# Robust Drone Delivery with Weather Information

Chun Cheng
Institute of Supply Chain Analytics, Dongbei University of Finance and Economics, Dalian, 116025, Liaoning, China
chun.cheng@polymtl.ca

Yossiri Adulyasak
HEC Montréal and GERAD, Montréal, H3T 2A7, Canada
yossiri.adulyasak@hec.ca

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

Melvyn Sim
Department of Analytics & Operations, NUS Business School, National University of Singapore
dscsimm@nus.edu.sg

Drone delivery has garnered significant attention recently due to its potential for faster delivery at lower cost relative to 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. 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 provision for different delivery schedules in the afternoon that adapt to updated weather information available by midday. Our approach minimizes the essential riskiness index (Zhang et al. 2019), which can limit 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 to the empirical distribution. The cluster-wise ambiguity set enables us to adapt the intraday delivery schedule depending on which cluster on the wind vector chart the observed morning wind vector belongs to. A branch-and-cut algorithm is developed to solve the adaptive distributionally robust optimization model.

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

*History*: July 14, 2020.

## 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). The 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 such as 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). Mehra (2015) and Enderle (2019) suggest that delays caused by weather conditions should be overcome before drone-based order delivery becomes a reality. Black (2017) further state 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 which 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 wind condition 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 $(r,\theta)$, where $r$ denotes the speed of the wind, and $\theta$ 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 $(d_i, \phi_i)$, where $d_i$ is the distance from the depot.

The launch speed of drones in the forward direction (*i.e.*, from the depot to customers), which depends on the weight and size of the customer's payload, is $\bar{r}_i$. In the absence of a payload, the launch speed of drones in the backward direction is $\bar{r}_0$, $\bar{r}_0 \geq \bar{r}_i$, $i \in [N]$. 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 condition, to offset the wind's influence on flight direction. We can derive the nominal forward flight time $u_i$ and the backward flight time $v_i$ as

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

Note that since $r < \min_{i\in[N]} \bar{r}_i$, the launch speed of the drone has a positive influence on flight duration. While a drone may not increase its launch speed, it 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 $\tau$ 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 condition cannot be perfectly predicted and the actual flight times may deviate from the nominal values determined solely by the wind speed. To account for this uncertainty, for a given wind vector, $(r,\theta)$, the forward and backward flight times would be uncertain and given by

$$\tilde{u}_i = u_i(r,\theta)\tilde{\omega}_i^u \qquad \tilde{v}_i = v_i(r,\theta)\tilde{\omega}_i^v$$

for some random factors $\tilde{\omega}_i^u, \tilde{\omega}_i^v$, each with support $[\underline{\omega}, \bar{\omega}]$, mean $\hat{\omega}$, and identical marginal distribution denoted by $\mathbb{P}_\omega \in \mathcal{P}(\mathbb{R})$.

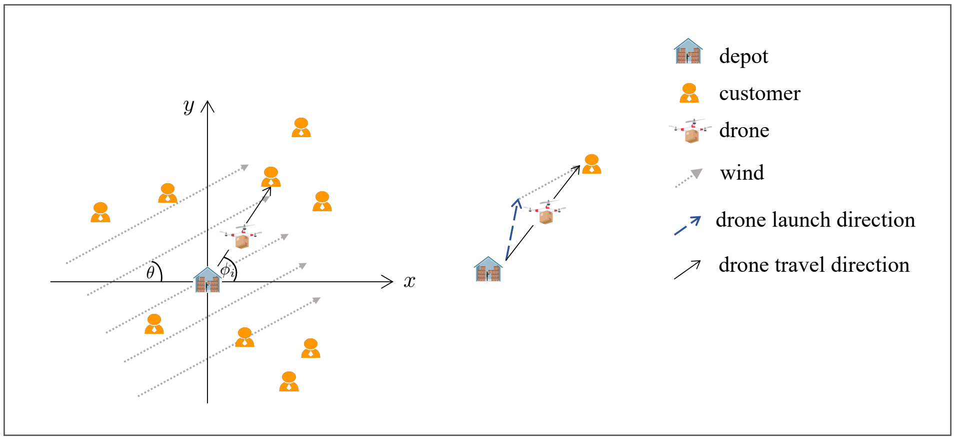

**Figure 1** Illustration for the calculation of flight times. The right part is based on vector addition in Physics.

Weather data, such as historical wind observations, are commonly available from the national or local meteorological information center, and can be used as predictors of 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 direction, $(r_h, \theta_h)$, $h \in [H]$, the corresponding forward and backward flight times of a drone to the $i$th customer, $i \in [N]$ are given by $u_i(r_h, \theta_h)$ and $v_i(r_h, \theta_h)$, respectively.

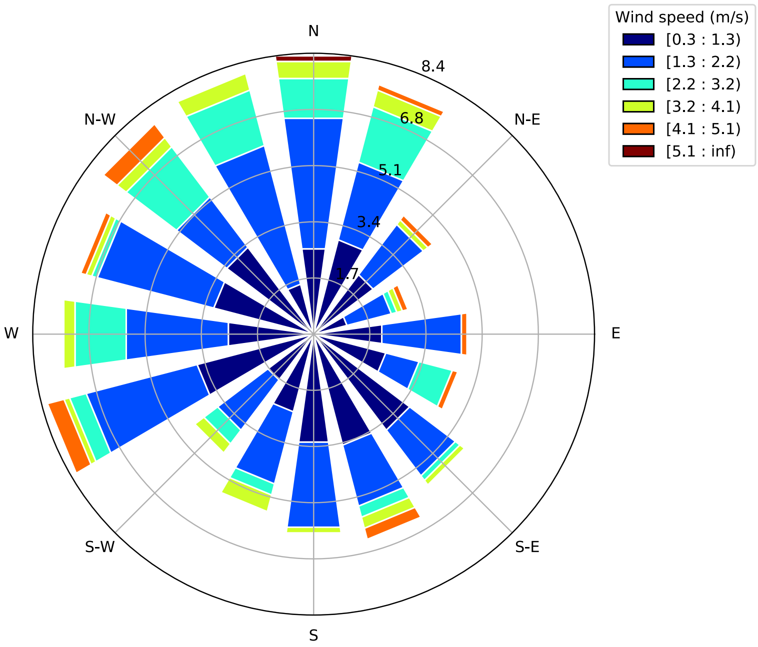

**Figure 2** Wind information collected from 145 subregions of Sichuan Province, China, ranging from time interval 00:00 to 04:00 on September 14th, 2019. Data are downloaded from the China National Meteorological Information Center (http://data.cma.cn/site/index.html). The color-coded bands represent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

## 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. The specific challenges and 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 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 to the empirical distribution. We incorporate

other weather information in the form of discrete scenarios to enhance the ambiguity set and improve the drone schedule adaptation.

3. Our approach minimizes a decision criterion that is inspired by the *essential riskiness index* (Zhang et al. 2019), which has performed well in vehicle routing problems in terms of meeting delivery deadlines by limiting the probability of tardy delivery and the magnitude of lateness. We exploit structures in our proposed decision criterion to reduce the complexity of our model.
4. For 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 in comparison with other classical models.

## Literature Review

This section reviews related works on drone delivery problems and data-driven robust optimization (RO). For more details about drones' civil applications, see the review paper by Otto et al. (2018). For RO methods, we refer interested readers to the review papers by Gorissen et al. (2015), Yanıkoğlu et al. (2019), and Rahimian and Mehrotra (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 come back. They denote the problem as the TSP with drone (TSP-D). Many papers focus on developing solution methods for the FSTSP or the TSP-D, using simulated annealing heuristic (Ponza 2016), variable neighborhood search (Freitas and Penna 2020), dynamic programming (Bouman et al. 2018), branch-and-bound (Poikonen et al. 2019), branch-and-price (Roberti and Ruthmair 2019), among others. Other variants are also studied. For example, Marinelli et al. (2017) extend the TSP-D by allowing the launching and rendezvous operations to be performed en route. Sacramento et al. (2019) address a vehicle routing problem (VRP) with drones, where multiple capacitated trucks, each equipped with one drone, are considered.

There are also other drone-related delivery problems, where trucks and drones do not necessarily work in tandem. Murray and Chu (2015) introduce a parallel drone scheduling TSP, where drones are launched and recovered at the depot, and the truck makes deliveries without carrying any drones. Dorling et al. (2016) study a multi-trip drone routing problem, where only drones are involved in the delivery system, and each drone is allowed to visit several customers per trip as long as the battery power capacity is respected. They use a linear approximation function to calculate the battery energy consumption. Cheng et al. (2020) address a similar problem in a B&C framework, where two types of cuts are proposed to calculate the nonlinear energy consumption, which is a convex function of payload.

One major aspect which makes drone delivery problems particularly challenging, as opposed to the common vehicle delivery problems, is the influence of weather conditions on drones' operational efficiency. Some recent studies attempted to tackle this important issue by incorporating weather conditions either directly or indirectly into the optimization models. Radzki et al. (2019) study a drone routing problem considering the impact of wind on energy consumption. They first assume that each drone travels at a fixed speed during a trip and the wind speed and direction remain unchanged (*i.e.*, given parameters) during the time horizon. They further assume that the total weight of a drone is constant during a trip. As all of the influencing factors are given parameters, energy consumption on each arc is a constant number. Thus, energy constraints are ultimately transformed into generalized resource constraints. Thibbotuwawa et al. (2019) design a decision support tool for the multi-trip drone fleet mission planning problem under different weather conditions by decomposing the problem into several subproblems. They first divide the time period into several flying time slots, and the weather condition in each slot is known and analogous. They then determine the clusters of customers for each slot and the routes for each cluster. Finally, they decide on the sequence of routes within each cluster. Kim et al. (2017) consider the influence of wind and obstacles on flight time by setting symmetric and asymmetric flight distances.

Vural et al. (2019) study a drone location routing problem under different weather conditions. They state that weather such as snow, fog, and heavy rain would limit drones' 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 assume that battery energy consumption can be negatively affected under certain weather conditions, which in turn affects drones' flight range. In order to incorporate the risk due to weather conditions, 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 specifically assume that 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 the impact of air temperature on battery duration. They use a RO method for the problem 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, then calculate the flight range deviation using a regression function. These deviations are fixed values and considered as different scenarios with weights. Thus, their uncertainty sets are constructed to represent the randomness of the scenario weights.

The aforementioned works 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, or follow a known distribution, or belong to some scenarios with known probabilities at the moment of making decisions. Only Kim et al. (2018) use a distribution-free method; however, they consider limited deviation scenarios and need to specify the budget or radius of the uncertainty sets. Unlike nearly all of the 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 robust optimization.** Our paper falls within the realm of data-driven RO, where the key step is to construct an uncertainty or ambiguity set from historical data. Details on the construction of these sets can be found in Delage and Ye (2010), Bertsimas et al. (2018), Ning and You (2018), and Chassein et al. (2019). 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. (2020) 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 not rainy). They use a multivariate regression tree to construct the ambiguity set. Perakis et al. (2020) study a two-period, multi-item joint pricing and production problem, where the $K$-means clustering algorithm is utilized to partition the demand residuals and then a cluster-wise ambiguity set is constructed. Our work is related to Zhang et al. (2019) and Zhang et al. (2020) where they study a vehicle routing problem with time window (VRPTW) under travel time uncertainty by minimizing the *essential riskiness index*, a convex decision criterion that takes into account of the probability of tardy delivery and the magnitude of lateness. While Zhang et al. (2019) explores a moment-based ambiguity set, Zhang et al. (2020) considers a Wasserstein distance-based ambiguity set. Quite distinct from their static decision models, we consider an adaptive decision model, which has the benefit of alleviating the conservatism of the robust solutions. We do so by considering a cluster-wise ambiguity set that is compatible with our two-period adaptive 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 distributionally robust model and presents the solution method. Section 5 reports numerical results. Section 6 concludes this paper.

**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 $\mathcal{W}$. We use $\xi^+$ to represent $\max\{\xi, 0\}$. 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 $i$th 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 sufficient supply of charged batteries, we do not need to incorporate charging decisions into our model. If a fully charged battery is not sufficient 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, $(r^1, \theta^1)$, 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 $(r_h^1, \theta_h^1)$ and $(r_h^2, \theta_h^2)$ 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 Fig. 3). There are various ways to partition the regions on the wind vector chart, and we will explore this further in our computational studies.

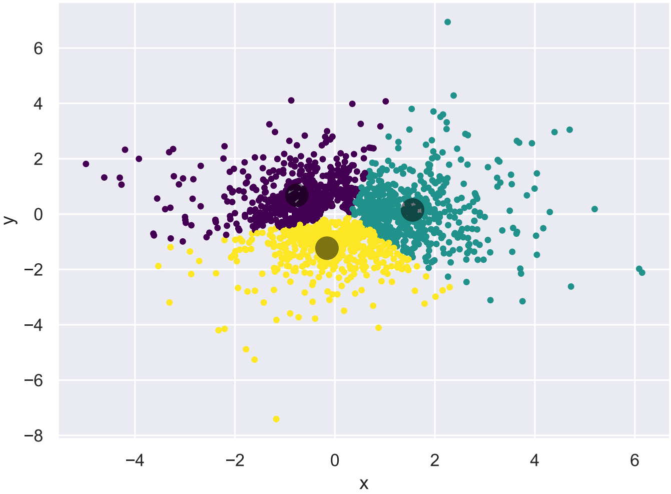

**Figure 3** Partitioning the wind vector chart into clusters using $K$-means clustering algorithm. Note that wind vectors are converted from polar coordinates to Cartesian coordinates when performing the clustering operation. The dark points in each cluster are the centroids.

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

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:

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

where the mean flight times associated with the cluster $k$, $k \in [K^1]$ are

$$\boldsymbol{\mu}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \boldsymbol{u}(r_h^1, \theta_h^1)\hat{\omega}, \quad \boldsymbol{\nu}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \boldsymbol{v}(r_h^1, \theta_h^1)\hat{\omega},$$

the mean absolute deviations are

$$\boldsymbol{\sigma}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \mathbb{E}_{\mathbb{P}_\omega}\left[|\boldsymbol{u}(r_h^1, \theta_h^1)\tilde{\omega} - \boldsymbol{\mu}_k^1|\right], \quad \boldsymbol{\varsigma}_k^1 = \frac{1}{|\mathcal{L}_k|} \sum_{h \in \mathcal{L}_k} \mathbb{E}_{\mathbb{P}_\omega}\left[|\boldsymbol{v}(r_h^1, \theta_h^1)\tilde{\omega} - \boldsymbol{\nu}_k^1|\right],$$

and the parameters of the supports are

$$\begin{array}{ll} [\underline{\boldsymbol{u}}_k^1]_i = \min\limits_{h \in [\mathcal{L}_k]} u_i(r_h^1, \theta_h^1)\underline{\omega}, \quad [\underline{\boldsymbol{v}}_k^1]_i = \min\limits_{h \in [\mathcal{L}_k]} v_i(r_h^1, \theta_h^1)\underline{\omega}, & \\ & \forall i \in [N]. \\ [\bar{\boldsymbol{u}}_k^1]_i = \max\limits_{h \in [\mathcal{L}_k]} u_i(r_h^1, \theta_h^1)\bar{\omega}, \quad [\bar{\boldsymbol{v}}_k^1]_i = \max\limits_{h \in [\mathcal{L}_k]} v_i(r_h^1, \theta_h^1)\bar{\omega}. & \end{array}$$

In the spirit of RO, the ambiguity set is often designed to encompass distributions that may deviate from the empirical distribution, which has the benefit of mitigating the risks of overfitting that may result in poor out-of-sample performance. Observe that when $K^1 = 1$, $\mathcal{G}^1$ is reduced to a marginal moment ambiguity set which, analogous to the *box uncertainty set*, can be quite conservative when applied to solving DRO problems. As we increase the number of clusters, the level of conservativeness reduces, though the computational complexity of the model would increase. In the extreme, each cluster can contain exactly one sample, in which case, due to the support and deviations constraint, the ambiguity set would only contain the empirical distribution if the factors $\tilde{\omega}_i^u, \tilde{\omega}_i^v$ are deterministic. In practice, the number of clusters is determined empirically in validation tests. If the number is small, this approach would benefit from greater scalability.

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 such as atmospheric pressures, weather forecasts, and other covariates, which could be used to predict the wind condition in the afternoon. To incorporate this information into our ambiguity set, we have to discretize the weather information into $S$ finite scenarios, which can be done using standard classification approaches used 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,

$$\mathcal{V}_s = \{h \in [H] \mid s_h = s\}. \tag{4}$$

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 condition 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 $(r_h^2, \theta_h^2)$, $h \in \mathcal{V}_s$ 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

$$\mathcal{L}_{sg}^2 = \left\{h \in \mathcal{V}_s \mid (r_h^2, \theta_h^2) \in \mathcal{U}_g^2\right\}.$$

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:

$$\mathcal{G}_s^2 = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{2N} \times [K^2]) \left| \begin{array}{ll} (\tilde{\boldsymbol{u}}^2, \tilde{\boldsymbol{v}}^2, \tilde{\kappa}^2) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{u}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\mu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^2 \mid \tilde{\kappa}^2 = g\right] = \boldsymbol{\nu}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2 - \boldsymbol{\mu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \le \boldsymbol{\sigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2 - \boldsymbol{\nu}_{sg}^2| \mid \tilde{\kappa}^2 = g\right] \le \boldsymbol{\varsigma}_{sg}^2 & \forall g \in [K^2] \\ \mathbb{P}\left[ \begin{array}{c} \underline{\boldsymbol{u}}_{sg}^2 \le \tilde{\boldsymbol{u}}^2 \le \bar{\boldsymbol{u}}_{sg}^2 \\ \underline{\boldsymbol{v}}_{sg}^2 \le \tilde{\boldsymbol{v}}^2 \le \bar{\boldsymbol{v}}_{sg}^2 \end{array} \middle| \tilde{\kappa}^2 = g \right] = 1 & \forall g \in [K^2] \\ \mathbb{P}\left[\tilde{\kappa}^2 = g\right] = |\mathcal{L}_{sg}^2| / |\mathcal{V}_s| & \forall g \in [K^2] \end{array} \right. \right\}, \tag{5}$$

where the mean flight times associated with cluster $g$, $g \in [K^2]$ under scenario $s$, $s \in \mathcal{S}$ are

$$\boldsymbol{\mu}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \boldsymbol{u}(r_h^2, \theta_h^2)\hat{\omega}, \quad \boldsymbol{\nu}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \boldsymbol{v}(r_h^2, \theta_h^2)\hat{\omega},$$

the mean absolute deviations are

$$\boldsymbol{\sigma}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \mathbb{E}_{\mathbb{P}_\omega}\left[|\boldsymbol{u}(r_h^2, \theta_h^2)\tilde{\omega} - \boldsymbol{\mu}_{sg}^2|\right], \quad \boldsymbol{\varsigma}_{sg}^2 = \frac{1}{|\mathcal{L}_{sg}^2|} \sum_{h \in \mathcal{L}_{sg}^2} \mathbb{E}_{\mathbb{P}_\omega}\left[|\boldsymbol{v}(r_h^2, \theta_h^2)\tilde{\omega} - \boldsymbol{\nu}_{sg}^2|\right],$$

and the parameters of the supports are

$$\begin{aligned}
\left[\underline{\boldsymbol{u}}^2_{sg}\right]_i &= \min_{h\in[\mathcal{L}^2_{sg}]} u_i(r^2_h,\theta^2_h)\underline{\omega}, & \left[\underline{\boldsymbol{v}}^2_{sg}\right]_i &= \min_{h\in[\mathcal{L}^2_{sg}]} v_i(r^2_h,\theta^2_h)\underline{\omega}, \\
\left[\bar{\boldsymbol{u}}^2_{sg}\right]_i &= \max_{h\in[\mathcal{L}^2_{sg}]} u_i(r^2_h,\theta^t_h)\bar{\omega}, & \left[\bar{\boldsymbol{v}}^2_{sg}\right]_i &= \max_{h\in[\mathcal{L}^2_{sg}]} v_i(r^2_h,\theta^2_h)\bar{\omega}.
\end{aligned}\qquad \forall i\in[N].$$

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:

$$\mathcal{F}=\left\{\mathbb{Q}\in\mathcal{P}(\mathbb{R}^{4N}\times[S])\;\middle|\;\begin{array}{ll}
(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{s})\sim\mathbb{Q} & \\
\exists\mathbb{P}\in\mathcal{P}(\mathbb{R}^{4N}\times\mathcal{W}): & \\
\qquad(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,(\tilde{\kappa}^1,\tilde{\kappa}^2,\tilde{s}))\sim\mathbb{P} & \\
\exists\mathbb{Q}^1\in\mathcal{G}^1:(\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\kappa}^1)\sim\mathbb{Q}^1 & \\
\exists\mathbb{Q}^2_s\in\mathcal{G}^2_s:(\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{\kappa}^2)|_{\tilde{s}=s}\sim\mathbb{Q}^2_s & \forall s\in[S] \\
\mathbb{P}\left[(\tilde{\kappa}^1,\tilde{\kappa}^2,\tilde{s})=(k,g,s)\right]=q_{kgs} & \forall(k,g,s)\in\mathcal{W}
\end{array}\right\}, \tag{6}$$

where

$$q_{kgs}=\frac{|\mathcal{V}_s\cap\mathcal{L}_k\cap\mathcal{L}^2_{sg}|}{H},$$

or explicitly as

$$\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}): & \\
\qquad(\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}^1_k & \forall k\in[K^1] \\
\mathbb{E}_{\mathbb{P}}\left[\tilde{\boldsymbol{v}}^1\mid\tilde{\kappa}^1=k\right]=\boldsymbol{\nu}^1_k & \forall k\in[K^1] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^1-\boldsymbol{\mu}^1_k|\mid\tilde{\kappa}^1=k\right]\le\boldsymbol{\sigma}^1_k & \forall k\in[K^1] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^1-\boldsymbol{\nu}^1_k|\mid\tilde{\kappa}^1=k\right]\le\boldsymbol{\varsigma}^1_k & \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}^2_{sg} & \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}^2_{sg} & \forall s\in[S],g\in[K^2] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{u}}^2-\boldsymbol{\mu}^2_{sg}|\mid\tilde{s}=s,\tilde{\kappa}^2=g\right]\le\boldsymbol{\sigma}^2_{sg} & \forall s\in[S],g\in[K^2] \\
\mathbb{E}_{\mathbb{P}}\left[|\tilde{\boldsymbol{v}}^2-\boldsymbol{\nu}^2_{sg}|\mid\tilde{s}=s,\tilde{\kappa}^2=g\right]\le\boldsymbol{\varsigma}^2_{sg} & \forall s\in[S],g\in[K^2] \\
\mathbb{P}\left[\begin{array}{c|c}
\underline{\boldsymbol{u}}^1_k\le\tilde{\boldsymbol{u}}^1\le\bar{\boldsymbol{u}}^1_k & \tilde{\kappa}^1=k, \\
\underline{\boldsymbol{v}}^1_k\le\tilde{\boldsymbol{v}}^1\le\bar{\boldsymbol{v}}^1_k & \tilde{\kappa}^2=g, \\
\underline{\boldsymbol{u}}^2_{sg}\le\tilde{\boldsymbol{u}}^2\le\bar{\boldsymbol{u}}^2_{sg} & \tilde{s}=s \\
\underline{\boldsymbol{v}}^2_{sg}\le\tilde{\boldsymbol{v}}^2\le\bar{\boldsymbol{v}}^2_{sg} &
\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\}. \tag{7}$$

## 3. The Drone Delivery Model

Our goal is to robustly optimize the schedules of the drone delivery system to mitigate the risks of delivery delays caused by wind uncertainty. Customers specify their preference for delivery in either the morning or in the 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 could not 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 *essential riskiness index* (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, to help meet delivery deadlines by limiting the probability of tardy delivery and the magnitude of lateness (Zhang et al. 2020).

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

$$\rho_\tau(\tilde{\xi}) = \min \left\{ \gamma \geq 0 \;\middle|\; \sup_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} \left[ \max \left\{ \tilde{\xi} - \tau, -\gamma \right\} \right] \leq 0 \right\}, \tag{8}$$

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 $\sup\limits_{\mathbb{P} \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\tilde{\xi}\right] > \tau$, 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 arriving 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. Finally, as shown in the property of delay bounds, the ERI ensures that the probability of lateness reduces reciprocally as the magnitude of lateness increases in multiples of the index value. Thus, unlike the probability of service delay, the ERI accounts for both the probability of delay and its magnitude. ERI has other useful and interesting properties and we refer interested readers to Zhang et al. (2019) and Zhang et al. (2020).

The drone delivery problem can be modeled as a multi-trip VRPTW proposed in Zhang et al. (2020), where the objective of the model 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

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

where $\tilde{\xi}_i$ and $\tau_i$ are, respectively, the uncertain arriving time and expected delivery time for the $i$th customer. The benefit of this criterion in achieving high-quality solutions with reasonable computational effort have been demonstrated in Zhang et al. (2020). 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 the 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 arriving 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, which has an impact on the quality of the solutions as well as the computational time to solve the model. We will illustrate this in our computational studies.

While it may be possible to cast our problem as a static robust VPRTW proposed in Zhang et al. (2020), our goal is to extend to an adaptive model, where it is possible to improve the drone delivery schedule in response to the new information available by midday. The static robust VPRTW of Zhang et al. (2020) would be far too complex for our purpose. One simplifying assumption that enables us to take this approach is that customers can only choose between two delivery deadlines, $\{\bar{\tau}^1, \bar{\tau}^2\}$. Alongside this, we propose the following joint decision criterion:

$$\varrho_{\boldsymbol{\tau}}(\tilde{\boldsymbol{\xi}}) \triangleq \sum_{d\in[D]:\mathcal{K}_d^1\neq\emptyset} \left( \max_{i\in\mathcal{K}_d^1} \left\{ \rho_{\bar{\tau}^1}\left(\tilde{\xi}_i\right) \right\} \right) + \sum_{d\in[D]:\mathcal{K}_d^2\neq\emptyset} \left( \max_{i\in\mathcal{K}_d^2} \left\{ \rho_{\bar{\tau}^2}\left(\tilde{\xi}_i\right) \right\} \right), \tag{9}$$

where $\mathcal{K}_d^t \subseteq [N]$ denotes the set of customers assigned to be served by drone $d$, $d\in[D]$ in period $t$, $t\in[2]$. Observe that the joint decision criterion also satisfies the properties of monotonicity, satisficing, non-abandonment, and convexity.

The main reason for our proposed joint decision criterion is the potential to have a simpler formulation that is more scalable computationally, as we will demonstrate in the following proposition.

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

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

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

*Proof.* 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

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

□

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

$$\tilde{\xi}_{\ell_d^1} = \sum_{i\in\mathcal{K}_d^1\setminus\{\ell_d^1\}} (\tilde{u}_i^1 + \tilde{v}_i^1) + \tilde{u}_{\ell_d^1}^1. \tag{10}$$

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 is able to reduce its speed or 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 $\underline{\ell}^2_d$. Hence, the random arrival time at the last customer by drone $d$, $d \in [D]$ in Period 2 is

$$\tilde{\xi}_{\ell^2_d} = \max\left\{ \sum_{i\in\mathcal{K}^1_d} (\tilde{u}^1_i + \tilde{v}^1_i) + \tilde{u}^2_{\underline{\ell}^2_d}, \bar{\tau}^1 \right\} + \tilde{v}^2_{\underline{\ell}^2_d} + \sum_{i\in\mathcal{K}^2_d\setminus\{\underline{\ell}^2_d, \ell^2_d\}} \left(\tilde{u}^2_i + \tilde{v}^2_i\right) + \tilde{u}^2_{\ell^2_d} \qquad \forall d \in [D]. \tag{11}$$

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 customer in Period 2, would not affect the joint decision criterion. This greatly reduces the complexity of the model compared to a VRPTW formulation, in terms of the number of binary variables needed to model the problem.

**Scheduling Decisions and Event-wise Adaptations**

To obtain an explicit mathematical optimization model, we first 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 $i \in \mathcal{K}^1_d \setminus \{\ell^1_d\}$.

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 $y^2_{id}(s), x^2_{id}(s), z^2_{id}(s)$, $i \in \bar{\mathcal{C}}^2$, $d \in [D]$. For a given scenario $s \in [S]$, the binary variable $y^2_{id}(s) = 1$ if and only if $i \in \mathcal{K}^2_d \cap \{\ell^2_d\}$, the binary variable $x^2_{id}(s) = 1$ if and only if $i \in \mathcal{K}^2_d \setminus \{\underline{\ell}^2_d, \ell^2_d\}$ and the binary variable $z^2_{id}(s) = 1$ if and only if $i \in \mathcal{K}^2_d \cap \{\underline{\ell}^2_d\}$.

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 the 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:

$$\mathcal{X} = \left\{ \begin{array}{c} (\boldsymbol{x}^1, \boldsymbol{y}^1, \\ \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \end{array} \;\middle|\; \begin{array}{lr} \sum_{d \in [D]} (x^1_{id} + y^1_{id}) = 1 & \forall i \in \mathcal{C}^1, \\ \sum_{d \in [D]} \left( x^2_{id}(s) + y^2_{id}(s) + z^2_{id}(s) \right) = 1 & \forall i \in \mathcal{C}^2, s \in [S], \\ \sum_{d \in [D]} \left( x^1_{id} + y^1_{id} + x^2_{id}(s) + y^2_{id}(s) + z^2_{id}(s) \right) = 1 & \forall i \in \mathcal{C}^3, s \in [S], \\ \sum_{i \in \bar{\mathcal{C}}^1} y^1_{id} = 1 & \forall d \in [D], \\ \sum_{i \in \bar{\mathcal{C}}^2} y^2_{id}(s) = 1 & \forall d \in [D], s \in [S], \\ \sum_{i \in \bar{\mathcal{C}}^2} z^2_{id}(s) = 1 & \forall d \in [D], s \in [S], \\ \boldsymbol{x}^1, \boldsymbol{y}^1 \in \{0,1\}^{\bar{\mathcal{C}}^1 \times D} & \\ \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2 \in \mathcal{A}(\mathcal{S}). & \end{array} \right\} \quad \begin{array}{r} (12a) \\ (12b) \\ (12c) \\ (12d) \\ (12e) \\ (12f) \\ \\ \\ \end{array}$$

Constraints (12a)–(12c) mean that customers in each set are respectively scheduled to be visited exactly once in their allowable delivery time slots. Constraints (12d)–(12e) impose that each drone at each period can only have one customer as the last served one. Constraints (12f) mean that for each drone in Period 2, it can only have one customer as the first visited customer. The index $s$ in the constraints suggests that the afternoon schedule should be feasible under any scenario $s$, $s \in [S]$.

The schedule decision can be enhanced by the following lexicographic ordering constraints, which break the symmetry between drones (Adulyasak et al. 2014)

$$\sum_{i=1, i \in \mathcal{C}^1}^{j} 2^{j-i} x^1_{id} \geq \sum_{i=1, i \in \mathcal{C}^1}^{j} 2^{j-i} x^1_{id+1} \qquad \forall j \in \mathcal{C}^1, d \in [D-1]. \tag{13}$$

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

$$\zeta^1_d(\boldsymbol{x}^1, \boldsymbol{y}^1, \tilde{\boldsymbol{u}}^1, \tilde{\boldsymbol{v}}^1) \triangleq \sum_{i \in \bar{\mathcal{C}}^1} \left( x^1_{id}(\tilde{u}^1_i + \tilde{v}^1_i) + y^1_{id} \tilde{u}^1_i \right), \tag{14}$$

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

$$\begin{aligned}&\zeta_d^2(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{s})\\ \triangleq\ &\max\left\{\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(\tilde{u}^1_i+\tilde{v}^1_i)+\sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(\tilde{s})\tilde{u}^2_i,\bar{\tau}^1\right\}\\ &\quad+\sum_{i\in\bar{\mathcal{C}}^2}\left(z^2_{id}(\tilde{s})\tilde{v}^2_i+x^2_{id}(\tilde{s})(\tilde{u}^2_i+\tilde{v}^2_i)+y^2_{id}(\tilde{s})\tilde{u}^2_i\right).\end{aligned} \tag{15}$$

Thus, under our proposed decision criterion, we solve the following DRO problem:

$$\begin{aligned}&\inf\sum_{d\in[D]}(\gamma^1_d+\gamma^2_d)\\ &\text{s.t.}\sup_{\mathbb{P}\in\mathcal{F}}\mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta^1_d(\boldsymbol{x}^1,\boldsymbol{y}^1,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1)-\bar{\tau}^1,-\gamma^1_d\right\}\right]\le 0 && \forall d\in[D] && \text{(16a)}\\ &\quad\ \sup_{\mathbb{P}\in\mathcal{F}}\mathbb{E}_{\mathbb{P}}\left[\max\left\{\zeta^2_d(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1,\tilde{\boldsymbol{u}}^2,\tilde{\boldsymbol{v}}^2,\tilde{s})-\bar{\tau}^2,-\gamma^2_d\right\}\right]\le 0 && \forall d\in[D] && \text{(16b)}\\ &\quad\ \gamma^1_d,\gamma^2_d\ge 0 && \forall d\in[D],\\ &\quad\ (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X}.\end{aligned}$$

The DRO model (16) with cluster-wise moment-based ambiguity set and event-wise adaption fits into the *robust stochastic optimization* framework recently 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 MATLAB-based toolbox generates numerous auxiliary variables and results in slow computational performance. Our goal is to eradicate all of the 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, we present a classical RO reformulation, which would enable us to develop a B&C algorithm to solve the model. We first address the supremum over the ambiguity set in the constraints (16a).

Note that $\sup_{\mathbb{P}\in\mathcal{F}}\mathbb{E}_{\mathbb{P}}[\max\{\zeta^1_d(\boldsymbol{x}^1,\boldsymbol{y}^1,\tilde{\boldsymbol{u}}^1,\tilde{\boldsymbol{v}}^1)-\bar{\tau}^1,-\gamma^1_d\}]$ is equivalent to

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

Under the law of total probability, constraints (16a) can be reformulated as

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

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

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

and

$$\Xi_k^1 \triangleq \left\{ (\boldsymbol{u}^1, \boldsymbol{a}^1, \boldsymbol{v}^1, \boldsymbol{b}^1) \in \mathbb{R}^{4N} \;\middle|\; \begin{array}{l} \underline{\boldsymbol{u}}_k^1 \le \boldsymbol{u}^1 \le \bar{\boldsymbol{u}}_k^1 \\ \boldsymbol{a}^1 \ge |\boldsymbol{u}^1 - \boldsymbol{\mu}_k^1| \\ \underline{\boldsymbol{v}}_k^1 \le \boldsymbol{v}^1 \le \bar{\boldsymbol{v}}_k^1 \\ \boldsymbol{b}^1 \ge |\boldsymbol{v}^1 - \boldsymbol{\nu}_k^1| \end{array} \right\}.$$

Note here that we adopt the "lifted ambiguity set" first proposed by Wiesemann et al. (2014) to keep the expectation constraints within the ambiguity set linear.

Theorem 1. *The optimization problem*

$$\sup_{\mathbb{P} \in \mathcal{G}_k^1} \mathbb{E}_{\mathbb{P}} \left[ \max \left\{ \sum_{i \in \bar{\mathcal{C}}^1} \left( x_{id}^1 (\tilde{u}_i^1 + \tilde{v}_i^1) + y_{id}^1 \tilde{u}_i^1 \right) - \bar{\tau}^1, -\gamma_d^1 \right\} \right] \tag{17}$$

*is equivalent to*

$$\max_{(\boldsymbol{u}_1^1, \boldsymbol{v}_1^1, p_1, p_2) \in \mathcal{Y}_{kd}^1} \left\{ \left( \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1) \boldsymbol{e}_i \right)' \boldsymbol{u}_1^1 + \left( \sum_{i \in \bar{\mathcal{C}}^1} x_{id}^1 \boldsymbol{e}_i \right)' \boldsymbol{v}_1^1 - \bar{\tau}^1 p_1 - \gamma_d^1 p_2 \right\}, \tag{18}$$

*where*

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

*Proof.* See Appendix A.1. □

To address the supremum over the ambiguity set in the constraints (16b), we also note that under the law of total probability, constraints (16b) is equivalent to

$$\begin{aligned}\sum_{(k,g,s)\in\mathcal{W}} q_{kgs} \sup_{\mathbb{P}\in\mathcal{F}_{kgs}} \mathbb{E}_{\mathbb{P}}\Bigg[\max\Bigg\{ \max\Bigg\{\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(\tilde{u}^1_i+\tilde{v}^1_i)+\sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(s)\tilde{u}^2_i,\bar{\tau}^1\Bigg\}& \\ +\sum_{i\in\bar{\mathcal{C}}^2}\left(z^2_{id}(s)\tilde{v}^2_i+x^2_{id}(s)(\tilde{u}^2_i+\tilde{v}^2_i)+y^2_{id}(s)\tilde{u}^2_i\right)-\bar{\tau}^2,-\gamma^2_d\Bigg\}\Bigg]\le 0 & \qquad \forall d\in[D] \qquad (19)\end{aligned}$$

where

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

and

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

THEOREM 2. *The optimization problem,*

$$\begin{aligned}\sup_{\mathbb{P}\in\mathcal{F}_{kgs}}\mathbb{E}_{\mathbb{P}}\Bigg[\max\Bigg\{\ \max\Bigg\{\sum_{i\in\bar{\mathcal{C}}^1}(x^1_{id}+y^1_{id})(\tilde{u}^1_i+\tilde{v}^1_i)+\sum_{i\in\bar{\mathcal{C}}^2}z^2_{id}(s)\tilde{u}^2_i,\bar{\tau}^1\Bigg\}& \\ +\sum_{i\in\bar{\mathcal{C}}^2}(z^2_{id}(s)\tilde{v}^2_i+x^2_{id}(s)(\tilde{u}^2_i+\tilde{v}^2_i)+y^2_{id}(s)\tilde{u}^2_i)-\bar{\tau}^2,-\gamma^2_d\Bigg\}\Bigg],&\end{aligned}\qquad (20)$$

*is equivalent to*

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

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

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

*Proof.* See Appendix A.2. □

With Theorems 1 and 2, we can now transform the DRO problem to a classical linear RO problem (see, for instance, Ben-Tal and Nemirovski 1998) as follows:

$$
\begin{aligned}
&\inf\sum_{d\in[D]}(\gamma_d^1+\gamma_d^2)\\
&\text{s.t.}\sum_{k\in[K^1]}q_k\left(\left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^{1k}+\left(\sum_{i\in\bar{\mathcal{C}}^1}x_{id}^1\boldsymbol{e}_i\right)'\boldsymbol{v}_1^{1k}-\bar{\tau}^1p_1^k-\gamma_d^1p_2^k\right)\\
&\qquad\le 0\qquad\forall(\boldsymbol{u}_1^{1k},\boldsymbol{v}_1^{1k},p_1^k,p_2^k)\in\mathcal{Y}_{kd}^1,k\in[K^1],\qquad\forall d\in[D]
\end{aligned}\tag{21a}
$$

$$
\begin{aligned}
&\sum_{(k,g,s)\in\mathcal{W}} q_{kgs}\Bigg( \left(\sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^{1kgs}+\boldsymbol{v}_1^{1kgs}) + \left(\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2(s)+x_{id}^2(s))\boldsymbol{e}_i\right)'(\boldsymbol{u}_1^{2kgs}+\boldsymbol{v}_1^{2kgs}) \\
&+\left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_1^{2kgs} - \bar{\tau}^2 p_1^{kgs} + \bar{\tau}^1 p_2^{kgs} + \left(\sum_{i\in\bar{\mathcal{C}}^2} z_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{v}_2^{2kgs} + \left(\sum_{i\in\bar{\mathcal{C}}^2} x_{id}^2(s)\boldsymbol{e}_i\right)'(\boldsymbol{u}_2^{2kgs}+\boldsymbol{v}_2^{2kgs}) \\
&+\left(\sum_{i\in\bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)'\boldsymbol{u}_2^{2kgs} - \bar{\tau}^2 p_2^{kgs} - \gamma_d^2 p_3^{kgs}\Bigg) \le 0 \\
&\qquad \forall(\boldsymbol{u}_1^1,\boldsymbol{v}_1^1,\boldsymbol{u}_1^2,\boldsymbol{v}_1^2,\boldsymbol{u}_2^2,\boldsymbol{v}_2^2,p_1,p_2,p_3)\in\mathcal{Y}_{kgsd}^2,(k,g,s)\in\mathcal{W} \qquad \forall d\in[D] \qquad \text{(21b)} \\
&\gamma_d^1,\gamma_d^2 \ge 0 \qquad \forall d\in[D] \\
&(\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X},
\end{aligned}
$$

where constraints (21a) and (21b) 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 (21) by ignoring these two groups of constraints. Whenever an integer solution is generated, we solve the equivalent maximum models presented in Theorem 1 and Theorem 2. If the resulting solutions violate the constraints, cuts are generated and added to the model, and the model is solved again. 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 set drones' launch speed in both directions to 20 $m/s$, *i.e.*, $\bar{r}_0 = \bar{r}_i = 20$ $m/s$, $i\in[N]$, based on Amazon Prime Air and Workhorse HorseFly drones (used by UPS), which can fly at speeds up to 50 *mph* (*i.e.*, 22.35 $m/s$) (Lavars 2015, Burns 2017). 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 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 speed of 20 $m/s$) to deliver packages (Lee 2019) and the HorseFly drone has a 30-minute flight time (Burns 2017). 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 $\bar{\tau}^2=8000$. We randomly generate instances of 15 and 20 customers, where 2 and 3 drones are used in the delivery system, respectively. 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 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 14th to 19th, 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 14th as the (in-sample) training data, *i.e.*, $H = 1740$, and the data from September 15th to 19th 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 note that 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. All of the algorithms and models are implemented in Python programming language using Gurobi 7.5.1 as the solver. The computations are executed on an Intel Core i5 2.3 GHz processor with 8GB 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 that maximizes the slack of 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 drone delivery system is service-oriented, a natural objective would be to construct a drone schedule that maximizes the slack of time in a deterministic model.

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

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 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 to guarantee 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:

$$\begin{aligned} &\max \ \mathbb{P}\left[\boldsymbol{\zeta} - \bar{\boldsymbol{\tau}} \leq 0\right] \\ &\text{s.t.} \ \ (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X}, \end{aligned} \tag{22}$$

where $\boldsymbol{\zeta} = \begin{bmatrix} \zeta_d^1, \dots, \zeta_D^1 \\ \zeta_d^2, \dots, \zeta_D^2 \end{bmatrix}$ 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 (22) 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 (22) as follows:

$$\begin{aligned} \max \ & \frac{1}{H} \sum_{h \in [H]} I_h \\ \text{s.t.} \ & \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1(u_{ih}^1 + v_{ih}^1) + y_{id}^1 u_{ih}^1) - \bar{\tau}^1 \leq (1 - I_h) M_h^1 && \forall d \in [D], h \in [H], \\ & \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)(u_{ih}^1 + v_{ih}^1) + \sum_{i \in \bar{\mathcal{C}}^2} ((z_{id}^2 + x_{id}^2)(u_{ih}^2 + v_{ih}^2) + y_{id}^2 u_{ih}^2) - \bar{\tau}^2 \leq (1 - I_h) M_h^2 && \forall d \in [D], h \in [H], \\ & \bar{\tau}^1 + \sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2 v_{ih}^2 + x_{id}^2(u_{ih}^2 + v_{ih}^2) + y_{id}^2 u_{ih}^2) - \bar{\tau}^2 \leq (1 - I_h) M_h^2 && \forall d \in [D], h \in [H], \\ & I_h \in \{0, 1\} && \forall h \in [H], \\ & (\boldsymbol{x}^1, \boldsymbol{y}^1, \boldsymbol{x}^2, \boldsymbol{y}^2, \boldsymbol{z}^2) \in \mathcal{X}, \end{aligned}$$

where $u_{ih}^t \in \mathbb{R}$ and $v_{ih}^t \in \mathbb{R}$ are the travel times in Period $t$, $t \in [2]$ with respect to sample $h$, $h \in [H]$. $I_h = 1$ if and only if all drones at all periods serve customers on time under sample $h$, $h \in [H]$. $M_h^t$, $h \in [H]$, $t \in [2]$ is a sufficiently large number. We choose $M_h^1 = \sum_{i \in \bar{\mathcal{C}}^1} (u_{ih}^1 + v_{ih}^1)$ and $M_h^2 = \max\left\{ \sum_{i \in \bar{\mathcal{C}}^1} (u_{ih}^1 + v_{ih}^1), \bar{\tau}^1 \right\} + \sum_{i \in \bar{\mathcal{C}}^2} (u_{ih}^2 + v_{ih}^2)$.

**Minimizing ERI Using Empirical Distribution (ERI-E).** For Model (16), we assume the travel times follow an empirical distribution, thus the expectations in constraints (16a) and (16b) are now evaluated over a known distribution. The resulting model is

$$\begin{aligned}
\inf\ & \sum_{d\in[D]}(\gamma_d^1+\gamma_d^2) & \\
\text{s.t. } & \tfrac{1}{H}\sum_{h\in[H]} w_{dh}^t \le 0 & \forall d\in[D], t\in[2],\\
& w_{dh}^1 \ge \sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1(u_{ih}^1+v_{ih}^1)+y_{id}^1u_{ih}^1)-\bar{\tau}^1 & \forall d\in[D], h\in[H],\\
& w_{dh}^2 \ge \sum_{i\in\bar{\mathcal{C}}^1}(x_{id}^1+y_{id}^1)(u_{ih}^1+v_{ih}^1)+\sum_{i\in\bar{\mathcal{C}}^2}((z_{id}^2+x_{id}^2)(u_{ih}^2+v_{ih}^2)+y_{id}^2u_{ih}^2)-\bar{\tau}^2 & \forall d\in[D], h\in[H],\\
& w_{dh}^2 \ge \bar{\tau}^1+\sum_{i\in\bar{\mathcal{C}}^2}(z_{id}^2v_{ih}^2+x_{id}^2(u_{ih}^2+v_{ih}^2)+y_{id}^2u_{ih}^2)-\bar{\tau}^2 & \forall d\in[D], h\in[H],\\
& w_{dh}^t \ge -\gamma_d^t & \forall d\in[D], h\in[H], t\in[2],\\
& \gamma_d^t \ge 0 & \forall d\in[D], t\in[2],\\
& (\boldsymbol{x}^1,\boldsymbol{y}^1,\boldsymbol{x}^2,\boldsymbol{y}^2,\boldsymbol{z}^2)\in\mathcal{X}. &
\end{aligned}$$

**Minimizing the Maximum ERI (M-ERI).** We can also minimize the maximum ERI with empirical distribution. To do so, we replace the objective of the ERI-E model with

$$\min\gamma$$

and add the additional constraints

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

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. In addition, 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.

### Algorithm Performance

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 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 Performance comparison (CPU time in seconds) between B&C and RSOME**

| $N$ | $K^1$ | $K^2$ | B&C | RSOME | Time ratio |
|---|---|---|---|---|---|
| 15 | 1 | 1 | 4.98 | 3.36 | 1.48 |
| | | 2 | 10.09 | 19.38 | 0.52 |
| | | 3 | 18.57 | 37.42 | 0.50 |
| | 2 | 1 | 19.93 | 29.15 | 0.68 |
| | | 2 | 50.53 | 120.05 | 0.42 |
| | | 3 | 102.70 | 344.24 | 0.30 |
| | 3 | 1 | 88.89 | 201.41 | 0.44 |
| | | 2 | 243.65 | 294.56 | 0.83 |
| | | 3 | 471.87 | 1868.23 | 0.25 |
| | 4 | 1 | 880.01 | 466.13 | 1.89 |
| | | 2 | 2353.60 | 5485.28 | 0.43 |
| | 5 | 1 | 3478.40 | 6615.62 | 0.53 |

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 particular, when $K^2 = 3$, the average CPU time of the B&C is equal to or less than half of the computing time of the RSOME. Therefore, we can conclude that the B&C algorithm can generally achieve a greater scalability for the adaptive DRO model compared to RSOME.

### Comparison of Decision Criteria

We first 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 very 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, the value of *MaxPro* is around 3%.

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

| $N$ | Model | MaxPro | SumPro | MaxExp | SumExp | CPU time |
|---|---|---|---|---|---|---|
| 15 | DM | 0.1835 | 0.2410 | 17.17 | 25.43 | 0.06 |
| | MOP | 0.0362 | 0.0743 | 7.93 | 16.11 | 2563.31 |
| | M-ERI | 0.0360 | 0.0732 | 7.56 | 15.65 | 530.04 |
| | ERI-E | 0.0339 | 0.0668 | 7.57 | 14.81 | 90.22 |
| 20 | DM | 0.1303 | 0.1809 | 17.38 | 29.01 | 0.06 |
| | MOP | 0.0309 | 0.0892 | 7.11 | 20.36 | 3580.08 |
| | M-ERI | 0.0319 | 0.1053 | 6.90 | 22.78 | 3600.00 |
| | ERI-E | 0.0298 | 0.0850 | 7.09 | 19.86 | 3599.33 |

In general, Method ERI-E provides a better out-of-sample performance than the three other methods. Specifically, it shows a superiority in indicators *MaxPro*, *SumPro*, and *SumExp*. It also consumes much less CPU time than Methods M-ERI and MOP when $N = 15$. Method M-ERI performs better with regard to *MaxExp*, which is consistent with its objective function; however, it performs worse in other indicators and requires more CPU time in comparison with ERI-E. This is because there exist multiple optimal solutions under this criterion. Based on the results here, we use Method ERI-E to evaluate our robust framework in the next section.

### Comparison Between Robust Method and Empirical Distribution

In this section, we compare the out-of-sample performance of the static DRO model and Method ERI-E. The DRO model is solved by the B&C algorithm with the same optimality gap and time limit as those of Method ERI-E. The average results are reported in Table 3, where the first row under each $N$ presents the results of Method ERI-E.

Table 3 shows that the static DRO model can find solutions with better out-of-sample performance in all indicators for $N = 15$ when $(K^1, K^2) = (5, 1)$ and for $N = 20$ when $(K^1, K^2) = (4, 2)$. Moreover, the static DRO model consumes less CPU time in these two cases compared to its empirical counterpart, ERI-E. When $N = 20$ and $(K^1, K^2) = (5, 1)$, the DRO model produces solutions with better performance in indicators *MaxPro*, *SumPro*, and *MaxExp*, and the value of *SumExp* is larger than that of Method ERI-E by only 0.01. However, the DRO model requires much less computing time (1428.63 seconds versus 3599.33 seconds) on average. Thus, we can conclude that the robust method is more effective in mitigating the service lateness than the empirical distribution for our drone delivery problem.

### Comparison Between Adaptive and Static Robust Models

This section compares the performance of the adaptive and static DRO models. The average results are reported in Table 4, with results of the static model shown in parentheses. Under the same cluster numbers, if the static model performs better in an indicator, we mark the results in bold.

**Table 3 Average results of out-of-sample tests generated by the static DRO model**

| $N$ | $K^1$ | $K^2$ | MaxPro | SumPro | MaxExp | SumExp | CPU time |
|---|---|---|---|---|---|---|---|
| 15 | ERI-E | | 0.0339 | 0.0668 | 7.57 | 14.81 | 90.22 |
| | 1 | 1 | 0.0391 | 0.0677 | 8.61 | 15.23 | 5.15 |
| | | 2 | 0.0344 | 0.0655 | 8.01 | 15.06 | 11.37 |
| | | 3 | 0.0332 | 0.0647 | 7.70 | 14.71 | 22.49 |
| | 2 | 1 | 0.0349 | 0.0647 | 8.05 | 14.86 | 9.90 |
| | | 2 | 0.0337 | 0.0652 | 7.77 | 14.65 | 25.68 |
| | | 3 | 0.0364 | 0.0657 | 7.83 | 14.90 | 56.56 |
| | 3 | 1 | 0.0340 | 0.0650 | 7.93 | 14.83 | 26.41 |
| | | 2 | 0.0330 | 0.0646 | 7.81 | 14.77 | 72.18 |
| | | 3 | 0.0325 | 0.0644 | 7.83 | 14.79 | 93.55 |
| | 4 | 1 | 0.0334 | 0.0650 | 7.70 | 14.88 | 35.93 |
| | | 2 | 0.0340 | 0.0650 | 7.71 | 14.82 | 89.10 |
| | 5 | 1 | 0.0329 | 0.0644 | 7.53 | 14.63 | 65.45 |
| 20 | ERI-E | | 0.0298 | 0.0850 | 7.09 | 19.86 | 3599.33 |
| | 1 | 1 | 0.0406 | 0.0931 | 8.74 | 21.15 | 18.58 |
| | | 2 | 0.0321 | 0.0865 | 7.34 | 20.34 | 49.37 |
| | | 3 | 0.0323 | 0.0838 | 7.59 | 20.06 | 96.90 |
| | 2 | 1 | 0.0364 | 0.0889 | 8.14 | 20.66 | 68.16 |
| | | 2 | 0.0300 | 0.0838 | 7.33 | 20.04 | 475.42 |
| | | 3 | 0.0343 | 0.0849 | 7.69 | 20.00 | 595.31 |
| | 3 | 1 | 0.0350 | 0.0855 | 8.26 | 20.12 | 265.55 |
| | | 2 | 0.0304 | 0.0830 | 7.23 | 19.82 | 891.86 |
| | | 3 | 0.0303 | 0.0836 | 7.43 | 19.85 | 1734.09 |
| | 4 | 1 | 0.0297 | 0.0822 | 7.16 | 19.71 | 671.79 |
| | | 2 | 0.0283 | 0.0821 | 6.81 | 19.73 | 2860.55 |
| | 5 | 1 | 0.0293 | 0.0839 | 6.98 | 19.87 | 1428.63 |

Table 4 shows that under the same cluster numbers, in most cases the adaptive model can find solutions with better out-of-sample performance in all indicators compared to the static model. When $N = 20$, under some cluster numbers, even though the adaptive model is not solved to optimality, it can still generate solutions with better performance. The static model performs better in indicators *MaxPro* and *MaxExp* under some cluster numbers, while the adaptive model demonstrates superiority in the two other indicators. In the previous section, we observe that the static model performs better in all indicators when $(K^1, K^2) = (5, 1)$ for $N = 15$ compared to Method ERI-E, whereas the adaptive model can further improve performance under multiple cluster numbers besides $(K^1, K^2) = (5, 1)$, *e.g.*, $(K^1, K^2) = (2, 2)$ and $(2, 3)$, etc. Likewise, for $N =$

**Table 4 Average results of out-of-sample tests generated by the adaptive and static DRO models**

| $N$ | $K^1$ | $K^2$ | MaxPro | SumPro | MaxExp | SumExp | CPU time |
|---|---|---|---|---|---|---|---|
| 15 | 2 | 1 | **0.0355 (0.0349)** | 0.0612 (0.0647) | 7.95 (8.05) | 14.07 (14.86) | 18.62 (9.90) |
| | | 2 | 0.0308 (0.0337) | 0.0590 (0.0652) | 7.41 (7.77) | 13.67 (14.65) | 57.03 (25.68) |
| | | 3 | 0.0312 (0.0364) | 0.0587 (0.0657) | 7.24 (7.83) | 13.81 (14.90) | 135.64 (56.56) |
| | 3 | 1 | 0.0326 (0.0340) | 0.0606 (0.0650) | 7.62 (7.93) | 13.89 (14.83) | 76.69 (26.41) |
| | | 2 | 0.0318 (0.0330) | 0.0602 (0.0646) | 7.54 (7.81) | 14.00 (14.77) | 277.58 (72.18) |
| | | 3 | 0.0320 (0.0325) | 0.0595 (0.0644) | 7.52 (7.83) | 13.74 (14.79) | 537.31 (93.55) |
| | 4 | 1 | **0.0335 (0.0334)** | 0.0612 (0.0650) | 7.62 (7.70) | 14.04 (14.88) | 189.63 (35.93) |
| | | 2 | 0.0325 (0.0340) | 0.0596 (0.0650) | 7.27 (7.71) | 13.68 (14.82) | 578.85 (89.10) |
| | 5 | 1 | 0.0321 (0.0329) | 0.0607 (0.0644) | 7.20 (7.53) | 13.94 (14.63) | 491.41 (65.45) |
| 20 | 2 | 1 | 0.0306 (0.0364) | 0.0814 (0.0889) | 7.31 (8.14) | 19.65 (20.66) | 336.44 (68.16) |
| | | 2 | **0.0342 (0.0300)** | 0.0812 (0.0838) | **7.91 (7.33)** | 19.14 (20.04) | 2920.24 (475.42) |
| | | 3 | 0.0299 (0.0343) | 0.0785 (0.0849) | 7.14 (7.69) | 19.00 (20.00) | 3174.50 (595.31) |
| | 3 | 1 | 0.0329 (0.0350) | 0.0782 (0.0855) | 7.70 (8.26) | 18.23 (20.12) | 3022.32 (265.55) |
| | | 2 | 0.0282 (0.0304) | 0.0755 (0.0830) | 7.13 (7.23) | 18.33 (19.82) | 3600.00 (891.86) |
| | | 3 | 0.0293 (0.0303) | 0.0773 (0.0836) | 6.98 (7.43) | 18.26 (19.85) | 3600.00 (1734.09) |
| | 4 | 1 | **0.0331 (0.0297)** | 0.0792 (0.0822) | **7.40 (7.16)** | 18.33 (19.71) | 3600.00 (671.79) |
| | | 2 | 0.0269 (0.0283) | 0.0783 (0.0821) | 6.64 (6.81) | 17.76 (19.73) | 3600.00 (2860.55) |
| | 5 | 1 | **0.0338 (0.0293)** | 0.0836 (0.0839) | **7.20 (6.98)** | 18.51 (19.87) | 3600.00 (1428.63) |

20, the performance of the adaptive model also dominates that of the static model in all indicators when $(K^1, K^2) = (4, 2)$. Another observation is that with a small number of clusters, both robust models outperform the empirical distribution; however, the out-of-sample performance does not necessarily improve with a larger number of clusters, because the ambiguity set would converge to the empirical distribution with increasing clusters, which may lead to overfitting. In practice, we can determine the number of clusters via validation tests.

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, which is 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. 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.

**Table 5 Detailed solutions of the DRO models for one instance**

| Delivery schedule in Period 1 | Adaptive | Static |
|---|---|---|
| | [4, 1, 2, 3], [5, 15, 6] | [4, 1, 2, 6], [5, 15, 3] |
| Centroid (Cartesian coordinate) | Delivery schedule in Period 2 | |
| | 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] | |

## 6. Conclusions

In this paper, we introduce a distributionally robust optimization 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. For greater scalability, we develop a branch-and-cut algorithm for the adaptive robust model. 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.

## References

Adulyasak, Yossiri, Jean-François Cordeau, Raf Jans. 2014. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* **26**(1) 103–120.

Agatz, Niels, Paul Bouman, Marie Schmidt. 2018. Optimization approaches for the traveling salesman problem with drone. *Transportation Science* **52**(4) 965–981.

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

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

Bertsimas, Dimitris, Vishal Gupta, Nathan Kallus. 2018. Data-driven robust optimization. *Mathematical Programming* **167**(2) 235–292.

Bertsimas, Dimitris, Melvyn Sim, Meilin Zhang. 2019. Adaptive distributionally robust optimization. *Management Science* **65**(2) 604–618.

Black, Thomas. 2017. The future of drone delivery depends on predicting the weather. Accessed February 15, 2020, `https://mashable.com/2017/06/22/drone-delivery-weather/`.

Bouman, Paul, Niels Agatz, Marie Schmidt. 2018. Dynamic programming approaches for the traveling salesman problem with drone. *Networks* **72**(4) 528–542.

Burns, Stephen. 2017. Drone meets delivery truck. Accessed February 15, 2020, `https://www.ups.com/us/es/services/knowledge-center/article.page?kid=cd18bdc2`.

Chassein, André, Trivikram Dokka, Marc Goerigk. 2019. Algorithms and uncertainty sets for data-driven robust shortest path problems. *European Journal of Operational Research* **274**(2) 671–686.

Chen, Zhi, Melvyn Sim, Peng Xiong. 2020. Robust stochastic optimization made easy with RSOME. *Management Science.* Forthcoming.

Cheng, Chun, Yossiri Adulyasak, Louis-Martin Rousseau. 2020. Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological* **139** 364–387.

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

Doherty, Jennifer. 2019. Alphabet's Wing begins first commercial drone delivery service in U.S., beating Amazon, Uber. Accessed December 6, 2019, `https://www.newsweek.com/wing-drone-first-commercial-delivery-1466471`.

Dorling, Kevin, Jordan Heinrichs, Geoffrey G Messier, Sebastian Magierowski. 2016. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **47**(1) 70–85.

Enderle, Rob. 2019. The 5 most pressing problems with drone delivery. Accessed February 15, 2020, `https://www.technewsworld.com/story/86060.html`.

Freitas, Júlia Cária de, Puca Huachi Vaz Penna. 2020. A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research* **27**(1) 267–290.

Gorissen, Bram L, İhsan Yanıkoğlu, Dick den Hertog. 2015. A practical guide to robust optimization. *Omega* **53** 124–137.

Ham, Andy M. 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, Zhaowei, Long He, Zhenyu Hu, Jun Jiang. 2020. Robust vehicle pre-allocation with uncertain covariates. *Production and Operations Management* **29**(4) 955–972.

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

Kim, Seon Jin, Gino J Lim, Jaeyoung Cho. 2018. Drone flight scheduling under uncertainty on battery duration and air temperature. *Computers & Industrial Engineering* **117** 291–302.

Kim, Seon Jin, Gino J Lim, Jaeyoung Cho, Murray J Côté. 2017. Drone-aided healthcare services for patients with chronic diseases in rural areas. *Journal of Intelligent & Robotic Systems* **88**(1) 163–180.

Lavars, Nick. 2015. Amazon to begin testing new delivery drones in the US. Accessed February 15, 2020, https://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/.

Lee, Dave. 2019. Amazon to deliver by drone 'within months'. Accessed February 15, 2020, https://www.bbc.com/news/technology-48536319.

Marinelli, Mario, Leonardo Caggiani, Michele Ottomanelli, Mauro Dell'Orco. 2017. En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems* **12**(4) 253–261.

Mehra, Gagan. 2015. 8 obstacles to drone delivery, for ecommerce. Accessed February 15, 2020, https://www.practicalecommerce.com/8-Obstacles-to-Drone-Delivery-for-Ecommerce.

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

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

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

Ning, Chao, Fengqi You. 2018. Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. *Computers & Chemical Engineering* **112** 190–210.

Otto, Alena, Niels Agatz, James Campbell, Bruce Golden, Erwin Pesch. 2018. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks* **72**(4) 411–458.

Perakis, Georgia, Melvyn Sim, Qinshen Tang, Peng Xiong. 2020. Robust pricing and production with information partitioning and adaptation. Working paper, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3305039.

Poikonen, Stefan, Bruce Golden, Edward A Wasil. 2019. A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing* **31**(2) 335–346.

Ponza, Andrea. 2016. Optimization of drone-assisted parcel delivery. Master's thesis, University of Padova, Italy.

Rabta, Boualem, Christian Wankmüller, Gerald Reiner. 2018. A drone fleet model for last-mile distribution in disaster relief operations. *International Journal of Disaster Risk Reduction* **28** 107–112.

Radzki, Grzegorz, Amila Thibbotuwawa, Grzegorz Bocewicz. 2019. Uavs flight routes optimization in changing weather conditionsconstraint programming approach. *Applied Computer Science* **15**(3) 5–20.

Rahimian, Hamed, Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. Working paper, https://arxiv.org/pdf/1908.05659.pdf.

Roberti, Roberto, Mario Ruthmair. 2019. Exact methods for the traveling salesman problem with drone. Working paper, `http://www.optimization--online.org/DB_FILE/2019/11/7457.pdf`.

Rose, Charlie. 2013. Amazon's Jeff Bezos looks to the future. Accessed February 15, 2020, `https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/`.

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

Thibbotuwawa, Amila, Grzegorz Bocewicz, Banaszak Zbigniew, Peter Nielsen. 2019. A solution approach for uav fleet mission planning in changing weather conditions. *Applied Sciences* **9**(19) 3972.

Vural, Danışment, Robert F Dell, Erkan Kose. 2019. Locating unmanned aircraft systems for multiple missions under different weather conditions. *Operational Research* 1–20.

Walker, Lauren. 2014. Drone delivery for Amazon and Google slowed by headwinds. Accessed February 15, 2020, `https://www.newsweek.com/will-wind-be-end-commercial-drone-delivery-amazon-and-google-275999`.

Wiesemann, Wolfram, Daniel Kuhn, Melvyn Sim. 2014. Distributionally robust convex optimization. *Operations Research* **62**(6) 1358–1376.

Yanıkoğlu, İhsan, Bram L Gorissen, Dick den Hertog. 2019. A survey of adjustable robust optimization. *European Journal of Operational Research* **277**(3) 799–813.

Zhang, Yu, Roberto Baldacci, Melvyn Sim, Jiafu Tang. 2019. Routing optimization with time windows under uncertainty. *Mathematical Programming* **175**(1-2) 263–305.

Zhang, Yu, Zhenzhen Zhang, Andrew Lim, Melvyn Sim. 2020. Robust data-driven vehicle routing with time windows. *Operations Research.* Forthcoming.

# Appendix A Proofs of Theorems

## A.1 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 (17) 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 $\mathcal{G}_k^1$:

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

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 (A.1) is equivalent to the following maximization problem:

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

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

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

$$
\begin{aligned}
&\boldsymbol{a}_j^1 \geq |\boldsymbol{u}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [2]\\
&\underline{\boldsymbol{v}}_k^1 \leq \boldsymbol{v}_j^1 \leq \bar{\boldsymbol{v}}_k^1 && \forall j \in [2]\\
&\boldsymbol{b}_j^1 \geq |\boldsymbol{v}_j^1 - \boldsymbol{\mu}_k^1| && \forall j \in [2]\\
&p_1, p_2 \geq 0 &&\\
&\boldsymbol{u}_j^1, \boldsymbol{a}_j^1, \boldsymbol{v}_j^1, \boldsymbol{v}_j^1 \in \mathbb{R}^N && \forall j \in [2].
\end{aligned}
$$

By perspective transformation, *i.e.*, $\boldsymbol{u}_1^1 p_1 \rightarrow \boldsymbol{u}_1^1$, $\boldsymbol{v}_1^1 p_1 \rightarrow \boldsymbol{v}_1^1$, we obtain the linear optimization model presented in Equation (18), 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 $p_j = 0$. □

## A.2 Proof of Theorem 2

The proof is similar to the proof of Theorem 1. Problem (20) is equivalent to the following minimization problem:

$$
\begin{aligned}
\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\\
\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\\
&\qquad \geq \sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)(u_i^1 + v_i^1) + \sum_{i \in \bar{\mathcal{C}}^2} \big((z_{id}^2(s) + x_{id}^2(s))(u_i^2 + v_i^2) + y_{id}^2(s) u_i^2\big) - \bar{\tau}^2\\
&\qquad\qquad\qquad\qquad\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_{kgs}^2\\
& (\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\\
&\qquad \geq \bar{\tau}^1 + \sum_{i \in \bar{\mathcal{C}}^2} \big(z_{id}^2(s) v_i^2 + x_{id}^2(s)(u_i^2 + v_i^2) + y_{id}^2(s) u_i^2\big) - \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_{kgs}^2\\
& (\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_d^2\\
&\qquad\qquad\qquad\qquad\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_{kgs}^2\\
& \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,

$$
\begin{aligned}
&\max_{(\boldsymbol{u}_j^1, \boldsymbol{v}_j^1, \boldsymbol{a}_j^1, \boldsymbol{b}_j^1, \boldsymbol{u}_j^2, \boldsymbol{v}_j^2, \boldsymbol{a}_j^2, \boldsymbol{b}_j^2) \in \Xi_{kgs}^2 \forall j \in [3]} \min \Big((\boldsymbol{\mu}_k^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{\nu}_k^1)'\boldsymbol{\beta}^1 + (\boldsymbol{\sigma}_k^1)'\boldsymbol{\delta}^1 + (\boldsymbol{\varsigma}_k^1)'\boldsymbol{\epsilon}^1\\
&\qquad\qquad\qquad + (\boldsymbol{\mu}_{sg}^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{\nu}_{sg}^2)'\boldsymbol{\beta}^2 + (\boldsymbol{\sigma}_{sg}^2)'\boldsymbol{\delta}^2 + (\boldsymbol{\varsigma}_{sg}^2)'\boldsymbol{\epsilon}^2 + \eta\Big)\\
&\text{s.t. } (\boldsymbol{u}_1^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}_1^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}_1^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}_1^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}_1^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}_1^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}_1^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}_1^2)'\boldsymbol{\epsilon}^2 + \eta\\
&\qquad \geq \left(\sum_{i \in \bar{\mathcal{C}}^1} (x_{id}^1 + y_{id}^1)\boldsymbol{e}_i\right)' (\boldsymbol{u}_1^1 + \boldsymbol{v}_1^1) + \left(\sum_{i \in \bar{\mathcal{C}}^2} (z_{id}^2(s) + x_{id}^2(s))\boldsymbol{e}_i\right)' (\boldsymbol{u}_1^2 + \boldsymbol{v}_1^2) + \left(\sum_{i \in \bar{\mathcal{C}}^2} y_{id}^2(s)\boldsymbol{e}_i\right)' \boldsymbol{u}_1^2 - \bar{\tau}^2\\
&\quad (\boldsymbol{u}_2^1)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}_2^1)'\boldsymbol{\beta}^1 + (\boldsymbol{a}_2^1)'\boldsymbol{\delta}^1 + (\boldsymbol{b}_2^1)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}_2^2)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}_2^2)'\boldsymbol{\beta}^2 + (\boldsymbol{a}_2^2)'\boldsymbol{\delta}^2 + (\boldsymbol{b}_2^2)'\boldsymbol{\epsilon}^2 + \eta
\end{aligned}
$$

$$
\begin{aligned}
&\geq \bar{\tau}^1 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{v}^2_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} x^2_{id}(s)\boldsymbol{e}_i\right)' (\boldsymbol{u}^2_2+\boldsymbol{v}^2_2) + \left(\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{u}^2_2 - \bar{\tau}^2 \\
&(\boldsymbol{u}^1_3)'\boldsymbol{\alpha}^1 + (\boldsymbol{v}^1_3)'\boldsymbol{\beta}^1 + (\boldsymbol{a}^1_3)'\boldsymbol{\delta}^1 + (\boldsymbol{b}^1_3)'\boldsymbol{\epsilon}^1 + (\boldsymbol{u}^2_3)'\boldsymbol{\alpha}^2 + (\boldsymbol{v}^2_3)'\boldsymbol{\beta}^2 + (\boldsymbol{a}^2_3)'\boldsymbol{\delta}^2 + (\boldsymbol{b}^2_3)'\boldsymbol{\epsilon}^2 + \eta \geq -\gamma^2_d \\
&\boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \geq 0 \qquad\qquad \forall n \in [2] \\
&\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n, \boldsymbol{\delta}^n, \boldsymbol{\epsilon}^n \in \mathbb{R}^N, \eta \in \mathbb{R} \qquad \forall n \in [2].
\end{aligned}
$$

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 \\
& + \left(\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{u}^2_1 p_1 - \bar{\tau}^2 p_1 + \bar{\tau}^1 p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} z^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{v}^2_2 p_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)p_2 + \left(\sum_{i\in\bar{\mathcal{C}}^2} y^2_{id}(s)\boldsymbol{e}_i\right)' \boldsymbol{u}^2_2 p_2 - \bar{\tau}^2 p_2 - \gamma^2_d p_3 \\
\text{s.t. } & \boldsymbol{u}^1_1 p_1 + \boldsymbol{u}^1_2 p_2 + \boldsymbol{u}^1_3 p_3 = \boldsymbol{\mu}^1_k \\
& \boldsymbol{v}^1_1 p_1 + \boldsymbol{v}^1_2 p_2 + \boldsymbol{v}^1_3 p_3 = \boldsymbol{\nu}^1_k \\
& \boldsymbol{a}^1_1 p_1 + \boldsymbol{a}^1_2 p_2 + \boldsymbol{a}^1_3 p_3 \leq \boldsymbol{\sigma}^1_k \\
& \boldsymbol{b}^1_1 p_1 + \boldsymbol{b}^1_2 p_2 + \boldsymbol{b}^1_3 p_3 \leq \boldsymbol{\varsigma}^1_k \\
& \boldsymbol{u}^2_1 p_1 + \boldsymbol{u}^2_2 p_2 + \boldsymbol{u}^2_3 p_3 = \boldsymbol{\mu}^2_{sg} \\
& \boldsymbol{v}^2_1 p_1 + \boldsymbol{v}^2_2 p_2 + \boldsymbol{v}^2_3 p_3 = \boldsymbol{\nu}^2_{sg} \\
& \boldsymbol{a}^2_1 p_1 + \boldsymbol{a}^2_2 p_2 + \boldsymbol{a}^2_3 p_3 \leq \boldsymbol{\sigma}^2_{sg} \\
& \boldsymbol{b}^2_1 p_1 + \boldsymbol{b}^2_2 p_2 + \boldsymbol{b}^2_3 p_3 \leq \boldsymbol{\varsigma}^2_{sg} \\
& p_1 + p_2 + p_3 = 1 \\
& \boldsymbol{u}^1_k \leq \boldsymbol{u}^1_j \leq \bar{\boldsymbol{u}}^1_k && \forall j \in [3] \\
& \underline{\boldsymbol{v}}^1_k \leq \boldsymbol{v}^1_j \leq \bar{\boldsymbol{v}}^1_k && \forall j \in [3] \\
& \boldsymbol{a}^1_j \geq |\boldsymbol{u}^1_j - \boldsymbol{\mu}^1_k| && \forall j \in [3] \\
& \boldsymbol{b}^1_j \geq |\boldsymbol{v}^1_j - \boldsymbol{\nu}^1_k| && \forall j \in [3] \\
& \underline{\boldsymbol{u}}^2_{sg} \leq \boldsymbol{u}^2_j \leq \bar{\boldsymbol{u}}^2_{sg} && \forall j \in [3] \\
& \underline{\boldsymbol{v}}^2_{sg} \leq \boldsymbol{v}^2_j \leq \bar{\boldsymbol{v}}^2_{sg} && \forall j \in [3] \\
& \boldsymbol{a}^2_j \geq |\boldsymbol{u}^2_j - \boldsymbol{\mu}^2_{sg}| && \forall j \in [3] \\
& \boldsymbol{b}^2_j \geq |\boldsymbol{v}^2_j - \boldsymbol{\nu}^2_{sg}| && \forall j \in [3] \\
& p_1, p_2, p_3 \geq 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^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}^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_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}^2p_2-\gamma^2_dp_3\\
\text{s.t. } & \boldsymbol{u}^1_1+\boldsymbol{u}^1_2+\boldsymbol{u}^1_3=\boldsymbol{\mu}^1_k\\
& \boldsymbol{v}^1_1+\boldsymbol{v}^1_2+\boldsymbol{v}^1_3=\boldsymbol{\nu}^1_k\\
& \boldsymbol{a}^1_1+\boldsymbol{a}^1_2+\boldsymbol{a}^1_3\le\boldsymbol{\sigma}^1_k\\
& \boldsymbol{b}^1_1+\boldsymbol{b}^1_2+\boldsymbol{b}^1_3\le\boldsymbol{\varsigma}^1_k\\
& \boldsymbol{u}^2_1+\boldsymbol{u}^2_2+\boldsymbol{u}^2_3=\boldsymbol{\mu}^2_{sg}\\
& \boldsymbol{v}^2_1+\boldsymbol{v}^2_2+\boldsymbol{v}^2_3=\boldsymbol{\nu}^2_{sg}\\
& \boldsymbol{a}^2_1+\boldsymbol{a}^2_2+\boldsymbol{a}^2_3\le\boldsymbol{\sigma}^2_{sg}\\
& \boldsymbol{b}^2_1+\boldsymbol{b}^2_2+\boldsymbol{b}^2_3\le\boldsymbol{\varsigma}^2_{sg}\\
& p_1+p_2+p_3=1\\
& \underline{\boldsymbol{u}}^1_kp_j\le\boldsymbol{u}^1_j\le\bar{\boldsymbol{u}}^1_kp_j && \forall j\in[3]\\
& \underline{\boldsymbol{v}}^1_kp_j\le\boldsymbol{v}^1_j\le\bar{\boldsymbol{v}}^1_kp_j && \forall j\in[3]\\
& \boldsymbol{a}^1_j\ge|\boldsymbol{u}^1_j-\boldsymbol{\mu}^1_kp_j| && \forall j\in[3]\\
& \boldsymbol{b}^1_j\ge|\boldsymbol{v}^1_j-\boldsymbol{\nu}^1_kp_j| && \forall j\in[3]\\
& \underline{\boldsymbol{u}}^2_{sg}p_j\le\boldsymbol{u}^2_j\le\bar{\boldsymbol{u}}^2_{sg}p_j && \forall j\in[3]\\
& \underline{\boldsymbol{v}}^2_{sg}p_j\le\boldsymbol{v}^2_j\le\bar{\boldsymbol{v}}^2_{sg}p_j && \forall j\in[3]\\
& \boldsymbol{a}^2_j\ge|\boldsymbol{u}^2_j-\boldsymbol{\mu}^2_{sg}p_j| && \forall j\in[3]\\
& \boldsymbol{b}^2_j\ge|\boldsymbol{v}^2_j-\boldsymbol{\nu}^2_{sg}p_j| && \forall j\in[3]\\
& p_1,p_2,p_3\ge 0.
\end{aligned}
$$

□