s} = s$, we can characterize Period 2 flight times $(\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2)$ via a cluster-wise ambiguity set as before. We partition the wind vectors (r_h^2, θ_h^2), $h \in \mathcal{V}_s$ into K^2 non-overlapping clusters $\mathcal{U}_g^2$, $g \in [K^2]$. Each cluster $g \in [K^2]$ is associated with the subset of samples $\mathcal{L}_{sg} = \left\{h \in \mathcal{V}_s \mid (r_h^2, \theta_h^2) \in \mathcal{U}_g^2\right\}$. We also introduce the random variable $\tilde{\kappa}^2$, taking discrete values in $[K^2]$ to represent the scenario $\tilde{\kappa}^2 = g$ associated with the cluster $\mathcal{U}_g^2$ that would contain the realized wind vector observed in Period 2. For each scenario s, $s \in [S]$, we construct the ambiguity set

$\mathcal{G}_s^2 \subseteq \mathcal{P}(\mathbb{R}^{2N} \times [K^2])$ as follows:

$$\mathcal{G}_s^2 = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{2N} \times [K^2]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\kappa}^2) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\mu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\nu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2 - \boldsymbol{\mu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\sigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2 - \boldsymbol{\nu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \leq \boldsymbol{\varsigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{P}\left[\begin{array}{l} \underline{\boldsymbol{u}}_{sg}^2 \leq \tilde{\boldsymbol{u}}^2 \leq \bar{\boldsymbol{u}}_{sg}^2 \\ \underline{\boldsymbol{v}}_{sg}^2 \leq \tilde{\boldsymbol{v}}^2 \leq \bar{\boldsymbol{v}}_{sg}^2 \end{array} \middle| \, \tilde{\kappa}^2 = g \right] = 1 & \forall g \in [K^2] \\ \mathbb{P}\left[\tilde{\kappa}^2 = g\right] = |\mathcal{L}_{sg}|/|\mathcal{V}_s| & \forall g \in [K^2] \end{array} \right\}, \tag{4}$$

where the parameters of the forward flight times in the cluster g, $g \in [K^2]$, are determined as

$$\boldsymbol{\mu}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}|} \sum_{h \in \mathcal{L}_{sg}} \boldsymbol{u}(r_h^2, \theta_h^2), \qquad \boldsymbol{\sigma}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}|} \sum_{h \in \mathcal{L}_{sg}} \left|\boldsymbol{u}(r_h^2, \theta_h^2) - \boldsymbol{\mu}_{sg}^2\right|,$$
$$\left[\underline{\boldsymbol{u}}_{sg}^2\right]_i = \min_{h \in \mathcal{L}_{sg}} u_i(r_h^2, \theta_h^2), \qquad \left[\bar{\boldsymbol{u}}_{sg}^2\right]_i = \max_{h \in \mathcal{L}_{sg}} u_i(r_h^2, \theta_h^2), \qquad \forall i \in [N].$$

The parameters of the backward flight times can be characterized similarly.

In general, the discrete random variable $\tilde{\kappa}^1$, $\tilde{\kappa}^2$, and $\tilde{s}$ are dependent, and hence we denote $\mathcal{W}$ as the support of the joint discrete random variable $(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})$. We are now ready to characterize the full ambiguity set. The actual joint distribution of $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{P}$ is unknown but belongs to an ambiguity set, $\mathcal{F}$, as follows:

$$\mathcal{F} = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{4N} \times [S]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{Q} & \\ \exists \mathbb{P} \in \mathcal{P}(\mathbb{R}^{4N} \times \mathcal{W}): & \\ \quad (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, (\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})) \sim \mathbb{P} & \\ \exists \mathbb{Q}^1 \in \mathcal{G}^1 : (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\kappa}^1) \sim \mathbb{Q}^1 & \\ \exists \mathbb{Q}_s^2 \in \mathcal{G}_s^2 : (\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\kappa}^2)|_{\tilde{s}=s} \sim \mathbb{Q}_s^2 & \forall s \in [S] \\ \mathbb{P}\left[(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s}) = (k, g, s)\right] = q_{kgs} & \forall (k, g, s) \in \mathcal{W} \end{array} \right\}, \tag{5}$$

where

$$q_{kgs} = \frac{|\mathcal{V}_s \cap \mathcal{L}_k \cap \mathcal{L}_{sg}|}{H}.$$

The explicit form of the ambiguity set, $\mathcal{F}$, is provided in Appendix A.

3. The Drone Delivery Model

We aim to robustly optimize the drone schedules to mitigate the risks of delivery delays caused by wind uncertainty. Customers specify their preference for delivery in the morning or afternoon. Hence, we partition the customers into three groups, $\mathcal{C}^1$, $\mathcal{C}^2$, and $\mathcal{C}^3$. The first two groups, $\mathcal{C}^1$ and $\mathcal{C}^2$, are the sets of customers that expect to have their delivery made in Period 1 and Period

2, respectively. The third group $\mathcal{C}^3$ is the set of unconstrained customers who can be served in either period. Customers in $\mathcal{C}^3$ are notified to expect their delivery in either Period 1 or Period 2. Hence, customers assigned to be served in Period 1 should be served within $[0, \bar{\tau}^1]$, while customers assigned to be served in Period 2 should be served within $[\bar{\tau}^1, \bar{\tau}^2]$. Poor service occurs whenever the scheduled delivery time windows cannot be met.

Decision Criterion

For a given delivery policy, the arriving time of a drone is a function of the random variable $(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{P}, \mathbb{P} \in \mathcal{F}$, which we denote by $\xi(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s})$ for some function $\xi : \mathbb{R}^{4N} \times [S] \mapsto \mathbb{R}$. For convenience, we use $\tilde{\xi} \triangleq \xi(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s})$ to represent the random arriving time. A service is delayed whenever $\tilde{\xi} > \tau$, where $\tau \in \{\bar{\tau}^1, \bar{\tau}^2\}$ is the delivery time expected by the customer.

Our goal is to mitigate the risks of service delays under distributional ambiguity, and we evaluate such risks using the ERI proposed by Zhang et al. (2019). The index is endowed with good computational properties, and it performs well as an optimization criterion in a vehicle routing problem (Zhang et al. 2021). Although the entropic value-at-risk (Nemirovski and Shapiro 2007) and the requirement violation index (Jaillet et al. 2016) are also used to evaluate risks for some delivery problems, they admit a tractable formulation only if the random variables are independent (Long and Qi 2014, Zhang et al. 2021). However, the random flight times in our problem are correlated, as they jointly depend on the wind conditions.

Definition 1. Given an arriving time function, $\xi : \mathbb{R}^{4N} \times [S] \mapsto \mathbb{R}$ and the expected delivery time, $\tau \in \mathbb{R}_+$. The essential riskiness index is defined as

$$\rho_\tau(\tilde{\xi}) = \min \left\{ \gamma \geq 0 \;\middle|\; \sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[\max \left\{ \tilde{\xi} - \tau, -\gamma \right\} \right] \leq 0 \right\}, \tag{6}$$

where $\min \emptyset = \infty$ by convention.

We briefly present some of the important properties as ERI.

i) **Monotonicity**: If $\mathbb{P}\left[\tilde{\xi}_1 \geq \tilde{\xi}_2\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$, then $\rho_\tau(\tilde{\xi}_1) \geq \rho_\tau(\tilde{\xi}_2)$;

ii) **Satisficing**: $\rho_\tau(\tilde{\xi}) = 0$ if and only if $\mathbb{P}\left[\tilde{\xi} \leq \tau\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$;

iii) **Infeasibility**: If $\sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}\right] > \tau$, then $\rho_\tau(\tilde{\xi}) = \infty$;

iv) **Convexity**: For all $\lambda \in [0,1]$, $\rho_\tau(\lambda\tilde{\xi}_1 + (1-\lambda)\tilde{\xi}_2) \leq \lambda\rho_\tau(\tilde{\xi}_1) + (1-\lambda)\rho_\tau(\tilde{\xi}_2)$;

v) **Delay bounds**:

$$\mathbb{P}\left[\tilde{\xi} - \tau > \rho_\tau(\tilde{\xi})\vartheta\right] \leq \frac{1}{1+\vartheta}, \quad \forall \vartheta > 0, \mathbb{P} \in \mathcal{F}.$$

Monotonicity ensures that an uncertain arrival time that dominates another would not be better off under ERI. *Satisficing* ensures that if the delivery time can always be met, then the risk of poor service under ERI is zero. The *infeasibility* property ensures that the average arriving time, even under distributional ambiguity, should not exceed the expected delivery time. The property of

convexity provides a more tractable formulation compared to non-convex ones, such as the criterion based on the probability of service delays. *Delay bounds* provide insights into the probability bounds of the event, namely that the arriving time exceeds the expected delivery time plus any duration. This property indicates that ERI ensures the probability of lateness reduces reciprocally as the magnitude of lateness increases in multiples of the index value. Specifically, when ϑ takes a larger value, although the magnitude of lateness increases, the probability bound $1/(1+\vartheta)$ for this event decreases. In this way, the ERI can account for both the probability of delay and its magnitude.

The drone delivery problem can be modeled as a multi-trip VRPTW proposed in Zhang et al. (2021), where the model's objective is to minimize the sum of the ERIs associated with service delay risks for all customers. Accordingly, to ensure high quality of services across all customers, the drone delivery system could also minimize the following joint decision criterion

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) \triangleq \sum_{i\in[N]} \rho_{\tau_i}(\tilde{\xi}_i), \tag{7}$$

where $\tilde{\xi}_i$ and τ_i are, respectively, the uncertain arriving time and expected delivery time for the ith customer. Note that the basic ERI (6) with a random scalar has been extended to the case with a random vector $\tilde{\boldsymbol{\xi}}$ in (7), representing the sum of ERIs from multiple customers. The benefit of this criterion in achieving high-quality solutions with reasonable computational effort has been demonstrated in Zhang et al. (2021). Moreover, the joint decision criterion satisfies the following salient properties:

i) **Monotonicity**: If $\mathbb{P}\left[\tilde{\boldsymbol{\xi}}_1 \geq \tilde{\boldsymbol{\xi}}_2\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$, then $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_1) \geq \varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_2)$;
ii) **Satisficing**: $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = 0$ if and only if $\mathbb{P}\left[\tilde{\boldsymbol{\xi}} \leq \boldsymbol{\tau}\right] = 1$, $\forall \mathbb{P} \in \mathcal{F}$;
iii) **Non-abandonment**: If there exists $i \in [N]$ such that $\sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}_i\right] > \tau_i$, then $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = \infty$;
iv) **Convexity**: For all $\lambda \in [0,1]$, $\varrho_{\boldsymbol{\tau}}(\lambda\tilde{\boldsymbol{\xi}}_1 + (1-\lambda)\tilde{\boldsymbol{\xi}}_2) \leq \lambda\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_1) + (1-\lambda)\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}_2)$.

Because we are using a nonstandard decision criterion, these properties serve as an axiomatic framework for which our decision criterion would be justified. In our opinion, these properties are reasonable for any joint decision criterion that evaluates overall customer service. In particular, the non-abandonment property ensures that any feasible solution for which the objective value is finite implies that the average arrival time at every customer's location would not exceed the expected delivery time. We are cognizant of other joint decision criteria that would also preserve the same properties, such as the one that takes the value of the highest ERI among all customers. However, from our numerical experience, minimizing the maximum ERI results in having an excess of multiple optimal solutions, influencing the quality of the solutions as well as the computing time. We will illustrate this in our computational studies.

While we can cast our problem as a robust VRPTW proposed in Zhang et al. (2021), the resulting model would not be suitable and applicable to our problem due to the following reasons. First, to minimize the joint decision criterion (7), a number of binary decision variables denoting the visiting sequence of customers must be introduced. Second, the robust model in Zhang et al. (2021) is static, while we consider an adaptive model in this work. Thus, we propose an alternative joint decision criterion,

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) \triangleq \sum_{d\in[D]:\mathcal{K}_d^1\neq\emptyset} \left(\max_{i\in\mathcal{K}_d^1} \left\{ \rho_{\bar{\tau}^1}\left(\tilde{\xi}_i\right) \right\} \right) + \sum_{d\in[D]:\mathcal{K}_d^2\neq\emptyset} \left(\max_{i\in\mathcal{K}_d^2} \left\{ \rho_{\bar{\tau}^2}\left(\tilde{\xi}_i\right) \right\} \right), \tag{8}$$

where $\mathcal{K}_d^t \subseteq [N]$ denotes the set of customers to be served by drone d, $d \in [D]$ in period t, $t \in [2]$.

The proposed criterion also satisfies the properties of monotonicity, satisficing, non-abandonment, and convexity. More importantly, it helps us construct a simpler, more computationally scalable formulation, as demonstrated in the following proposition.

PROPOSITION 1. *The joint decision criterion $\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}})$ is equivalent to*

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) = \sum_{d\in[D]:\mathcal{K}_d^1\neq\emptyset} \rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) + \sum_{d\in[D]:\mathcal{K}_d^2\neq\emptyset} \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right),$$

where ℓ_d^t denotes the last customer served by drone d, $d \in D$ in Period t, $t \in [2]$.

Observe that the random arrival time at the last customer by drone d, $d \in [D]$ in Period 1 is

$$\tilde{\xi}_{\ell_d^1} = \sum_{i\in\mathcal{K}_d^1\setminus\{\ell_d^1\}} (\tilde{u}_i^1 + \tilde{v}_i^1) + \tilde{u}_{\ell_d^1}^1. \tag{9}$$

Note that the arrival time in Period 2 is not an additive function of the flight times because customers in Period 2 can only be served after $\bar{\tau}^1$. We assume that whenever the anticipated arrival is before $\bar{\tau}^1$, the drone can delay its departure from the depot so that the arrival would be punctual at $\bar{\tau}^1$. We also need to account for the first customer to be served in Period 2 by drone d, $d \in [D]$, which we denote by $\underline{\ell}_d^2$. Hence, the random arrival time at the last customer by drone d, $d \in [D]$ in Period 2 is

$$\tilde{\xi}_{\ell_d^2} = \max\left\{ \sum_{i\in\mathcal{K}_d^1} (\tilde{u}_i^1 + \tilde{v}_i^1) + \tilde{u}_{\underline{\ell}_d^2}^2, \bar{\tau}^1 \right\} + \tilde{v}_{\underline{\ell}_d^2}^2 + \sum_{i\in\mathcal{K}_d^2\setminus\left\{\underline{\ell}_d^2,\ell_d^2\right\}} \left(\tilde{u}_i^2 + \tilde{v}_i^2\right) + \tilde{u}_{\ell_d^2}^2 \qquad \forall d \in [D]. \tag{10}$$

Under our proposed joint decision criterion, Proposition 1 implies that we only need to keep track of, for every drone, the last customer to be served in the morning, and the first and last customers served in the afternoon by the drone. In particular, the sequence of customers being served before the last customer in Period 1, and the sequence of customers being served between the first and the last customers in Period 2, would not affect the joint decision criterion. This greatly reduces the complexity of the model compared to a VRPTW formulation regarding the number of binary variables needed to model the problem.

Scheduling Decisions and Event-wise Adaptations

We denote $\bar{\mathcal{C}}^1 = \mathcal{C}^1 \cup \mathcal{C}^3$ and $\bar{\mathcal{C}}^2 = \mathcal{C}^2 \cup \mathcal{C}^3$ as the sets of customers who can expect their deliveries in the morning and afternoon, respectively. For scheduling decisions (y^1_{id}, x^1_{id}), $i \in \bar{\mathcal{C}}^1$, $d \in [D]$ associated with Period 1, we define the binary variable $y^1_{id} = 1$ if and only if customer i is the last customer visited by drone d in Period 1, *i.e.*, $i \in \mathcal{K}^1_d \cap \{\ell^1_d\}$, and the binary variable $x^1_{id} = 1$ if and only if $i \in \mathcal{K}^1_d \backslash \{\ell^1_d\}$.

As we have mentioned, the drone schedule in the afternoon can flexibly adapt to wind information, which is available by midday when the random scenario $\tilde{s}$ is realized. Hence, the decisions associated with Period 2 are functions of $s \in [S]$, which we denote by $y^2_{id}(s), z^2_{id}(s), x^2_{id}(s)$, $i \in \bar{\mathcal{C}}^2$, $d \in [D]$. For a given scenario $s \in [S]$, the binary variable $y^2_{id}(s) = 1$ if and only if customer i is the last customer visited by drone d in Period 2, *i.e.*, $i \in \mathcal{K}^2_d \cap \{\ell^2_d\}$, the binary variable $z^2_{id}(s) = 1$ if and only if customer i is the first customer visited by drone d in Period 2, *i.e.*, $i \in \mathcal{K}^2_d \cap \{\underline{\ell}^2_d\}$, and the binary variable $x^2_{id}(s) = 1$ if and only if $i \in \mathcal{K}^2_d \backslash \{\underline{\ell}^2_d, \ell^2_d\}$.

The adaptive scheduling decisions in Period 2 can be defined within a family of binary function maps. For flexibility in adaptation, we adopt the event-wise adaptation introduced in Chen et al. (2020). We first define an event $\mathcal{E} \subseteq [S]$ by a subset of scenarios. A partition of scenarios then induces a collection $\mathcal{S}$ of mutually exclusive and collectively exhaustive (MECE) events. Correspondingly, we define a mapping $f_{\mathcal{S}} : [S] \mapsto \mathcal{S}$ such that $f_{\mathcal{S}}(s) = \mathcal{E}$, for which $\mathcal{E}$ is the only event in $\mathcal{S}$ that contains the scenario s. Given a collection $\mathcal{S}$ of MECE events, we define the event-wise adaptation to characterize the scheduling decisions in Period 2 as follows:

$$\mathcal{A}(\mathcal{S}) \triangleq \left\{ \boldsymbol{x} : [S] \mapsto \{0,1\}^{\bar{\mathcal{C}}^2 \times D} \;\middle|\; \begin{array}{l} \boldsymbol{x}(s) = \boldsymbol{x}^{\mathcal{E}},\ \mathcal{E} = f_{\mathcal{S}}(s) \\ \text{for some } \boldsymbol{x}^{\mathcal{E}} \in \{0,1\}^{\bar{\mathcal{C}}^2 \times D} \end{array} \right\}.$$

In the case of a non-adaptive (or static) policy, we have $\mathcal{S} = \{[S]\}$, so that the scheduling decisions in Period 2 would not change its solutions in response to the outcomes of scenario $\tilde{s}$. For the case of full adaption, the collection $\mathcal{S}$ would have S different events, each containing an element of $[S]$.

The feasible drone delivery schedule is as follows:

$$\mathcal{X}=\left\{(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\;\middle|\;\begin{array}{ll}\sum_{d\in[D]}(x^1_{id}+y^1_{id})=1 & \forall i\in\mathcal{C}^1, \quad (11a)\\ \sum_{d\in[D]}\left(x^2_{id}(s)+y^2_{id}(s)+z^2_{id}(s)\right)=1 & \forall i\in\mathcal{C}^2, s\in[S], \quad (11b)\\ \sum_{d\in[D]}\left(x^1_{id}+y^1_{id}+x^2_{id}(s)+y^2_{id}(s)+z^2_{id}(s)\right)=1 & \forall i\in\mathcal{C}^3, s\in[S], \quad (11c)\\ \sum_{i\in\bar{\mathcal{C}}^1}y^1_{id}=1 & \forall d\in[D], \quad (11d)\\ \sum_{i\in\bar{\mathcal{C}}^2}y^2_{id}(s)=1 & \forall d\in[D], s\in[S], \quad (11e)\\ \sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(s)=1 & \forall d\in[D], s\in[S], \quad (11f)\\ \boldsymbol{x}^1,\boldsymbol{y}^1\in\{0,1\}^{\bar{\mathcal{C}}^1\times D},\ \boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2\in\mathcal{A}(\mathcal{S}). & \end{array}\right\}$$

Constraints (11a)–(11c) mean that customers in each set are respectively scheduled to be visited exactly once in their allowable delivery time slots. Constraints (11d)–(11e) impose that each drone at each period can only have one customer as the last served one. Constraints (11f) mean that each drone can only have one customer as the first visited customer in Period 2. The index s in the constraints suggests that the afternoon schedule should be feasible under any scenario.

An example is provided in Appendix B to show the construction of the cluster-wise ambiguity sets and the definition of the event-wise adaptations. Additionally, the schedule decision can be enhanced by the following lexicographic ordering constraints, which break the symmetry between drones (Adulyasak et al. 2014)

$$\sum_{i=1,i\in\mathcal{C}^1}^{j}2^{j-i}x^1_{id}\geq\sum_{i=1,i\in\mathcal{C}^1}^{j}2^{j-i}x^1_{id+1}\qquad\forall j\in\mathcal{C}^1,d\in[D-1]. \tag{12}$$

DRO Model

Now, for a given d, $d\in[D]$, the arrival time at the last customer in Period 1 can be written as the function

$$\zeta^1_d(\boldsymbol{x}^1,\boldsymbol{y}^1,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1)\triangleq\sum_{i\in\bar{\mathcal{C}}^1}\left(x^1_{id}(\tilde{u}^1_i+\tilde{v}^1_i)+y^1_{id}\tilde{u}^1_i\right), \tag{13}$$

and likewise, the arrival time at the last customer in Period 2 can be expressed as

$$\begin{aligned}&\zeta^2_d(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{s})\\ \triangleq\ &\max\left\{\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(\tilde{u}^1_i+\tilde{v}^1_i)+\sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(\tilde{s})\tilde{u}^2_i,\bar{\tau}^1\right\}\\ &+\sum_{i\in\bar{\mathcal{C}}^2}\left(z^2_{id}(\tilde{s})\tilde{v}^2_i+x^2_{id}(\tilde{s})(\tilde{u}^2_i+\tilde{v}^2_i)+y^2_{id}(\tilde{s})\tilde{u}^2_i\right).\end{aligned} \tag{14}$$

Note that these two functions are the explicit expressions of equations (9) and (10), respectively.

Recall that our goal is to minimize the risks of service delays under distributional ambiguity, where the risks are evaluated by the ERIs. Combining the definition of ERI in (6), the equivalent form of the joint decision criterion in Proposition 1, and the constraints (11) for defining a feasible drone delivery schedule, we construct the final DRO model as follows:

$$
\begin{aligned}
&\inf \sum_{d\in[D]} (\gamma_d^1 + \gamma_d^2) \\
&\text{s.t.} \sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[\max\left\{\zeta_d^1(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) - \bar{\tau}^1, -\gamma_d^1\right\}\right] \le 0 && \forall d\in[D], && \text{(15a)} \\
&\quad \sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[\max\left\{\zeta_d^2(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) - \bar{\tau}^2, -\gamma_d^2\right\}\right] \le 0 && \forall d\in[D], && \text{(15b)} \\
&\quad \gamma_d^1, \gamma_d^2 \ge 0 && \forall d\in[D], \\
&\quad (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X}.
\end{aligned}
$$

The DRO model (15) with cluster-wise moment-based ambiguity set and event-wise adaption fits into the *robust stochastic optimization* framework introduced by Chen et al. (2020). They also provide an algebraic modeling toolbox, *RSOME - Robust Stochastic Optimization Made Easy*, which facilitates rapid prototyping of our model. Specifically, by using affine recourse adaptation on a lifted ambiguity set, we can model the problem intuitively via RSOME. However, while it is convenient to do so, the toolbox generates numerous auxiliary variables and results in slow computational performance. We aim to eradicate all auxiliary variables and propose a B&C approach to solve our problem. We use RSOME as a reference to check the correctness of our approach.

4. Solving via Branch-and-Cut

For greater scalability, this section first presents a classical RO reformulation and then develops a B&C algorithm to solve the reformulation.

Reformulating the DRO Model

We first address the supremum over the ambiguity set in constraints (15a). Note that $\sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[\max\left\{\zeta_d^1(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) - \bar{\tau}^1, -\gamma_d^1\right\}\right]$ is equivalent to

$$
\sup_{\mathbb{P}\in\mathcal{G}^1} \mathbb{E}_{\mathbb{P}} \left[\max\left\{\zeta_d^1(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) - \bar{\tau}^1, -\gamma_d^1\right\}\right].
$$

Under the law of total probability, constraints (15a) can be reformulated as

$$
\sum_{k\in[K^1]} q_k \sup_{\mathbb{P}\in\mathcal{G}_k^1} \mathbb{E}_{\mathbb{P}} \left[\max\left\{\sum_{i\in\bar{\mathcal{C}}^1} \left(x_{id}^1(\tilde{u}_i^1 + \tilde{v}_i^1) + y_{id}^1 \tilde{u}_i^1\right) - \bar{\tau}^1, -\gamma_d^1\right\}\right] \le 0 \qquad \forall d\in[D],
$$

where $q_k = |\mathcal{L}_k|/H$,

$$\mathcal{G}_k^1 = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{4N}) \;\middle|\; \begin{array}{l} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{a}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{b}}^1) \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^1\right] = \boldsymbol{\mu}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1\right] = \boldsymbol{\nu}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{a}}^1\right] \leq \boldsymbol{\sigma}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^1\right] \leq \boldsymbol{\varsigma}_k^1 \\ \mathbb{P}\left[(\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{a}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{b}}^1) \in \Xi_k^1\right] = 1 \end{array} \right\}, \text{ and } \Xi_k^1 \triangleq \left\{ (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \mathbb{R}^{4N} \;\middle|\; \begin{array}{l} \underline{\boldsymbol{u}}_k^1 \leq \boldsymbol{u}^1 \leq \bar{\boldsymbol{u}}_k^1 \\ \boldsymbol{a}^1 \geq |\boldsymbol{u}^1 - \boldsymbol{\mu}_k^1| \\ \underline{\boldsymbol{v}}_k^1 \leq \boldsymbol{v}^1 \leq \bar{\boldsymbol{v}}_k^1 \\ \boldsymbol{b}^1 \geq |\boldsymbol{v}^1 - \boldsymbol{\nu}_k^1| \end{array} \right\}.$$

Note here that the *lifted ambiguity set* first proposed by Wiesemann et al. (2014) is adopted to keep the expectation constraints within the ambiguity set linear. We next tackle the supremum over set $\mathcal{G}_k^1$ and provide the result in the following theorem.

THEOREM 1. *The optimization problem*

$$\sup_{\mathbb{P} \in \mathcal{G}_k^1} \mathbb{E}_{\mathbb{P}} \left[\max \left\{ \sum_{i \in \bar{\mathcal{C}}^1} \left(x_{id}^1 (\tilde{u}_i^1 + \tilde{v}_i^1) + y_{id}^1 \tilde{u}_i^1 \right) - \bar{\tau}^1, -\gamma_d^1 \right\} \right] \tag{16}$$

is equivalent to

$$\max_{(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, p_1, p_2) \in \mathcal{Y}_{kd}^1} \left\{ \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 + \left(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 - \bar{\tau}^1 p_1 - \gamma_d^1 p_2 \right\}, \tag{17}$$

where

$$\mathcal{Y}_{kd}^1 = \left\{ (\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, p_1, p_2) \;\middle|\; \begin{array}{l} \exists \boldsymbol{a}_j^1, \boldsymbol{b}_j^1, j \in [2]: \\ \boldsymbol{u}_1^1 + \boldsymbol{u}_2^1 = \boldsymbol{\mu}_k^1, \quad \boldsymbol{v}_1^1 + \boldsymbol{v}_2^1 = \boldsymbol{\nu}_k^1, \\ \boldsymbol{a}_1^1 + \boldsymbol{a}_2^1 \leq \boldsymbol{\sigma}_k^1, \quad \boldsymbol{b}_1^1 + \boldsymbol{b}_2^1 \leq \boldsymbol{\varsigma}_k^1, \\ \underline{\boldsymbol{u}}_k^1 p_j \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 p_j, \quad \boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1 p_j| \quad \forall j \in [2], \\ \underline{\boldsymbol{v}}_k^1 p_j \leq \boldsymbol{v}_j^1 \leq \bar{\boldsymbol{v}}_k^1 p_j, \quad \boldsymbol{b}_j^1 \geq |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1 p_j| \quad \forall j \in [2], \\ p_1 + p_2 = 1, \quad p_1, p_2 \geq 0, \\ \boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{b}_j^1 \in \mathbb{R}^N \quad \forall j \in [2] \end{array} \right\}.$$

Until now, constraints (15a) have been transformed into a linear programming problem. Similarly, constraints (15b) can also be reformulated into an equivalent linear model. To keep the paper concise, we provide the reformulation process in Appendix C.3.

With Theorem 1 and the reformulation in Appendix C.3, we can now transform the DRO problem to a classical linear problem (see, for instance, Ben-Tal and Nemirovski 1998) as follows:

$$\inf \sum_{d \in [D]} (\gamma_d^1 + \gamma_d^2)$$

$$\text{s.t.} \sum_{k \in [K^1]} q_k \left(\Big(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \Big)' \boldsymbol{u}_1^{1k} + \Big(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \Big)' \boldsymbol{v}_1^{1k} - \bar{\tau}^1 p_1^k - \gamma_d^1 p_2^k \right) \leq 0$$

$$\forall(\boldsymbol{u}_1^{1k}, \boldsymbol{v}_1^{1k}, p_1^k, p_2^k) \in \mathcal{Y}_{kd}^1, k \in [K^1], \qquad \forall d \in [D], \tag{18a}$$

$$\sum_{(k,g,s)\in\mathcal{W}} q_{kgs} \Bigg(\Big(\sum_{i\in\bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)\boldsymbol{e}_i \Big)' (\boldsymbol{u}_1^{1kgs} + \boldsymbol{v}_1^{1kgs}) + \Big(\sum_{i\in\bar{\mathcal{C}}^2} (z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i \Big)' (\boldsymbol{u}_1^{2kgs} + \boldsymbol{v}_1^{2kgs})$$
$$+ \Big(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i \Big)' \boldsymbol{u}_1^{2kgs} - \bar{\tau}^2 p_1^{kgs} + \bar{\tau}^1 p_2^{kgs} + \Big(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i \Big)' \boldsymbol{v}_2^{2kgs} + \Big(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i \Big)' (\boldsymbol{u}_2^{2kgs} + \boldsymbol{v}_2^{2kgs})$$
$$+ \Big(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i \Big)' \boldsymbol{u}_2^{2kgs} - \bar{\tau}^2 p_2^{kgs} - \gamma_d^2 p_3^{kgs} \Bigg) \leq 0$$
$$\forall(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \in \mathcal{Y}_{kgsd}^2, (k,g,s) \in \mathcal{W}, \qquad \forall d \in [D], \tag{18b}$$

$$\gamma_d^1, \gamma_d^2 \geq 0 \qquad \forall d \in [D],$$

$$(\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X}.$$

Branch-and-Cut Algorithm

The B&C algorithm has been widely used to tackle mixed-integer linear programming problems, and it is an exact solution method that combines a cutting plane approach with a branch-and-bound (B&B) algorithm (Mitchell 2002). The B&C algorithm works by solving a linear programming (LP) relaxation problem at each node of the B&B tree. Each time when a fractional solution is obtained, violated cuts are generated, and the LP relaxation problem at the current node is resolved. If all the cuts are respected, and the solution still has fractional-valued integer variables, the algorithm selects a node to branch, generating two child nodes. If an integer solution is found and no violated cuts are detected, the algorithm updates the incumbent solution and prunes some nodes by comparing their LP relaxations' objective values with that of the incumbent solution. This process continues until all the B&B nodes are evaluated.

In our B&C scheme, constraints (18a) and (18b) are dynamically added, which has the advantage of keeping the number of decision variables small compared to the explicit formulation via RSOME. Specifically, we solve model (18) by ignoring these two groups of constraints. Whenever an integer solution is generated, we solve the equivalent maximum models presented in Theorem 1 and Appendix C.3. If the resulting solutions violate the constraints, cuts are generated and added to the model, and the model is resolved. This process continues until the optimality gap is satisfied.

5. Numerical Experiments

In this section, we first introduce the instance sets and benchmark approaches, and then conduct numerical tests to evaluate the proposed B&C algorithm and the cluster-wise DRO framework.

Instance Sets

We consider $N = 15$ and 20 customers, where 2 and 3 drones are used, respectively. We set drones' airspeed in both directions as $\bar{r}_0 = \bar{r}_i = 20\ m/s$, $i \in [N]$, based on Amazon Prime Air and Workhorse

HorseFly drones (used by UPS), which can fly at speeds up to 22.35 m/s (Lavars 2015, Straight 2018). For each customer i, $i \in [N]$, we generate a random number from the interval $[6000, 18000]$ to denote the distance d_i between the depot and the customer. This indicates that if the wind is neglected, it generally takes 10 to 30 minutes for a drone to perform a round trip. The distance data is based on the following reports: The Amazon Prime Air drone can fly 15 miles (in about 20 minutes at a ground speed of 20 m/s) to deliver packages (Lee 2019), and the HorseFly drone has a 30-minute flight time (Straight 2018). The angle $\phi_i, i \in [N]$ between customer i and the x-axis takes a random integer value between 0 and 360. We set $\mathcal{C}^1 = \{1, \ldots, 0.4N\}$, $\mathcal{C}^2 = \{0.4N + 1, \ldots, 0.8N\}$, and $\mathcal{C}^3 = \{0.8N + 1, \ldots, N\}$. Let $\bar{\tau}^1 = 4000$ and $\bar{\tau}^2 = 8000$. To mitigate the biases resulting from the data generation method, for a fixed number of customers, we generate 5 instances.

With respect to weather information, we use the wind data downloaded from the China National Meteorological Information Center (http://data.cma.cn/site/index.html). For each geographic region, the realized wind conditions of the last 7 days (including the current day) are available. Each region is divided into several subregions, where the average wind speed and degree are reported hourly, *i.e.*, 24 records per day. We utilize the wind data of Sichuan Province, a landlocked province in Southwest China, ranging from September 14^{th} to 19^{th}, 2019. The wind information is reported for 145 subregions; therefore, we have 3480 records per day. We consider every two hours as a time horizon, and the first hour is Period 1, and the second hour is Period 2. Thus, for each day, we can get 1740 wind samples. Note that for some missing data, we randomly generate continuous values based on the lower and upper bounds of the available data. We use the data of September 14^{th} as the (in-sample) training data, *i.e.*, $H = 1740$, and the data from September 15^{th} to 19^{th} as the (out-of-sample) test data, producing 8700 samples.

To construct the cluster-wise ambiguity set, we use the K-means clustering algorithm to partition the wind samples. We also tried other clustering algorithms, such as the hierarchical clustering (Murtagh and Contreras 2012), and our preliminary tests showed that the results were similar to those under the K-means clustering. Thus, we opt to use the K-means clustering as it is a widely adopted clustering algorithm and easy to implement. For the adaptive scheduling decisions in Period 2, we define each first-period cluster as an event. The B&C algorithm and all benchmark models are implemented in Python programming language using Gurobi 9.1.1 as the solver. All computations are executed on a macOS PC with an Apple M2 Pro chip and 16GB memory.

Benchmark Approaches

We evaluate our DRO framework by comparing its solutions with those generated by the four other benchmark models. The first one is the deterministic model maximizing the slack time. The second one maximizes the joint on-time service probability. The third one has the same objective function

as our DRO model, but it is implemented in a stochastic programming framework. The last one minimizes the maximum ERI.

Deterministic Model (DM). Since the considered problem is service-oriented, a natural objective is to construct a drone schedule that maximizes the slack time in a deterministic model.

$$
\begin{aligned}
\max\ & \sum_{d\in[D]}\sum_{t\in[2]} l_{dt} \\
\text{s.t.}\ & \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_i+v^1_i)+y^1_{id}u^1_i)\le \bar{\tau}^1-l_{d1} && \forall d\in[D],\\
& \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_i+v^1_i)+\sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_i+v^2_i)+y^2_{id}u^2_i)\le \bar{\tau}^2-l_{d2} && \forall d\in[D],\\
& \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_i+x^2_{id}(u^2_i+v^2_i)+y^2_{id}u^2_i)\le \bar{\tau}^2-l_{d2} && \forall d\in[D],\\
& l_{dt}\ge 0 && \forall d\in[D], t\in[2],\\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X}.
\end{aligned}
$$

The first three constraints calculate the slack time l_{dt} for drone d, $d\in[D]$ in period t, $t\in[2]$. Since the objective function only captures the magnitude of slack time and does not indicate its probability, if two solutions have the same magnitude of slack time, the one with a potential probability of 5% may have the same preference as the other one with a probability of 95%. In computational experiments, we set the flight times $\boldsymbol{u}^1$, $\boldsymbol{v}^1$, $\boldsymbol{u}^2$, and $\boldsymbol{v}^2$ as their sample means.

Maximizing On-time Probability (MOP). To improve service quality, decision-makers may prefer to penalize lateness for guaranteeing that drones can serve customers within stipulated time windows as well as possible. Thus, another benchmark method is to maximize the joint on-time service probability as follows:

$$
\begin{aligned}
&\max\ \mathbb{P}\left[\boldsymbol{\zeta}-\bar{\boldsymbol{\tau}}\le 0\right]\\
&\text{s.t.}\ \ (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
\tag{19}
$$

where $\boldsymbol{\zeta}=\begin{bmatrix}\zeta^1_d,\ldots,\zeta^1_D\\ \zeta^2_d,\ldots,\zeta^2_D\end{bmatrix}$ and $\bar{\boldsymbol{\tau}}=(\bar{\tau}^1,\bar{\tau}^2)'\boldsymbol{e}'$, $\boldsymbol{e}\in\mathbb{R}^D$. In contrast to the objective of the deterministic model, the objective of model (19) only captures the lateness probability and ignores the magnitude of lateness completely. Thus, if the lateness probabilities of two solutions are the same, the one with a lateness magnitude of 40 minutes may have the same preference as the one with a lateness of 1 minute. Moreover, as the objective to be maximized is not a concave function, we use an empirical

distribution approximation reformulation to solve model (19) as follows:

$$
\begin{aligned}
\max\ & \frac{1}{H}\sum_{h\in[H]} I_h \\
\text{s.t.}\ & \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_{ih}+v^1_{ih})+y^1_{id}u^1_{ih})-\bar{\tau}^1 \le (1-I_h)M^1_h && \forall d\in[D], h\in[H],\\
& \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_{ih}+v^1_{ih})+\sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 \le (1-I_h)M^2_h && \forall d\in[D], h\in[H],\\
& \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_{ih}+x^2_{id}(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 \le (1-I_h)M^2_h && \forall d\in[D], h\in[H],\\
& I_h\in\{0,1\} && \forall h\in[H],\\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
$$

where $u^t_{ih}\in\mathbb{R}$ and $v^t_{ih}\in\mathbb{R}$ are the travel times in Period t, $t\in[2]$ with respect to sample h, $h\in[H]$. Variable I_h equals 1 if and only if all drones at all periods serve customers on time under sample h, $h\in[H]$. Parameter M^t_h, $h\in[H]$, $t\in[2]$ is a sufficiently large number. We choose $M^1_h=\sum_{i\in\bar{\mathcal{C}}^1}(u^1_{ih}+v^1_{ih})$ and $M^2_h=\max\Big\{\sum_{i\in\bar{\mathcal{C}}^1}(u^1_{ih}+v^1_{ih}),\bar{\tau}^1\Big\}+\sum_{i\in\bar{\mathcal{C}}^2}(u^2_{ih}+v^2_{ih})$.

Minimizing ERI Using Empirical Distribution (ERI-E). For model (15), we assume the travel times follow an empirical distribution, thus the expectations in constraints (15a) and (15b) are now evaluated over a known distribution. The resulting model is

$$
\begin{aligned}
\inf\ & \sum_{d\in[D]}(\gamma^1_d+\gamma^2_d)\\
\text{s.t.}\ & \tfrac{1}{H}\sum_{h\in[H]} w^t_{dh}\le 0 && \forall d\in[D], t\in[2],\\
& w^1_{dh}\ge \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}(u^1_{ih}+v^1_{ih})+y^1_{id}u^1_{ih})-\bar{\tau}^1 && \forall d\in[D], h\in[H],\\
& w^2_{dh}\ge \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_{ih}+v^1_{ih})+\sum_{i\in\bar{\mathcal{C}}^2}((z^2_{id}+x^2_{id})(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 && \forall d\in[D], h\in[H],\\
& w^2_{dh}\ge \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}v^2_{ih}+x^2_{id}(u^2_{ih}+v^2_{ih})+y^2_{id}u^2_{ih})-\bar{\tau}^2 && \forall d\in[D], h\in[H],\\
& w^t_{dh}\ge -\gamma^t_d && \forall d\in[D], h\in[H], t\in[2],\\
& \gamma^t_d\ge 0 && \forall d\in[D], t\in[2],\\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
$$

where auxiliary variables $w^t_{dh}\in\mathbb{R}, d\in[D], t\in[2], h\in[H]$ are introduced to linearize the maximum terms in constraints (15a) and (15b).

Minimizing the Maximum ERI (M-ERI). Our joint decision criterion (8) minimizes the sum of the ERIs associated with each drone in each period. As mentioned in Section 3, other joint decision criteria that preserve salient properties also exist, such as minimizing the maximum ERI. Thus, we also consider this criterion and implement it using the empirical distribution. To do so, we replace the objective of the ERI-E model with

$$\min\gamma$$

and add the additional constraints

$$\gamma \geq \gamma_d^t \qquad \forall d \in [D], t \in [2].$$

Note that we can also implement this criterion using the DRO framework, which will be detailed in the subsequent experiments.

For all methods, we first derive the delivery decisions from training data, then evaluate their out-of-sample performance using the following four indicators, which are relevant to decision-makers in a service-oriented system:

- *MaxPro:* The maximum lateness probability across all customers.
- *SumPro:* The sum of lateness probabilities of all customers.
- *MaxExp:* The maximum expected lateness duration across all customers.
- *SumExp:* The sum of expected lateness durations of all customers.

When calculating the expected lateness, we assume the travel times follow an empirical distribution. Additionally, for the sequence of customers (except the last customer in Period 1 and the first and last customers in Period 2), we assume that drones will serve a customer first if the customer is nearer to the depot.

In the following sections, we first evaluate the performance of the B&C algorithm. We then progressively compare our DRO framework with the benchmark approaches. Specifically, we compare the ERI-E model with the DM, MOP, and M-ERI models to evaluate the joint decision criterion (8). The ERI-E and M-ERI models are subsequently compared with their corresponding DRO models to justify the robust method for our problem. Finally, we compare the adaptive DRO and the static DRO to justify the adaptive policy. A figure that summarizes the logic of our model comparisons is provided in Appendix D.1.

Evaluation of the B&C Algorithm

We compare the performance of the B&C algorithm with RSOME by solving the adaptive DRO model. The optimality gap for both methods is set to the default value in the Gurobi solver, which is 0.01%. Average results are reported in Table 1, where *time ratio* is obtained by using the computing time of the B&C algorithm to divide that of the RSOME. We conduct the comparison on instances with 15 customers, as RSOME fails to provide a solution within 6 hours for some instances with $N = 20$ when $K^1 \geq 2$.

Table 1 shows that the B&C algorithm can solve instances to optimality within a shorter time frame in most cases (10 out of 12). In 6 cases, the average CPU time of the B&C is less than half of the computing time of the RSOME. Therefore, we can conclude that the B&C algorithm can generally achieve greater scalability for the adaptive DRO model than RSOME.

Table 1 Performance comparison (CPU time in seconds) between B&C and RSOME on 15-customer instances

(K^1, K^2)	B&C	RSOME	Time ratio	(K^1, K^2)	B&C	RSOME	Time ratio
(1,1)	1.53	1.18	1.29	(3,1)	14.34	35.31	0.41
(1,2)	3.20	5.23	0.61	(3,2)	75.64	122.76	0.62
(1,3)	6.03	13.86	0.44	(3,3)	688.25	361.52	1.90
(2,1)	5.97	6.77	0.88	(4,1)	49.06	132.32	0.37
(2,2)	15.91	25.79	0.62	(4,2)	2395.28	5873.61	0.41
(2,3)	32.06	83.62	0.38	(5,1)	2156.23	4648.53	0.46

Table 2 Average results of out-of-sample tests under different decision criteria

	$N = 15$					$N = 20$				
Model	MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
DM	0.1835	0.2410	17.17	25.43	0.04	0.1303	0.1809	17.38	29.01	0.04
MOP	0.0362	0.0742	7.93	16.09	2217.70	0.0317	0.0894	7.21	20.49	3600.00
M-ERI	0.0360	0.0732	7.56	15.65	119.46	0.0326	0.1043	7.13	22.61	3600.00
ERI-E	0.0339	0.0668	7.57	14.81	25.82	0.0298	0.0850	7.09	19.86	2392.93

We further study the impact of parameters N, K^1, and K^2 on the CPU time of the B&C algorithm for solving the adaptive DRO model. We present the detailed experiment setting and results in Appendix D.2. The results show that the CPU time increases significantly when the number of customers increases, especially when it exceeds 20. The CPU time also increases with the numbers of clusters, K^1 and K^2. However, the value of K^1 has a more significant impact on the computing time than that of K^2. This is because more (adaptive) variables and constraints are required when K^1 increases. In particular, the numbers of constraints (11b), (11c), (11e), and (11f) increase with K^1, as these constraints should hold for each cluster.

Evaluation of the Joint Decision Criterion

In this section, we evaluate the performance of different decision criteria. For models MOP, M-ERI, and ERI-E, the optimality gap is set to 0.5%, and the time limit for solving each model is set to 3600 seconds. The average results of out-of-sample tests are reported in Table 2.

Table 2 shows that while Method DM consumes the least CPU time, it generates poor solutions, leading to significant out-of-sample lateness when uncertainty is present. Specifically, the lateness probability for the worst-case customer, *MaxPro*, is as much as 18.35% and 13.03% for $N = 15$ and 20, respectively; however, under other decision criteria, *MaxPro* is around 3%. Regarding other service indicators, Method DM also performs much worse. These comparison results underscore the importance of incorporating uncertainty.

Method ERI-E generally provides better out-of-sample performance than the three other methods. Specifically, it shows superiority in indicators *MaxPro*, *SumPro*, and *SumExp*. It also consumes much less CPU time than Methods M-ERI and MOP. Method M-ERI performs slightly better with regard to *MaxExp* when $N = 15$, which is consistent with its objective function; however, it performs worse in other service indicators and requires more CPU time than ERI-E. The comparison results in this section demonstrate the importance of choosing an appropriate decision criterion, which is expected to provide generally better performance on all the service indicators. Meanwhile, the resulting model can be solved in an acceptable time frame.

Table 3 Average out-of-sample performance of the static DRO model with the decision criterion (8)

	$N=15$					$N=20$				
(K^1,K^2)	MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
(1,1)	0.0375	0.0668	8.41	15.17	1.48	0.0361	0.0898	7.93	20.65	7.77
(1,2)	0.0344	0.0654	8.01	15.04	3.12	0.0357	0.0870	8.20	20.55	11.07
(1,3)	0.0331	0.0647	7.70	14.71	5.93	0.0328	0.0839	7.61	20.09	17.95
(2,1)	0.0341	0.0650	7.91	14.87	3.17	0.0314	0.0836	7.28	19.94	10.18
(2,2)	0.0337	0.0654	7.76	14.69	7.38	0.0309	0.0854	7.35	20.18	57.53
(2,3)	0.0350	0.0651	7.86	14.82	13.60	0.0324	0.0838	7.61	19.91	92.97
(3,1)	0.0353	0.0658	8.11	14.86	5.32	0.0345	0.0855	8.00	20.14	29.29
(3,2)	0.0333	0.0645	7.68	14.77	11.82	0.0301	0.0832	7.13	19.81	194.99
(3,3)	0.0325	0.0645	7.83	14.81	19.65	0.0295	0.0826	7.37	19.77	609.69
(4,1)	0.0333	0.0646	7.62	14.82	7.57	0.0306	0.0841	7.27	19.82	58.93
(4,2)	0.0341	0.0647	7.75	14.89	19.75	**0.0286**	**0.0819**	**6.88**	**19.73**	879.29
(5,1)	**0.0336**	**0.0646**	**7.48**	**14.60**	10.90	0.0302	0.0841	7.22	19.95	696.31

Performance of the Robust Method

In this section, we compare the out-of-sample performance of the ERI-E model with the static DRO model that minimizes the joint decision criterion (8), to justify the robust method for our drone delivery problem. Namely, these two models optimize the same decision criterion while implemented in different frameworks, *i.e.*, the empirical distribution against the robust optimization. Recall that in the static (non-adaptive) robust model, the delivery schedule in Period 2 remains the same for any realization of wind scenario $\tilde{\kappa}^1$. We solve the DRO model via the B&C algorithm, setting the optimality gap and time limit to be the same as the ERI-E model. The average results are reported in Table 3, with the outcomes highlighted in bold if the static DRO model provides better out-of-sample performance in all service indicators than the ERI-E model. Note that we also compare the M-ERO model with the static DRO model that minimizes the maximum ERI. To keep the paper concise, we present the results of these comparisons in Appendix D.3.

Table 3 shows that the DRO model can find solutions with better out-of-sample performance in all service indicators for $N=15$ when $(K^1,K^2)=(5,1)$ and for $N=20$ when $(K^1,K^2)=(4,2)$, compared to its empirical counterpart, ERI-E. Moreover, the DRO model consumes less CPU time in these two cases. In particular, when $N=20$, it consumes 879.29 seconds on average, whereas the ERI-E model requires 2392.93 seconds, as shown in Table 2. Similar observations are also obtained from the comparisons in Appendix D.3. We note that there may exist other clusters under which the robust model outperforms its empirical counterpart, whereas our main goal is to identify clusters that can achieve concurrent benefits in solution quality and scalability. Moreover, increasing clusters does not necessarily improve out-of-sample performance, which conversely increases the computing time (Perakis et al. 2023). Thus, we limit our exploration to a limited number of clusters, which suffice to discover good robust solutions within a reasonable time frame.

In addition, one may find that the robust method does not consistently exhibit superior out-of-sample performance compared to its empirical counterpart for certain clusters. This observation aligns with findings in Hao et al. (2020) (Figure 5 therein) and Perakis et al. (2023) (Table 1 and

Table 4 Average out-of-sample performance of the adaptive DRO model with the decision criterion (8)

	$N=15$					$N=20$				
(K^1,K^2)	MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
(2,1)	0.0353	0.0610	7.83	14.07	5.86	0.0342	0.0841	7.77	19.71	552.41
(2,2)	**0.0322**	**0.0597**	**7.46**	**13.78**	15.35	0.0322	0.0807	7.33	18.92	1672.85
(2,3)	**0.0316**	**0.0599**	**7.45**	**13.94**	31.68	**0.0285**	**0.0767**	**6.90**	**18.49**	2239.56
(3,1)	0.0367	0.0624	7.92	14.06	14.30	**0.0326**	**0.0772**	**7.62**	**18.11**	1976.44
(3,2)	**0.0315**	**0.0594**	**7.49**	**13.85**	65.64	0.0311	0.0768	7.38	17.95	3088.13
(3,3)	**0.0316**	**0.0596**	**7.45**	**13.75**	683.08	0.0301	0.0737	7.45	17.75	3600.00
(4,1)	0.0345	0.0622	7.74	14.24	28.63	0.0317	0.0764	7.47	18.11	3234.70
(4,2)	**0.0333**	**0.0600**	**7.32**	**13.74**	207.88	**0.0274**	**0.0744**	**6.58**	**17.46**	3600.00
(5,1)	**0.0320**	**0.0602**	**7.26**	**13.83**	64.17	**0.0264**	**0.0738**	**6.18**	**17.06**	3600.00

Figure 2 therein). We consider one reason to be the clustering algorithm. Clustering problems, like the K-means clustering in our study, are typically addressed through heuristic algorithms in machine learning libraries. Consequently, the resultant clusters lack a guarantee of optimality, potentially causing variability in the robust model's solutions. Moreover, when the empirical distributions in the training and testing sets are very similar, one can expect that the model based purely on the empirical distribution can perform reasonably well. Thus, we cannot statistically establish the superiority of the robust method over the empirical one. However, we can practically guarantee the effectiveness of the robust method by leveraging cross-validation techniques. To perform the cross-validation analysis, we can first divide the historical weather data into training and testing sets, enabling us to assess the out-of-sample performance of the robust model for specific clusters and compare it against the empirical model's performance. Based on these evaluations, we recommend cluster settings. To validate the recommendation, we employ a distinct customer data set and compare the out-of-sample performance of robust and empirical models. If the recommended clusters fail to yield satisfactory results, we can repeat the process above by gradually increasing the number of clusters. Note that these cross-validation steps are implemented in Appendix D.4 to verify the suggested clusters for our drone problem after finishing all the experiments in this section. Finally, we note that when $(K^1,K^2)=(H,1)$, the ambiguity set converges to the empirical distribution, *i.e.*, the static DRO model generalizes the empirical model, indicating that the robust method can always guarantee that its performance is not inferior to that of the empirical model.

Value of the Adaptive Policy

This section presents the out-of-sample performance of the adaptive DRO model with the joint decision criterion (8), as shown in Table 4. By comparing the results in Tables 3 and 4, we can gain insights regarding the effectiveness of the adaptive policy. Under the same clusters, if the adaptive model outperforms the static model in all service indicators, we highlight the corresponding results in bold. Note that the adaptive model reduces to the static model when $K^1=1$; thus, Table 4 does not display the results of the adaptive model for this particular case.

Table 4 shows that under the same clusters, in most cases, the adaptive model can find solutions with better out-of-sample performance in all service indicators, compared to the static model. When

Table 5 Detailed solutions of the DRO models for one instance

	Adaptive	Static
Delivery schedule in Period 1	[4, 1, 2, 3], [5, 15, 6]	[4, 1, 2, 6], [5, 15, 3]
	Delivery schedule in Period 2	
Centroids of first-period clusters	Adaptive	Static
(-0.76, 0.68)	[9, 8, 7, 14], [11, 13, 10, 12]	[13, 8, 11, 12], [10, 9, 7, 14]
(1.55, 0.15)	[11, 9, 14, 10], [7, 8, 13, 12]	
(-0.16, -1.23)	[7, 9, 8, 14], [10, 11, 13, 12]	

$N = 20$, under some clusters, even though the adaptive model is not solved to optimality, it can still generate solutions with better out-of-sample performance. For example, when $(K^1, K^2) = (5, 1)$, the adaptive model cannot be optimally solved within the 3600-second time limit; thus, suboptimal solutions are produced. In contrast, optimal solutions are generated for the static model, consuming 696.31 seconds on average. Even so, the average out-of-sample performance of the adaptive model is better than that of the static model. These results emphasize the importance of adaptability.

The previous section shows that the static DRO model performs better in all service indicators when $(K^1, K^2) = (5, 1)$ for $N = 15$, compared to the ERI-E model. Table 4 displays that the adaptive DRO model outperforms the ERI-E under multiple clusters, *e.g.*, $(K^1, K^2) = (2, 2)$, $(2, 3)$, and $(3, 2)$. More importantly, the performance of the adaptive model dominates that of the static model when $(K^1, K^2) = (5, 1)$. In other words, the static model performs better than the ERI-E model under $(K^1, K^2) = (5, 1)$, while the adaptive model performs even better than the static model. Likewise, for $N = 20$, when $(K^1, K^2) = (4, 2)$, the static DRO model delivers better out-of-sample performance than the ERI-E, while the performance of the adaptive model dominates that of the static model. Additionally, the adaptive DRO model outperforms the ERI-E model across multiple clusters, such as $(K^1, K^2) = (2, 3)$ and $(5, 1)$. These results indicate that with a small number of clusters, both robust models, particularly the adaptive one, outperform the empirical distribution.

Table 5 gives the detailed solutions of one instance under $(K^1, K^2) = (3, 2)$ for $N = 15$. The centroids of the first-period clusters can be easily obtained after applying the K-means clustering algorithm, available in many open-source machine learning libraries (*e.g.*, the scikit-learn library that we use). We can observe that under different clusters (or events), the adaptive DRO model generates different delivery schedules for Period 2; however, only one schedule is generated by the static DRO model. For any out-of-sample, when decision-makers have observed the wind realization in Period 1, they can calculate the distance between that realization and each cluster centroid to decide which cluster the observation belongs to, then adopt the corresponding delivery schedule in that cluster for Period 2.

Managerial Implications and Discussions

The numerical results in this section produce several managerial implications. First, it is essential to consider the uncertain travel times resulting from random weather conditions in drone delivery

problems. Otherwise, once uncertainty arises, the decisions provided by the deterministic model would lead to poor performance, causing significant delays in package delivery. Second, although there exist multiple service-oriented decision criteria, the modeler should consider at least two aspects when choosing a criterion: (i) the criterion's capability of accounting for both the probability of an undesirable event and its magnitude; (ii) the model's complexity (tractable or intractable) under the candidate decision criterion. Fortunately, our joint decision criterion based on the ERI offers a solution to the abovementioned two points. Third, utilizing realized first-period information to flexibly adjust the second-period decision can improve the DRO model's performance in out-of-sample tests even though the model may not be optimally solved. This observation underscores the importance of information utilization in decision-making. In general, with the cluster-wise ambiguity set, our proposed adaptive DRO model offers a practical decision-making tool for planners to schedule drone operations to reduce service lateness at customers.

Next, we discuss the choice of clusters. As indicated before, a practical way is to employ a cross-validation method and treat the numbers of clusters as hyperparameters in order to evaluate the solution performance based on different clusters on an out-of-sample basis (Perakis et al. 2023). When studying the joint pricing and production problem, Perakis et al. (2023) observe diminishing benefits with increasing clusters. Our numerical results in Table 4 support this observation. Thus, based on our empirical results, we recommend that decision-makers set the clusters to be $(K^1, K^2) = (2, 3)$ for the drone delivery problem, which is sufficient to ensure satisfactory performance. This suggestion is validated by additional numerical results in Appendix D.4.

Finally, we note that when observing the resolution process of the B&C algorithm for the adaptive DRO model, we find that, in many cases, the B&C algorithm can reduce the gap between the upper and lower bounds to be within 2% quickly. Most of the time is consumed to improve the lower bound to close the optimality gap. Thus, we increase the optimality gap threshold and analyze its impacts on solution performance and computing time. The detailed experiment settings and results are reported in Appendix D.5. The results show that slightly increasing the optimality gap threshold can significantly reduce the computing time without sacrificing solution quality too much. More importantly, the suboptimal solutions generated by the adaptive model still provide better out-of-sample performance than the static model with a smaller optimality gap threshold. Thus, in practical applications, decision-makers can consider sacrificing solution quality slightly to achieve greater scalability, striking a balance between solution performance and computing time.

6. Conclusions

This paper introduces a DRO model to solve a two-period drone scheduling problem with uncertain flight times, which can be implemented in a data-driven framework using historical weather

information. We propose a cluster-wise moment-based ambiguity set by partitioning the wind vector chart into different clusters, which allows us to adapt the delivery schedule in the afternoon to updated weather information available by midday. A branch-and-cut algorithm is developed to solve the adaptive robust model efficiently. To evaluate the proposed robust scheme, we benchmark our method against other classical models. Numerical results demonstrate that our robust framework, especially the adaptive robust model, can effectively reduce the service lateness at customers in out-of-sample tests. Since it is difficult to obtain analytical solutions for the case of a delivery network considered in this work, one potential future research direction is to derive analytical results that could be possible for a simple delivery network. Another direction is to develop powerful metaheuristic algorithms, such as tabu search (Glover 1990) and variable neighborhood search (Mladenović and Hansen 1997), within robust optimization frameworks (e.g., Li et al. (2023) and Zhang et al. (2021)) to achieve greater scalability with guaranteed out-of-sample performance.

Acknowledgments: We would like to extend our utmost appreciation and gratitude to Prof. Melvyn Sim from NUS Business School for his invaluable guidance and support. Without his assistance, this paper would have not been feasible. The authors gratefully acknowledge the valuable comments received from the Department Editor, Associate Editor, and anonymous referees on an earlier version of this article.

References

Adulyasak Y, Cordeau JF, Jans R (2014) Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26(1):103–120.

Agatz N, Bouman P, Schmidt M (2018) Optimization approaches for the traveling salesman problem with drone. *Transportation Science* 52(4):965–981.

Beck A, Ben-Tal A (2009) Duality in robust optimization: primal worst equals dual best. *Operations Research Letters* 37(1):1–6.

Ben-Tal A, Nemirovski A (1998) Robust convex optimization. *Mathematics of Operations Research* 23(4):769–805.

Bertsimas D, Gupta V, Kallus N (2018) Data-driven robust optimization. *Mathematical Programming* 167(2):235–292.

Bertsimas D, Sim M, Zhang M (2019) Adaptive distributionally robust optimization. *Management Science* 65(2):604–618.

Black T (2017) The future of drone delivery depends on predicting the weather. Accessed July 14, 2022, `https://mashable.com/2017/06/22/drone-delivery-weather/`.

Bocewicz G, Radzki G, Nielsen I, Witczak M, Zbigniew B (2020) Uavs fleet mission planning robust to changing weather conditions. *IFAC-PapersOnLine* 53(2):10518–10524.

Bouman P, Agatz N, Schmidt M (2018) Dynamic programming approaches for the traveling salesman problem with drone. *Networks* 72(4):528–542.

Chen Z, Sim M, Xiong P (2020) Robust stochastic optimization made easy with rsome. *Management Science* 66(8):3329–3339.

de Kemp MA, Mandjes M, Olver N (2021) Performance of the smallest-variance-first rule in appointment sequencing. *Operations Research* 69(6):1909–1935.

Delage E, Ye Y (2010) Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research* 58(3):595–612.

Denton B, Viapiano J, Vogl A (2007) Optimization of surgery sequencing and scheduling decisions under uncertainty. *Health Care Management Science* 10:13–24.

Doherty J (2019) Alphabet's Wing begins first commercial drone delivery service in U.S., beating Amazon, Uber. Accessed July 14, 2022, `https://www.newsweek.com/wing-drone-first-commercial-delivery-1466471`.

Dorling K, Heinrichs J, Messier GG, Magierowski S (2016) Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(1):70–85.

Enderle R (2019) The 5 most pressing problems with drone delivery. Accessed July 14, 2022, `https://www.technewsworld.com/story/86060.html`.

Gao X, Kong N, Griffin P (2023) Shortening emergency medical response time with joint operations of uncrewed aerial vehicles with ambulances. *Manufacturing & Service Operations Management* `https://doi.org/10.1287/msom.2022.0166`.

Glover F (1990) Tabu search: A tutorial. *Interfaces* 20(4):74–94.

Gupta V, Huang M, Rusmevichientong P (2022) Debiasing in-sample policy performance for small-data, large-scale optimization. *Operations Research* `https://doi.org/10.1287/opre.2022.2377`.

Ham AM (2018) Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies* 91:1–14.

Hao Z, He L, Hu Z, Jiang J (2020) Robust vehicle pre-allocation with uncertain covariates. *Production and Operations Management* 29(4):955–972.

Jaillet P, Qi J, Sim M (2016) Routing optimization under uncertainty. *Operations Research* 64(1):186–200.

Kim D, Lee K, Moon I (2019) Stochastic facility location model for drones considering uncertain flight distance. *Annals of Operations Research* 283(1):1283–1302.

Kim SJ, Lim GJ, Cho J (2018) Drone flight scheduling under uncertainty on battery duration and air temperature. *Computers & Industrial Engineering* 117:291–302.

Kim SJ, Lim GJ, Cho J, Côté MJ (2017) Drone-aided healthcare services for patients with chronic diseases in rural areas. *Journal of Intelligent & Robotic Systems* 88(1):163–180.

Lavars N (2015) Amazon to begin testing new delivery drones in the US. Accessed July 14, 2022, `https://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/`.

Lee D (2019) Amazon to deliver by drone 'within months'. Accessed July 14, 2022, `https://www.bbc.com/news/technology-48536319`.

Li R, Cui Z, Kuo YH, Zhang L (2023) Scenario-based distributionally robust optimization for the stochastic inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review* 176:103193.

Long DZ, Qi J (2014) Distributionally robust discrete optimization with entropic value-at-risk. *Operations Research Letters* 42(8):532–538.

Mak HY, Rong Y, Zhang J (2014) Sequencing appointments for service systems using inventory approximations. *Manufacturing & Service Operations Management* 16(2):251–262.

Mak HY, Rong Y, Zhang J (2015) Appointment scheduling with limited distributional information. *Management Science* 61(2):316–334.

Mancilla C, Storer R (2012) A sample average approximation approach to stochastic appointment sequencing and scheduling. *IIE Transactions* 44(8):655–670.

Mitchell JE (2002) Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization* 1(1):65–77.

Mladenović N, Hansen P (1997) Variable neighborhood search. *Computers & operations research* 24(11):1097–1100.

Mohajerin Esfahani P, Kuhn D (2018) Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming* 171(1-2):115–166.

Moshref-Javadi M, Winkenbach M (2021) Applications and research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications* 177:114854.

Murray CC, Chu AG (2015) The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies* 54:86–109.

Murtagh F, Contreras P (2012) Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2(1):86–97.

Nemirovski A, Shapiro A (2007) Convex approximations of chance constrained programs. *SIAM Journal on Optimization* 17(4):969–996.

Perakis G, Sim M, Tang Q, Xiong P (2023) Robust pricing and production with information partitioning and adaptation. *Management Science* 69(3):1398–1419.

Ponza A (2016) *Optimization of drone-assisted parcel delivery*. Master's thesis, University of Padova, Italy.

Rabta B, Wankmüller C, Reiner G (2018) A drone fleet model for last-mile distribution in disaster relief operations. *International Journal of Disaster Risk Reduction* 28:107–112.

Radzki G, Golinska-Dawson P, Bocewicz G, Banaszak Z (2021) Modelling robust delivery scenarios for a fleet of unmanned aerial vehicles in disaster relief missions. *Journal of Intelligent & Robotic Systems* 103:1–18.

Radzki G, Thibbotuwawa A, Bocewicz G (2019) Uavs flight routes optimization in changing weather conditions—constraint programming approach. *Applied Computer Science* 15(3):5–20.

Roberti R, Ruthmair M (2021) Exact methods for the traveling salesman problem with drone. *Transportation Science* 55(2):315–335.

Rose C (2013) Amazon's Jeff Bezos looks to the future. Accessed July 14, 2022, `https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/`.

Sacramento D, Pisinger D, Ropke S (2019) An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies* 102:289–315.

Smith JE, Winkler RL (2006) The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science* 52(3):311–322.

Straight B (2018) Workhorse now making residential deliveries with horsefly drone. Accessed July 14, 2022, `https://www.freightwaves.com/news/technology/drone-delivery-test-underway-in-cincinnati`.

Thibbotuwawa A, Bocewicz G, Zbigniew B, Nielsen P (2019) A solution approach for uav fleet mission planning in changing weather conditions. *Applied Sciences* 9(19):3972.

Vural D, Dell RF, Kose E (2019) Locating unmanned aircraft systems for multiple missions under different weather conditions. *Operational Research* 1–20.

Walker L (2014) Drone delivery for Amazon and Google slowed by headwinds. Accessed July 14, 2022, `https://www.newsweek.com/will-wind-be-end-commercial-drone-delivery-amazon-and-google-275999`.

Wiesemann W, Kuhn D, Sim M (2014) Distributionally robust convex optimization. *Operations Research* 62(6):1358–1376.

Yanıkoğlu İ, Gorissen BL, den Hertog D (2019) A survey of adjustable robust optimization. *European Journal of Operational Research* 277(3):799–813.

Zhang Y, Baldacci R, Sim M, Tang J (2019) Routing optimization with time windows under uncertainty. *Mathematical Programming* 175(1-2):263–305.

Zhang Y, Zhang Z, Lim A, Sim M (2021) Robust data-driven vehicle routing with time windows. *Operations Research* 69(2):469–485.

Online Supplement of "Robust Drone Delivery with Weather Information"

Appendix A Explicit Expression of the Ambiguity Set

The explicit form of the ambiguity set, $\mathcal{F}$, is as follows:

$$\mathcal{F} = \left\{ \mathbb{Q} \in \mathcal{P}(\mathbb{R}^{4N} \times [S]) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{s}) \sim \mathbb{Q} & \\ \exists \mathbb{P} \in \mathcal{P}(\mathbb{R}^{4N} \times \mathcal{W}): & \\ \quad (\tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1, \tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, (\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s})) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\mu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1 \mid \tilde{\kappa}^1 = k\right] = \boldsymbol{\nu}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^1 - \boldsymbol{\mu}_k^1| \mid \tilde{\kappa}^1 = k\right] \le \boldsymbol{\sigma}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^1 - \boldsymbol{\nu}_k^1| \mid \tilde{\kappa}^1 = k\right] \le \boldsymbol{\varsigma}_k^1 & \forall k \in [K^1] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2 \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] = \boldsymbol{\mu}_{sg}^2 & \forall s \in [S], g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2 \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] = \boldsymbol{\nu}_{sg}^2 & \forall s \in [S], g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2 - \boldsymbol{\mu}_{sg}^2| \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] \le \boldsymbol{\sigma}_{sg}^2 & \forall s \in [S], g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2 - \boldsymbol{\nu}_{sg}^2| \mid \tilde{s} = s, \tilde{\kappa}^2 = g\right] \le \boldsymbol{\varsigma}_{sg}^2 & \forall s \in [S], g \in [K^2] \\ \mathbb{P}\left[\begin{array}{l} \underline{\boldsymbol{u}}_k^1 \le \tilde{\boldsymbol{u}}^1 \le \bar{\boldsymbol{u}}_k^1 \\ \underline{\boldsymbol{v}}_k^1 \le \tilde{\boldsymbol{v}}^1 \le \bar{\boldsymbol{v}}_k^1 \\ \underline{\boldsymbol{u}}_{sg}^2 \le \tilde{\boldsymbol{u}}^2 \le \bar{\boldsymbol{u}}_{sg}^2 \\ \underline{\boldsymbol{v}}_{sg}^2 \le \tilde{\boldsymbol{v}}^2 \le \bar{\boldsymbol{v}}_{sg}^2 \end{array} \middle| \begin{array}{l} \tilde{\kappa}^1 = k, \\ \tilde{\kappa}^2 = g, \\ \tilde{s} = s \end{array} \right] = 1 & \forall (k,g,s) \in \mathcal{W} \\ \mathbb{P}\left[(\tilde{\kappa}^1, \tilde{\kappa}^2, \tilde{s}) = (k,g,s)\right] = q_{kgs} & \forall (k,g,s) \in \mathcal{W} \end{array} \right\}.$$

Appendix B Example for the Ambiguity Set and Adaptive Policy

We use Figure B1 to show the construction of the cluster-wise ambiguity sets and the definition of the event-wise adaptations. Suppose there are 280 samples, including only the wind conditions. To construct the ambiguity set associated with the random flight times in Period 1, we partition $(r_h^1, \theta_h^1)_{h\in[280]}$ into $K^1 = 4$ clusters. Since no other weather information is provided in the samples, we obtain $S = K^1 = 4$, $|\mathcal{V}_1| = 40$, $|\mathcal{V}_2| = 60$, $|\mathcal{V}_3| = 100$, and $|\mathcal{V}_4| = 80$. Subsequently, for each scenario $s \in [S]$, we partition the wind vectors $(r_h^2, \theta_h^2)_{h\in\mathcal{V}_s}$ into $K^2 = 2$ clusters, resulting in

$$|\mathcal{L}_{11}| = 25, \quad |\mathcal{L}_{12}| = 15, \quad |\mathcal{L}_{21}| = 20, \quad |\mathcal{L}_{22}| = 40,$$
$$|\mathcal{L}_{31}| = 60, \quad |\mathcal{L}_{32}| = 40, \quad |\mathcal{L}_{41}| = 33, \quad |\mathcal{L}_{42}| = 47.$$

For this example, we arbitrarily define two events as $\mathcal{S} = \{\{1,3\},\{2,4\}\}$. Then, under the event-wise adaptive policy, the drone schedule decisions in Period 2 should satisfy the following constraints

$$x_{id}^2(s) = x_{id}^2(s'), \quad y_{id}^2(s) = y_{id}^2(s'), \quad z_{id}^2(s) = z_{id}^2(s')$$

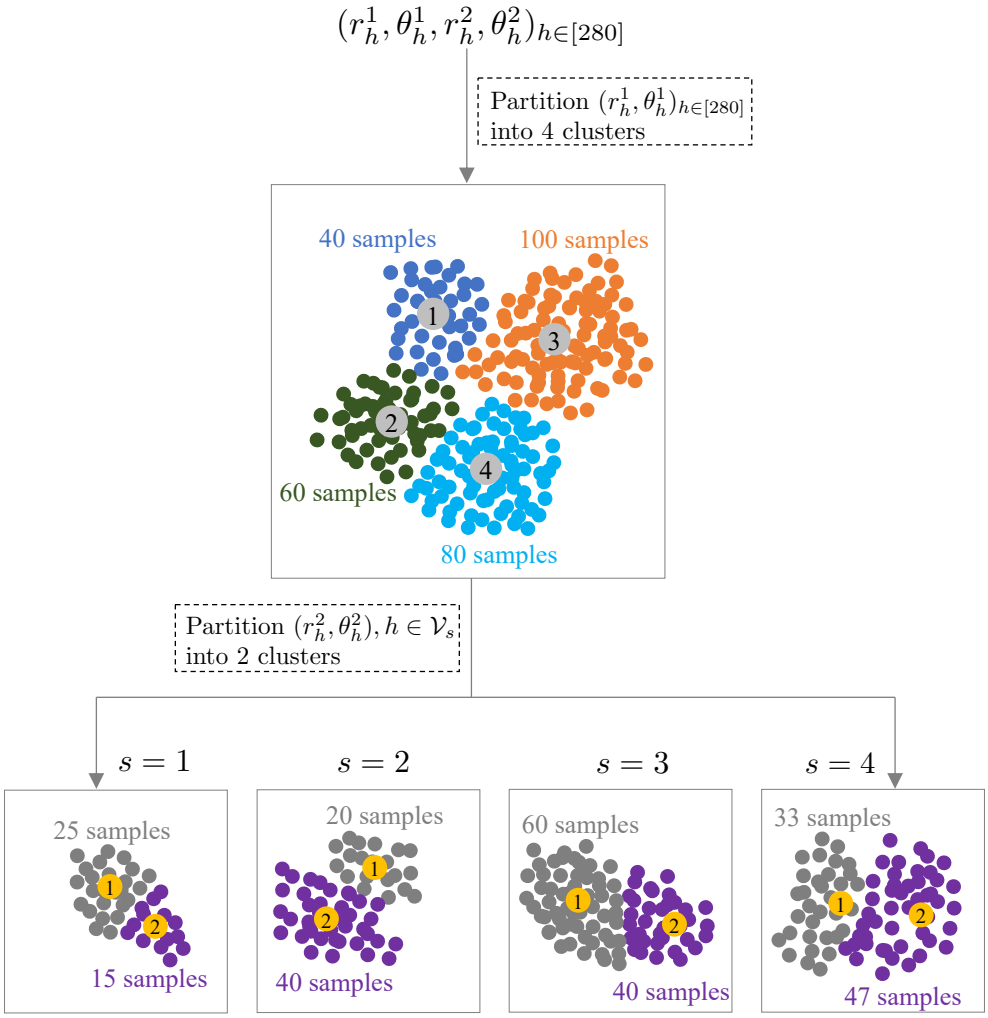


Figure B1 Partitioning wind samples to construct the cluster-wise ambiguity sets.

for $(s, s') = (1, 3)$ and $(s, s') = (2, 4)$.

Appendix C Proofs and Reformulation

C.1 Proof of Proposition 1

For each drone d, $d \in [D]$, in each period the arriving time at the last assigned customer is always larger than the arrival times at other assigned customers, *i.e.*,

$$\tilde{\xi}_{\ell_d^1} \geq \tilde{\xi}_i \qquad \forall i \in \mathcal{K}_d^1, \qquad\qquad \tilde{\xi}_{\ell_d^2} \geq \tilde{\xi}_i \qquad \forall i \in \mathcal{K}_d^2.$$

Based on the monotonicity property of $\varrho_\tau(\tilde{\xi})$, we have

$$\rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) \geq \rho_{\bar{\tau}^1}\left(\tilde{\xi}_i\right) \qquad \forall i \in \mathcal{K}_d^1, \qquad \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right) \geq \rho_{\bar{\tau}^2}\left(\tilde{\xi}_i\right) \qquad \forall i \in \mathcal{K}_d^2,$$

which indicate that

$$\rho_{\bar{\tau}^1}\left(\tilde{\xi}_{\ell_d^1}\right) = \max_{i\in\mathcal{K}_d^1}\left\{\rho_{\tau_1}(\tilde{\xi}_i)\right\} \quad \text{and} \quad \rho_{\bar{\tau}^2}\left(\tilde{\xi}_{\ell_d^2}\right) = \max_{i\in\mathcal{K}_d^2}\left\{\rho_{\tau_2}(\tilde{\xi}_i)\right\}.$$

C.2 Proof of Theorem 1

Because of the feasibility and the linearity of the ambiguity set (see, for instance, Bertsimas et al. 2019, Chen et al. 2020, Mohajerin Esfahani and Kuhn 2018), strong duality holds, thus we can reformulate Problem (16) as the following minimization problem, with $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \eta$ as the dual variables associated with the expectations and probability in $\mathcal{G}_k^1$:

$$\begin{aligned}
\min \quad & (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha} + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta} + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta} + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon} + \eta \\
\text{s.t.} \quad & (\boldsymbol{u}^1)'\boldsymbol{\alpha} + (\boldsymbol{v}^1)'\boldsymbol{\beta} + (\boldsymbol{a}^1)'\boldsymbol{\delta} + (\boldsymbol{b}^1)'\boldsymbol{\epsilon} + \eta \\
& \qquad \geq \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1(u_i^1 + v_i^1) + y_{id}^1 u_i^1) - \bar{\tau}^1 \qquad \forall (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \Xi_k^1, \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha} + (\boldsymbol{v}^1)'\boldsymbol{\beta} + (\boldsymbol{a}^1)'\boldsymbol{\delta} + (\boldsymbol{b}^1)'\boldsymbol{\epsilon} + \eta \geq -\gamma_d^1 \ \forall (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \Xi_k^1, \\
& \boldsymbol{\delta}, \boldsymbol{\epsilon} \geq \mathbf{0}, \\
& \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \in \mathbb{R}^N, \eta \in \mathbb{R}.
\end{aligned} \tag{C.1}$$

Using the duality result of RO, *i.e.*, the dual of the robust counterpart ('primal worst') is equal to the optimistic counterpart of the dual problem ('dual best') (Beck and Ben-Tal 2009), Problem (C.1) is equivalent to the following maximization problem:

$$\begin{aligned}
\max_{(\boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{b}_j^1) \in \Xi_k^1 \forall j \in [2]} \quad \min \quad & (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha} + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta} + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta} + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon} + \eta \\
\text{s.t.} \quad & (\boldsymbol{u}_1^1)'\boldsymbol{\alpha} + (\boldsymbol{v}_1^1)'\boldsymbol{\beta} + (\boldsymbol{a}_1^1)'\boldsymbol{\delta} + (\boldsymbol{b}_1^1)'\boldsymbol{\epsilon} + \eta \\
& \qquad \geq \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 + \left(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 - \bar{\tau}^1 \\
& (\boldsymbol{u}_2^1)'\boldsymbol{\alpha} + (\boldsymbol{v}_2^1)'\boldsymbol{\beta} + (\boldsymbol{a}_2^1)'\boldsymbol{\delta} + (\boldsymbol{b}_2^1)'\boldsymbol{\epsilon} + \eta \geq -\gamma_d^1 \\
& \boldsymbol{\delta}, \boldsymbol{\epsilon} \geq \mathbf{0}, \\
& \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\epsilon}, \in \mathbb{R}^N, \eta \in \mathbb{R}.
\end{aligned}$$

By duality of the inner linear optimization problem, we have, equivalently,

$$\begin{aligned}
\max \quad & \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 p_1 + \left(\sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 p_1 - \bar{\tau}^1 p_1 - \gamma_d^1 p_2 \\
\text{s.t.} \quad & \boldsymbol{u}_1^1 p_1 + \boldsymbol{u}_2^1 p_2 = \boldsymbol{\mu}_k^1 \\
& \boldsymbol{v}_1^1 p_1 + \boldsymbol{v}_2^1 p_2 = \boldsymbol{\nu}_k^1 \\
& \boldsymbol{a}_1^1 p_1 + \boldsymbol{a}_2^1 p_2 \leq \boldsymbol{\sigma}_k^1 \\
& \boldsymbol{b}_1^1 p_1 + \boldsymbol{b}_2^1 p_2 \leq \boldsymbol{\varsigma}_k^1 \\
& p_1 + p_2 = 1 \\
& \underline{\boldsymbol{u}}_k^1 \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 && \forall j \in [2] \\
& \boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [2]
\end{aligned}$$

$$\underline{\boldsymbol{v}}_k^1 \le \boldsymbol{v}_j^1 \le \bar{\boldsymbol{v}}_k^1 \qquad \forall j \in [2]$$
$$\boldsymbol{b}_j^1 \ge |\boldsymbol{v}_j^1 - \boldsymbol{\mu}_k^1| \qquad \forall j \in [2]$$
$$p_1, p_2 \ge 0$$
$$\boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{v}_j^1 \in \mathbb{R}^N \qquad \forall j \in [2].$$

By perspective transformation, *i.e.*, $\boldsymbol{u}_1^1 p_1 \to \boldsymbol{u}_1^1$, $\boldsymbol{v}_1^1 p_1 \to \boldsymbol{v}_1^1$, we obtain the linear optimization model presented in Equation (17), noting that the equivalence holds because the feasiblity $\mathcal{Y}_{kd}^1$ requires $\boldsymbol{u}_j^1 = \boldsymbol{a}_j^1 = \boldsymbol{v}_j^1 = \boldsymbol{b}_j^1 = \boldsymbol{0}$ whenever $p_j = 0$.

C.3 Reformulation of Constraints (15b)

Under the law of total probability, constraints (15b) is equivalent to

$$\sum_{(k,g,s)\in\mathcal{W}} q_{kgs} \sup_{\mathbb{P}\in\mathcal{F}_{kgs}} \mathbb{E}_{\mathbb{P}}\Bigg[\max\Bigg\{\max\Bigg\{\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)(\tilde{u}_i^1+\tilde{v}_i^1)+\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\tilde{u}_i^2, \bar{\tau}^1\Bigg\}$$
$$+\sum_{i\in\bar{\mathcal{C}}^2}\left(z_{id}^2(s)\tilde{v}_i^2 + x_{id}^2(s)(\tilde{u}_i^2+\tilde{v}_i^2) + y_{id}^2(s)\tilde{u}_i^2\right) - \bar{\tau}^2, -\gamma_d^2\Bigg\}\Bigg] \le 0 \qquad \forall d\in[D] \quad \text{(C.2)}$$

where

$$\mathcal{F}_{kgs} = \left\{ \mathbb{Q}\in\mathcal{P}(\mathbb{R}^{8N}) \;\middle|\; \begin{array}{l} (\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{a}}^1,\tilde{\boldsymbol{b}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{\boldsymbol{a}}^2,\tilde{\boldsymbol{b}}^2)\sim\mathbb{P} \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{u}}^1] = \boldsymbol{\mu}_k^1 \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{v}}^1] = \boldsymbol{\nu}_k^1 \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{a}}^1] \le \boldsymbol{\sigma}_k^1 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^1\right] \le \boldsymbol{\varsigma}_k^1 \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{u}}^2] = \boldsymbol{\mu}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{v}}^2] = \boldsymbol{\nu}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{a}}^2] \le \boldsymbol{\sigma}_{sg}^2 \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{b}}^2\right] \le \boldsymbol{\varsigma}_{sg}^2 \\ \mathbb{P}\left[(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{a}}^1,\tilde{\boldsymbol{b}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{\boldsymbol{a}}^2,\tilde{\boldsymbol{b}}^2)\in\Xi_{kgs}^2\right] = 1 \end{array} \right\},$$

and

$$\Xi_{kgs}^2 = \left\{ (\boldsymbol{u}^1,\boldsymbol{v}^1,\boldsymbol{a}^1,\boldsymbol{b}^1,\boldsymbol{u}^2,\boldsymbol{v}^2,\boldsymbol{a}^2,\boldsymbol{b}^2)\in\mathbb{R}^{8N} \;\middle|\; \begin{array}{ll} \underline{\boldsymbol{u}}_k^1 \le \boldsymbol{u}^1 \le \bar{\boldsymbol{u}}_k^1, & \underline{\boldsymbol{v}}_k^1 \le \boldsymbol{v}^1 \le \bar{\boldsymbol{v}}_k^1, \\ \boldsymbol{a}^1 \ge |\boldsymbol{u}^1 - \boldsymbol{\mu}_k^1|, & \boldsymbol{b}^1 \ge |\boldsymbol{v}^1 - \boldsymbol{\nu}_k^1|, \\ \underline{\boldsymbol{u}}_{sg}^2 \le \boldsymbol{u}^2 \le \bar{\boldsymbol{u}}_{sg}^2, & \underline{\boldsymbol{v}}_{sg}^2 \le \boldsymbol{v}^2 \le \bar{\boldsymbol{v}}_{sg}^2, \\ \boldsymbol{a}^2 \ge |\boldsymbol{u}^2 - \boldsymbol{\mu}_{sg}^2|, & \boldsymbol{b}^2 \ge |\boldsymbol{v}^2 - \boldsymbol{\nu}_{sg}^2| \end{array} \right\}.$$

Similar to the proof of Theorem 1, the supremum in Problem (C.2) is equivalent to the following minimization problem:

$$\min\ (\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta}^1 + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta}^1 + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{\mu}_{sg}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{\nu}_{sg}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{\sigma}_{sg}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{\varsigma}_{sg}^2)'\boldsymbol{\epsilon}^2 + \eta$$

$$
\begin{aligned}
\text{s.t. } & (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(u^1_i+v^1_i) + \sum_{i\in\bar{\mathcal{C}}^2}\left((z^2_{id}(s)+x^2_{id}(s))(u^2_i+v^2_i) + y^2_{id}(s)u^2_i\right) - \bar{\tau}^2 \\
& \qquad\qquad \forall(\boldsymbol{u}^1,\boldsymbol{v}^1,\boldsymbol{a}^1,\boldsymbol{b}^1,\boldsymbol{u}^2,\boldsymbol{v}^2,\boldsymbol{a}^2,\boldsymbol{b}^2)\in\Xi^2_{kgs} \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \bar{\tau}^1 + \sum_{i\in\bar{\mathcal{C}}^2}\left(z^2_{id}(s)v^2_i + x^2_{id}(s)(u^2_i+v^2_i) + y^2_{id}(s)u^2_i\right) - \bar{\tau}^2 \qquad \forall(\boldsymbol{u}^1,\boldsymbol{v}^1,\boldsymbol{a}^1,\boldsymbol{b}^1,\boldsymbol{u}^2,\boldsymbol{v}^2,\boldsymbol{a}^2,\boldsymbol{b}^2)\in\Xi^2_{kgs} \\
& (\boldsymbol{u}^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2)'\boldsymbol{\epsilon}^2 + \eta \geq -\gamma^2_d \\
& \qquad\qquad \forall(\boldsymbol{u}^1,\boldsymbol{v}^1,\boldsymbol{a}^1,\boldsymbol{b}^1,\boldsymbol{u}^2,\boldsymbol{v}^2,\boldsymbol{a}^2,\boldsymbol{b}^2)\in\Xi^2_{kgs} \\
& \boldsymbol{\delta}^n,\boldsymbol{\epsilon}^n \geq 0 \qquad\qquad \forall n\in[2] \\
& \boldsymbol{\alpha}^n,\boldsymbol{\beta}^n,\boldsymbol{\delta}^n,\boldsymbol{\epsilon}^n\in\mathbb{R}^N,\eta\in\mathbb{R} \qquad \forall n\in[2].
\end{aligned}
$$

Using the duality result of robust optimization we have, equivalently,

$$
\begin{aligned}
& \max_{(\boldsymbol{u}^1_j,\boldsymbol{v}^1_j,\boldsymbol{a}^1_j,\boldsymbol{b}^1_j,\boldsymbol{u}^2_j,\boldsymbol{v}^2_j,\boldsymbol{a}^2_j,\boldsymbol{b}^2_j)\in\Xi^2_{kgs}\,\forall j\in[3]} \min\ \Big((\boldsymbol{\mu}^1_k)'\boldsymbol{\alpha}^1 + (\boldsymbol{\nu}^1_k)'\boldsymbol{\beta}^1 + (\boldsymbol{\sigma}^1_k)'\boldsymbol{\delta}^1 + (\boldsymbol{\varsigma}^1_k)'\boldsymbol{\epsilon}^1 \\
& \qquad\qquad + (\boldsymbol{\mu}^2_{sg})'\boldsymbol{\alpha}^2 + (\boldsymbol{\nu}^2_{sg})'\boldsymbol{\beta}^2 + (\boldsymbol{\sigma}^2_{sg})'\boldsymbol{\delta}^2 + (\boldsymbol{\varsigma}^2_{sg})'\boldsymbol{\epsilon}^2 + \eta\Big) \\
\text{s.t. } & (\boldsymbol{u}^1_1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1_1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1_1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1_1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2_1)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2_1)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2_1)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2_1)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \left(\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})\boldsymbol{e}_i\right)'(\boldsymbol{u}^1_1+\boldsymbol{v}^1_1) + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}(s)+x^2_{id}(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}^2_1+\boldsymbol{v}^2_1) + \left(\sum_{i\in\bar{\mathcal{C}}^2}y^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{u}^2_1 - \bar{\tau}^2 \\
& (\boldsymbol{u}^1_2)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1_2)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1_2)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1_2)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2_2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2_2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2_2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2_2)'\boldsymbol{\epsilon}^2 + \eta \\
& \geq \bar{\tau}^1 + \left(\sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{v}^2_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2}x^2_{id}(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}^2_2+\boldsymbol{v}^2_2) + \left(\sum_{i\in\bar{\mathcal{C}}^2}y^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{u}^2_2 - \bar{\tau}^2 \\
& (\boldsymbol{u}^1_3)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1_3)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1_3)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1_3)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2_3)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2_3)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2_3)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2_3)'\boldsymbol{\epsilon}^2 + \eta \geq -\gamma^2_d \\
& \boldsymbol{\delta}^n,\boldsymbol{\epsilon}^n \geq 0 \qquad\qquad \forall n\in[2] \\
& \boldsymbol{\alpha}^n,\boldsymbol{\beta}^n,\boldsymbol{\delta}^n,\boldsymbol{\epsilon}^n\in\mathbb{R}^N,\eta\in\mathbb{R} \qquad \forall n\in[2].
\end{aligned}
$$

By duality of the inner optimization problem, we have, equivalently,

$$
\begin{aligned}
& \max \left(\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})\boldsymbol{e}_i\right)'(\boldsymbol{u}^1_1+\boldsymbol{v}^1_1)p_1 + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}(s)+x^2_{id}(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}^2_1+\boldsymbol{v}^2_1)p_1 \\
& \quad + \left(\sum_{i\in\bar{\mathcal{C}}^2}y^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{u}^2_1p_1 - \bar{\tau}^2p_1 + \bar{\tau}^1p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{v}^2_2p_2 \\
& \quad + \left(\sum_{i\in\bar{\mathcal{C}}^2}x^2_{id}(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}^2_2+\boldsymbol{v}^2_2)p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2}y^2_{id}(s)\boldsymbol{e}_i\right)'\boldsymbol{u}^2_2p_2 - \bar{\tau}^2p_2 - \gamma^2_dp_3 \\
& \text{s.t. } \boldsymbol{u}^1_1p_1 + \boldsymbol{u}^1_2p_2 + \boldsymbol{u}^1_3p_3 = \boldsymbol{\mu}^1_k \\
& \quad \boldsymbol{v}^1_1p_1 + \boldsymbol{v}^1_2p_2 + \boldsymbol{v}^1_3p_3 = \boldsymbol{\nu}^1_k
\end{aligned}
$$

$$
\begin{aligned}
&\boldsymbol{a}_1^1 p_1 + \boldsymbol{a}_2^1 p_2 + \boldsymbol{a}_3^1 p_3 \le \boldsymbol{\sigma}_k^1 \\
&\boldsymbol{b}_1^1 p_1 + \boldsymbol{b}_2^1 p_2 + \boldsymbol{b}_3^1 p_3 \le \boldsymbol{\varsigma}_k^1 \\
&\boldsymbol{u}_1^2 p_1 + \boldsymbol{u}_2^2 p_2 + \boldsymbol{u}_3^2 p_3 = \boldsymbol{\mu}_{sg}^2 \\
&\boldsymbol{v}_1^2 p_1 + \boldsymbol{v}_2^2 p_2 + \boldsymbol{v}_3^2 p_3 = \boldsymbol{\nu}_{sg}^2 \\
&\boldsymbol{a}_1^2 p_1 + \boldsymbol{a}_2^2 p_2 + \boldsymbol{a}_3^2 p_3 \le \boldsymbol{\sigma}_{sg}^2 \\
&\boldsymbol{b}_1^2 p_1 + \boldsymbol{b}_2^2 p_2 + \boldsymbol{b}_3^2 p_3 \le \boldsymbol{\varsigma}_{sg}^2 \\
&p_1 + p_2 + p_3 = 1 \\
&\boldsymbol{u}_k^1 \le \boldsymbol{u}_j^1 \le \bar{\boldsymbol{u}}_k^1 && \forall j \in [3] \\
&\underline{\boldsymbol{v}}_k^1 \le \boldsymbol{v}_j^1 \le \bar{\boldsymbol{v}}_k^1 && \forall j \in [3] \\
&\boldsymbol{a}_j^1 \ge |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [3] \\
&\boldsymbol{b}_j^1 \ge |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1| && \forall j \in [3] \\
&\underline{\boldsymbol{u}}_{sg}^2 \le \boldsymbol{u}_j^2 \le \bar{\boldsymbol{u}}_{sg}^2 && \forall j \in [3] \\
&\underline{\boldsymbol{v}}_{sg}^2 \le \boldsymbol{v}_j^2 \le \bar{\boldsymbol{v}}_{sg}^2 && \forall j \in [3] \\
&\boldsymbol{a}_j^2 \ge |\boldsymbol{u}_j^2 - \boldsymbol{\mu}_{sg}^2| && \forall j \in [3] \\
&\boldsymbol{b}_j^2 \ge |\boldsymbol{v}_j^2 - \boldsymbol{\nu}_{sg}^2| && \forall j \in [3] \\
&p_1, p_2, p_3 \ge 0
\end{aligned}
$$

By perspective transformation, it is equivalent to the following linear optimization problem

$$
\begin{aligned}
\max\ & \left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1 + y_{id}^1)\boldsymbol{e}_i\right)' (\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i\right)' (\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2) \\
&+ \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)' \boldsymbol{u}_1^2 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i\right)' \boldsymbol{v}_2^2 \\
&+ \left(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i\right)' (\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)' \boldsymbol{u}_2^2 - \bar{\tau}^2 p_2 - \gamma_d^2 p_3 \\
\text{s.t. } & \boldsymbol{u}_1^1 + \boldsymbol{u}_2^1 + \boldsymbol{u}_3^1 = \boldsymbol{\mu}_k^1 && \text{(C.3)} \\
& \boldsymbol{v}_1^1 + \boldsymbol{v}_2^1 + \boldsymbol{v}_3^1 = \boldsymbol{\nu}_k^1 && \text{(C.4)} \\
& \boldsymbol{a}_1^1 + \boldsymbol{a}_2^1 + \boldsymbol{a}_3^1 \le \boldsymbol{\sigma}_k^1 && \text{(C.5)} \\
& \boldsymbol{b}_1^1 + \boldsymbol{b}_2^1 + \boldsymbol{b}_3^1 \le \boldsymbol{\varsigma}_k^1 && \text{(C.6)} \\
& \boldsymbol{u}_1^2 + \boldsymbol{u}_2^2 + \boldsymbol{u}_3^2 = \boldsymbol{\mu}_{sg}^2 && \text{(C.7)} \\
& \boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 + \boldsymbol{v}_3^2 = \boldsymbol{\nu}_{sg}^2 && \text{(C.8)} \\
& \boldsymbol{a}_1^2 + \boldsymbol{a}_2^2 + \boldsymbol{a}_3^2 \le \boldsymbol{\sigma}_{sg}^2 && \text{(C.9)}
\end{aligned}
$$

$$\boldsymbol{b}_1^2 + \boldsymbol{b}_2^2 + \boldsymbol{b}_3^2 \leq \boldsymbol{\varsigma}_{sg}^2 \tag{C.10}$$

$$p_1 + p_2 + p_3 = 1 \tag{C.11}$$

$$\underline{\boldsymbol{u}}_k^1 p_j \leq \boldsymbol{u}_j^1 \leq \bar{\boldsymbol{u}}_k^1 p_j \qquad \forall j \in [3] \tag{C.12}$$

$$\underline{\boldsymbol{v}}_k^1 p_j \leq \boldsymbol{v}_j^1 \leq \bar{\boldsymbol{v}}_k^1 p_j \qquad \forall j \in [3] \tag{C.13}$$

$$\boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1 p_j| \qquad \forall j \in [3]$$

$$\boldsymbol{b}_j^1 \geq |\boldsymbol{v}_j^1 - \boldsymbol{\nu}_k^1 p_j| \qquad \forall j \in [3] \tag{C.14}$$

$$\underline{\boldsymbol{u}}_{sg}^2 p_j \leq \boldsymbol{u}_j^2 \leq \bar{\boldsymbol{u}}_{sg}^2 p_j \qquad \forall j \in [3] \tag{C.15}$$

$$\underline{\boldsymbol{v}}_{sg}^2 p_j \leq \boldsymbol{v}_j^2 \leq \bar{\boldsymbol{v}}_{sg}^2 p_j \qquad \forall j \in [3] \tag{C.16}$$

$$\boldsymbol{a}_j^2 \geq |\boldsymbol{u}_j^2 - \boldsymbol{\mu}_{sg}^2 p_j| \qquad \forall j \in [3] \tag{C.17}$$

$$\boldsymbol{b}_j^2 \geq |\boldsymbol{v}_j^2 - \boldsymbol{\nu}_{sg}^2 p_j| \qquad \forall j \in [3] \tag{C.18}$$

$$p_1, p_2, p_3 \geq 0. \tag{C.19}$$

Thus, the optimization problem,

$$\sup_{\mathbb{P} \in \mathcal{F}_{kgs}} \mathbb{E}_{\mathbb{P}} \Bigg[\max \Bigg\{ \max \Bigg\{ \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)(\tilde{u}_i^1 + \tilde{v}_i^1) + \sum_{i \in \bar{\mathcal{C}}^2} z_{id}^2(s) \tilde{u}_i^2, \bar{\tau}^1 \Bigg\} + \sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s)\tilde{v}_i^2 + x_{id}^2(s)(\tilde{u}_i^2 + \tilde{v}_i^2) + y_{id}^2(s)\tilde{u}_i^2) - \bar{\tau}^2, -\gamma_d^2 \Bigg\} \Bigg],$$

is equivalent to

$$\begin{aligned} \max_{(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \in \mathcal{Y}_{kgsd}^2} & \Bigg\{ \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s) + x_{id}^2(s)) \boldsymbol{e}_i \right)' (\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2) \\ & + \left(\sum_{i \in \bar{\mathcal{C}}^2} y_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^2 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 \\ & + \left(\sum_{i \in \bar{\mathcal{C}}^2} z_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{v}_2^2 + \left(\sum_{i \in \bar{\mathcal{C}}^2} x_{id}^2(s) \boldsymbol{e}_i \right)' (\boldsymbol{u}_2^2 + \boldsymbol{v}_2^2) \\ & + \left(\sum_{i \in \bar{\mathcal{C}}^2} y_{id}^2(s) \boldsymbol{e}_i \right)' \boldsymbol{u}_2^2 - \bar{\tau}^2 p_2 - \gamma_d^2 p_3 \Bigg\} \end{aligned}$$

where

$$\mathcal{Y}_{kgsd}^2 = \left\{ (\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, \boldsymbol{u}_1^2, \boldsymbol{v}_1^2, \boldsymbol{u}_2^2, \boldsymbol{v}_2^2, p_1, p_2, p_3) \;\middle|\; \begin{array}{l} \exists \boldsymbol{u}_2^1, \boldsymbol{v}_2^1, \boldsymbol{u}_3^1, \boldsymbol{v}_3^1, \boldsymbol{a}_j^1, \boldsymbol{b}_j^1, j \in [3] : \\ \text{Constraints (C.3)–(C.19)}, \\ \boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{b}_j^1, \boldsymbol{u}_j^2, \boldsymbol{a}_j^2, \boldsymbol{v}_j^2, \boldsymbol{b}_j^2 \in \mathbb{R}^N \;\forall j \in [3] \end{array} \right\}.$$

Appendix D Supplementary Descriptions and Results of Numerical Tests

D.1 Logic of Model Comparisons

Figure D1 summarizes the logic of our model comparisons in the numerical experiments.

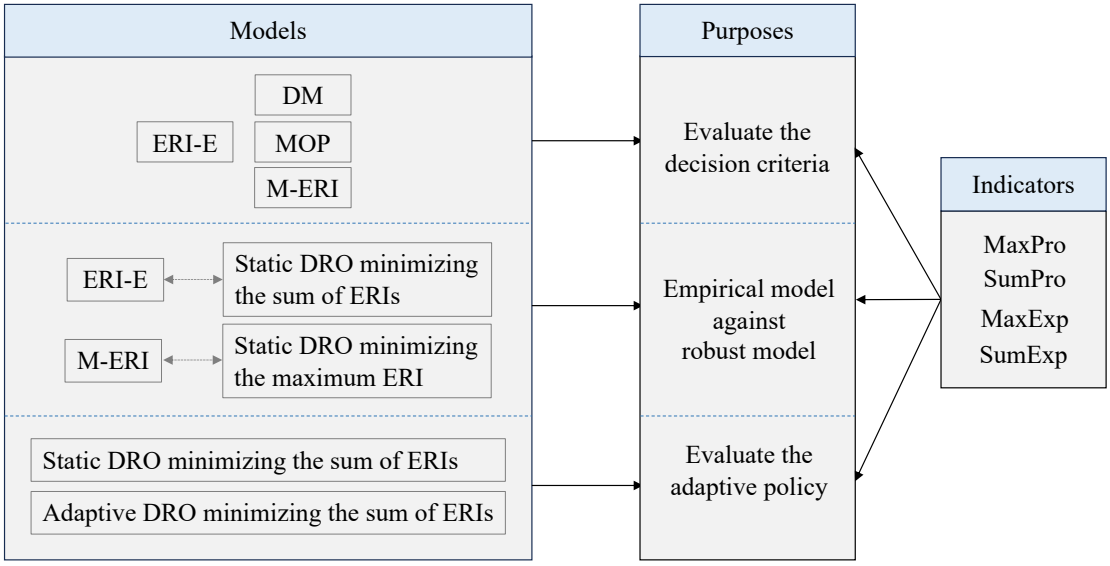


Figure D1 The logic of our model comparisons in the numerical experiments.

D.2 Impacts of Parameters on Computing Time

This section explores the impacts of parameters N, K^1, and K^2 on the adaptive DRO model's computing time. The optimality gap is set to 0.5%, and no time limit is imposed.

To study the impact of N, four combinations of clusters, *i.e.*, $(K^1, K^2) = (1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$, are selected. We further generate instances of 10 and 25 customers, where 2 and 4 drones are used, respectively. We tried to generate 30-customer instances; however, the gap between the upper and lower bounds was up to 8% after 17 hours, even though the numbers of clusters were set to small values, like $(K^1, K^2) = (2,1)$. Figure D2 shows that the computing time increases quickly with N, especially when it increases from 20 to 25.

Figure D3 shows that the computing time generally increases with the number of clusters. However, the value of K^1 has a more significant impact than that of K^2. For example, when K^2 is fixed to 2 and K^1 increases from 1 to 4, the CPU time increases from 3.12 seconds to 207.88 seconds. However, when K^1 is fixed to 2 and K^2 varies from 1 to 4, the CPU time just increases from 5.86 seconds to 33.12 seconds. This comparison result can be explained by the fact that more (adaptive) variables and constraints are required when K^1 increases. In particular, the number of constraints (11b), (11c), (11e), and (11f) increases, as these constraints should hold for each cluster (recall that we have $K^1 = S$ and define each first-period cluster as an event in our experiments).

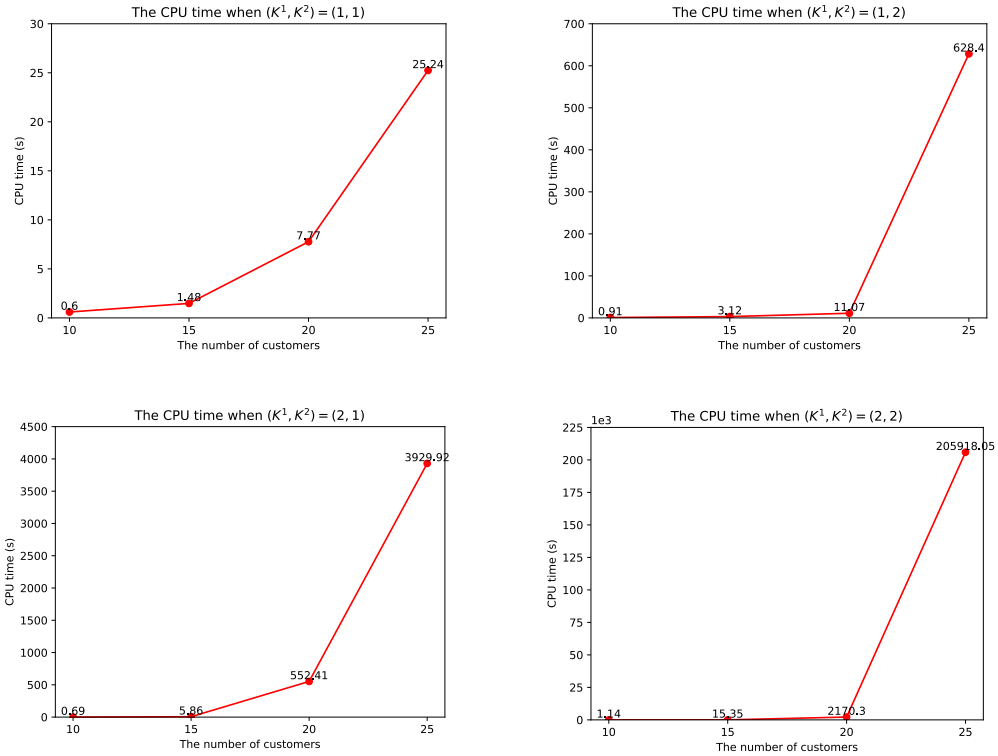


Figure D2 The impact of the number of customers on the adaptive DRO model's computing time

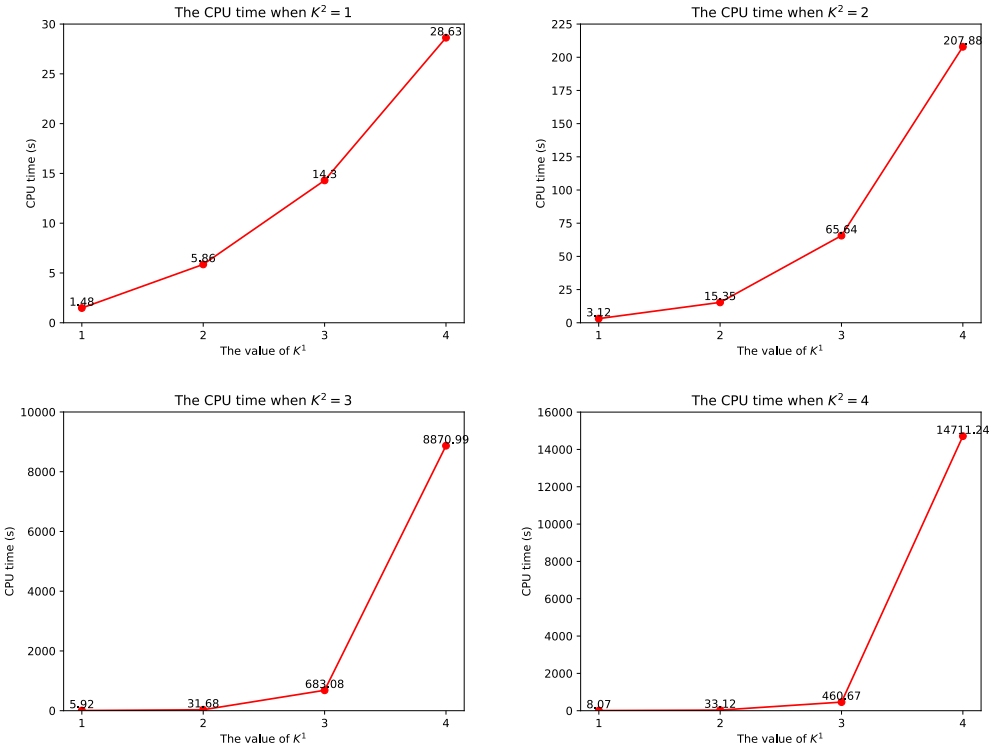
D.3 Supplementary Results of the Robust Method

This section compares the M-ERI model and the static DRO model that minimizes the maximum ERI (M-ERI-RO for short). The average results of the M-ERI-RO model in out-of-sample tests are reported in Table D1, where we also present the results of the M-ERI model in the last row for easy comparison. We mark the outcomes in bold if the M-ERI-RO model provides better out-of-sample performance in all service indicators than the M-ERI model.

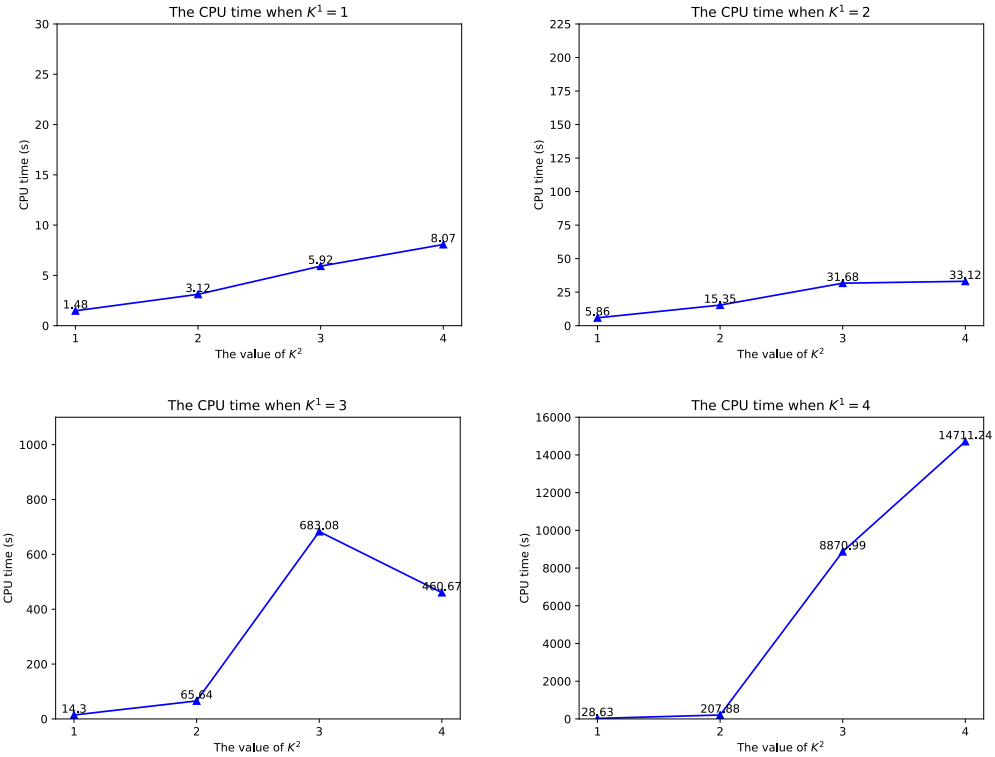
Table D1 Average out-of-sample performance of the DRO model minimizing the maximum ERI

		$N = 15$					$N = 20$				
Model	(K^1, K^2)	MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
M-ERI-RO	(1,1)	**0.0335**	**0.0712**	**7.44**	**15.43**	1.59	0.0388	0.1021	7.47	22.10	4.97
	(1,2)	0.0360	0.0675	8.13	15.11	3.66	0.0320	0.0880	7.50	20.73	1080.52
	(1,3)	0.0337	0.0656	7.70	14.98	4.81	0.0312	0.0832	7.49	20.00	737.96
	(2,1)	0.0369	0.0712	7.86	15.60	3.14	0.0377	0.1311	7.14	26.28	515.78
	(2,2)	0.0332	0.0705	7.56	15.42	6.72	0.0483	0.1148	8.02	23.60	1849.69
	(2,3)	0.0334	0.0664	7.58	15.14	11.10	**0.0320**	**0.0870**	**7.06**	**20.30**	1138.50
	(3,1)	0.0345	0.0791	7.85	16.44	5.21	0.0787	0.1974	8.79	31.41	361.56
	(3,2)	0.0420	0.0828	7.70	16.65	12.41	0.0619	0.1594	8.83	28.19	2544.44
	(3,3)	0.0340	0.0704	7.84	15.50	16.31	0.0310	0.0905	7.50	21.02	2366.00
	(4,1)	0.0869	0.1371	9.97	20.69	7.93	0.0991	0.2160	10.16	33.02	741.30
	(4,2)	0.0542	0.0971	8.47	17.76	13.75	0.0474	0.1476	7.61	27.46	2221.46
	(5,1)	0.0500	0.0970	7.97	17.72	8.96	0.0677	0.1856	8.09	30.54	1661.02
M-ERI		0.0360	0.0732	7.56	15.65	119.46	0.0326	0.1043	7.13	22.61	3600.00

Table D1 shows that the M-ERI-RO model provides better out-of-sample performance in all service indicators for $N = 15$ when $(K^1, K^2) = (1, 1)$ and for $N = 20$ when $(K^1, K^2) = (2, 3)$, compared



(a) The impact of parameter K^1 on computing time



(b) The impact of parameter K^2 on computing time

Figure D3 The impact of the numbers of clusters on the adaptive DRO model's computing time when $N = 15$

to its empirical counterpart, M-ERI. These results highlight the effectiveness of the robust method in mitigating service lateness for our drone delivery problem.

D.4 Experiments for Verifying the Suggested Numbers of Clusters

In this section, we conduct numerical tests to verify the suggested choice of clusters by randomly generating 50 instances for $N = 15$ and 20, respectively. We set $(K^1, K^2) = (2, 3)$ and other parameters be the same as in the main paper. The statistical results are provided in Table D2, where row *Std* reports the standard deviations and row *90th* (*95th*) provides the 90th (95th) percentile values. We highlight the results in bold if the adaptive model performs better in a service indicator.

Table D2 Average out-of-sample performance of the DRO models when $(K^1, K^2) = (2, 3)$

		Static DRO					Adaptive DRO				
N		MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
15	Average	0.0362	0.0636	6.88	12.45	11.62	**0.0333**	**0.0606**	**6.52**	**11.99**	28.50
	Std	0.0393	0.0667	5.07	9.28	3.83	**0.0366**	**0.0634**	**4.89**	**8.88**	14.96
	90th	0.1081	0.1858	16.06	27.71	17.83	**0.0728**	**0.1582**	**12.23**	**24.69**	52.21
	95th	0.1281	0.2196	18.18	32.99	19.55	**0.1272**	**0.2145**	**17.73**	**31.81**	65.79
20	Average	0.0127	0.0319	3.08	7.66	453.44	**0.0117**	**0.0295**	**2.81**	**7.15**	2739.60
	Std	0.0183	0.0486	3.35	9.12	733.16	**0.0165**	**0.0435**	**3.08**	**8.28**	1331.19
	90th	0.0335	0.0945	8.08	21.75	928.34	**0.0310**	**0.0833**	**7.26**	**18.92**	3600.00
	95th	0.0558	0.1434	11.05	29.37	2534.18	**0.0504**	**0.1300**	**10.07**	**27.24**	3600.00

Table D2 reveals that the adaptive DRO model consistently outperforms the static DRO model across all service indicators in the out-of-sample analysis. Moreover, the adaptive model exhibits lower 90th and 95th percentile values. The variations of the adaptive model's performance, *i.e.*, the standard deviations, are also smaller. Based on these observations, we can conclude that the suggested numbers of clusters, $(K^1, K^2) = (2, 3)$, adequately ensures excellent performance for our drone delivery problem.

D.5 Impacts of the Optimality Gap Threshold

In this section, we analyze the impacts of the optimality gap threshold (denoted by ϵ here for convenience) set in the solver on solution quality and computing time, using the 50 instances with $N = 20$ generated in Section D.4. We solve the adaptive DRO model and set ϵ to 1.5% and 2.0%, respectively. Table D3 reports the average out-of-sample performance.

Table D3 Average out-of-sample performance of the adaptive DRO under different optimality gap thresholds

	1.5% optimality gap					2.0% optimality gap				
	MaxPro	SumPro	MaxExp	SumExp	CPU time	MaxPro	SumPro	MaxExp	SumExp	CPU time
Average	0.0120	0.0299	2.91	7.27	1319.56	0.0121	0.0304	2.88	7.33	1015.18
Std	0.0166	0.0434	3.13	8.31	1510.87	0.0164	0.0435	3.07	8.30	1395.81
90th	0.0331	0.0831	7.56	18.93	3600.00	0.0299	0.0833	7.23	18.93	3600.00
95th	0.0504	0.1300	10.07	27.24	3600.00	0.0504	0.1300	10.07	27.24	3600.00

Table D3 shows that the average CPU time is 1319.56 seconds when $\epsilon = 1.5\%$, only half of that under $\epsilon = 0.5\%$. The CUP time is further reduced to 1015.18 seconds when increasing ϵ to 2.0%. Importantly, the adaptive model's solutions under these two cases still outperform those of the static model as presented in Table D2, even though the optimality gap threshold is set to 0.5% for the static model. Notably, among the three settings, the solutions under $\epsilon = 0.5\%$ perform the best, followed by the case of $\epsilon = 1.5\%$. Thus, in practical applications, decision-makers may consider sacrificing solution quality slightly to achieve greater scalability, striking a balance between solution performance and computing time.

Additionally, since the suboptimal solutions provided by the adaptive model still exhibit good out-of-sample performance, powerful metaheuristic algorithms, such as tabu search (Glover 1990) and variable neighborhood search (Mladenović and Hansen 1997), can be developed within robust optimization frameworks (e.g., Li et al. (2023) and Zhang et al. (2021)) to achieve greater scalability with guaranteed out-of-sample performance. Nevertheless, developing metaheuristic algorithms lies outside the scope of the current paper, as our main focuses are to model drones' uncertain travel times using historical data, propose an index to capture the risk of tardy delivery, and enhance the performance of solutions utilizing realized weather information. Thus, we present the development of metaheuristics as a future research direction.