Noname manuscript No.
(will be inserted by the editor)

A Real-time Rerouting Method for Drone Flights under Uncertain Flight Time

Seon Jin Kim · Gino J. Lim

Received: date / Accepted: date

Abstract This paper proposes a method for real-time rerouting drone flights under uncertain flight times. The battery runtime that remains of a drone in real-time may be insufficient to complete its flight mission. This may be due to external factors, such as unexpected severe weather or obstacles that move into the drone's flight path. Under unexpected situations, such as these, the drone cannot safely return to its depot, as planned. To ensure that the drone makes a safe return and that the flight mission is a success, there must be a real-time rerouting process for a drone's flight path in response to unforeseeable circumstances. Hence, this paper proposes a real-time rerouting process consisting of two optimization models that generate an optimal alternative flight path for a drone that has insufficient remaining battery runtime. The first model is used to find an optimal flight path to visit all remaining target waypoints. If the first model fails to obtain a feasible solution, the second model is implemented to find an optimal flight path to minimize the number of unvisited waypoints. To confine the total flight (travel) time to the insufficient battery runtime, both models include the constraint associated with uncertain flight (travel) times between waypoints. To capture this uncertainty, this paper proposes a chance constrained programming (CCP) approach under the assumption of a known mean, variance, and flight time intervals. Numerical examples show how the proposed rerouting process works, and the CCP method results in more conservative solutions as compared to the deterministic approach.

Keywords: Chance constrained programming; Drone; Rerouting; Uncertain flight time

1 Introduction

Drones (unmanned aerial vehicles) are gaining attraction as a potential solution for applications such as healthcare delivery services, border patrol, asset monitoring, disaster relief, and object tracking [1–4]. Figure 1 shows two categories for drone use. These include delivery and monitoring (remote sensing).

In the case of delivery, drones can load required items within the maximum loading capacity (payload capacity) and deliver them to service areas regardless of road conditions. Numerous studies are underway to supply emergency items such as defibrillators, and/or simple commercial items to customers using drones instead of (or combined with) ground vehicles [5–8].

Another purpose is to monitor objects or situations (remote sensing), in which cameras and remote sensing devices are equipped with the drones [9]. Instead of persons or manned vehicles, drones may conduct various missions in dangerous areas or harsh environments. For example, volcanic activities, forest fires, and Arctic ecosystems can be monitored and tracked by drones [10–12].

For these purposes, most drone flight paths are planned prior to taking off. When scheduling the flight paths, one of the key factors in determining the flight duration (maximum possible flight time) is the battery runtime of the drone. A feasible flight path which ensures the safe return of the drones and the fulfillment of assigned flight mission can be developed based on the battery runtime and flight distances (in time) between waypoints. However, the planned feasible flight path may not be valid anymore if the drones consume more

Seon Jin Kim (Corresponding author)
Republic of Korea Army,
Daejeon, South Korea, 34079
E-mail: sonjin64@gmail.com

Gino J. Lim
Department of Industrial Engineering,
University of Houston, Houston, TX, 77204
E-mail: ginolim@uh.edu

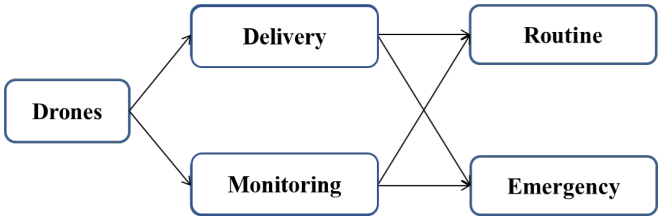


Fig. 1. Purposes of drone flights

battery power than planned for. This unintended and undesired case may occur due to unexpected harsh environmental conditions, such as strong winds or obstacles moving onto the flight path [13]. Such disturbances trigger excessive battery power consumption. As a result, the remaining battery runtime at a certain waypoint may be below the one planned for the flight mission. Therefore, the planned flight paths must be rerouted to ensure the safe return of the drones and minimize the unvisited flight waypoints (unmet demands).

For a real-time rerouting, there are three requirements for obtaining a feasible alternative flight path: 1) telemetry data of drone flight status; 2) a rerouting process for multi-drone flights, and 3) flight (travel) time reflecting real flight environment. The first requirement can be satisfied by using existing drone monitoring technologies. Various commercial drone flight control programs have been developed that provide the real-time status of drones, including the remaining battery runtime, flight altitude, flight distance, and radio transmission/reception status [14]. Thus, with the current technology, the telemetry data of drones can be obtained in real-time.

The second requirement is related to an autonomous flight control system. Traditionally, at least two ground crews (pilot and payload operator) are required to operate and control a drone, in real-time [15]. In the case of multiple drones needing alternative paths, it is difficult to provide each one with the alternative path that is manually operated by the ground crews. The autonomous flight control method is used to minimize the workload of the ground crews for multi-drone flights and rerouting time [16].

The third requirement is to delineate the real flight environment which affects flight times between waypoints. This could be fulfilled by considering environment conditions, such as winds, rain, air temperatures, and payload amount in the flight path optimization model. However, the real flight (travel) time may be different from a theoretical (estimated) flight time due to any kind of unforeseen weather patterns. The real flight time may be out of the forecasted value or range. Hence, fluctuations in the flight time need to be considered to ensure a feasible alternative path in real-time rerouting. Ignoring fluctuations (uncertainty) may result in failure to bring the drone back to depot and/or satisfy the demand.

Most of the research on drone flight scheduling focused on optimizing drone flight planning in advance [1–3, 5, 17, 18]. Recent research reports on drone flight scheduling problems considered the flight environment, including air temperature and wind speeds [17, 19]. Kim et al. [17] proposed a drone flight scheduling method that reflects changes in battery performance due to changes in the air temperature. A drawback of their work is its inability to reroute drones when any sort of unforeseeable circumstance (insufficient remaining battery runtime) occurs during the flight.

This paper aims to develop a real-time rerouting process to provide an alternative flight path for drones under uncertain flight times during the mission. As shown in Figure 2, when an unforeseen circumstance arises (i.e., the remaining battery runtime reaches a threshold value to terminate the flight), the proposed algorithm is used to find an alternative drone flight path in real-time. Hence, the approach enables the drones to be rerouted when it is necessary to ensure the safe return back to the depot while minimizing unmet demand during the flight.

To reflect uncertain environmental conditions, this paper proposes the use of a chance constrained programming (CCP) approach to handle uncertain flight times [20, 21]. The CCP finds a solution corresponding to the user-defined confidence level on constraint violation. It is commonly used to handle parameter uncertainty in a stochastic optimization model based on the decision maker's preference on the confidence level of the resulting solution [22]. The data in this paper show that the flight time varies in real time, and it follows a *Normal* or *Beta* distribution.

Overall, this work contributes to the existing body of literature as follows:

- To propose a real-time rerouting method for drones in flight. The proposed method provides the ability to reroute drones to ensure the safe return of drones to the depot and to minimize unmet demand (unvisited waypoints) in the case of insufficient battery runtime during the flight.
- To propose a CCP that can handle uncertainty in drone flight times between waypoints. This approach illustrates an uncertain nature of flight environment and enhances the feasibility of the developed alterna-

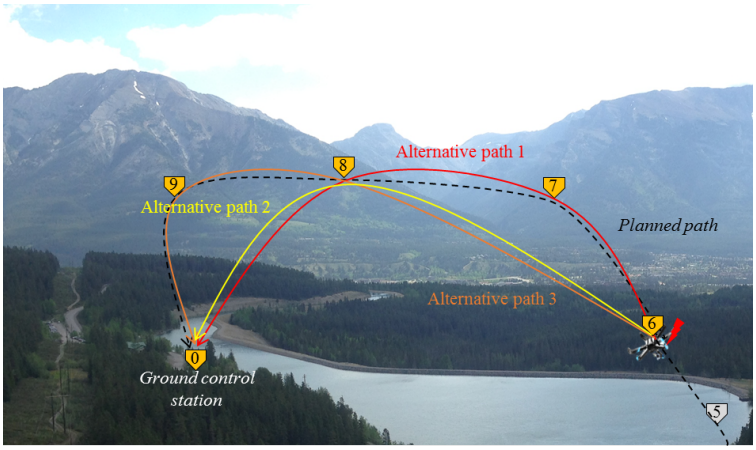


Fig. 2. A concept for real-time rerouting for drone flights under insufficient remaining battery runtime

tive flight paths. Moreover, this approach provides flexibility for decision-makers with a confidence level for each developed flight path.

The rest of this paper is organized as follows. Section 2 describes the problem in detail and presents the proposed rerouting process. Section 3 describes the notation and the two mathematical models for the problem. A CCP model is also defined based on the experimental data in Section 3. In Section 4, a case study is presented to illustrate the effectiveness of the proposed model in practice, and its results are discussed. Section 5 concludes the paper and suggests a potential extension of this work.

2 Problem Description

Consider a problem involving flying drones conducting a surveillance mission (monitoring: remote sensing) on a given area while minimizing: 1) the collision risk in the flight environment [23, 24] and 2) the detection risk during the flight. For the purpose of surveillance, the drones fly over geographically vulnerable areas where people or vehicles are not accessible. During the flight, it is possible that a drone may collide with nearby objects such as buildings, mountains, or trees. The risk also reflects a potential exposure to unfavorable conditions. For example, the purpose of a surveillance flight in a military operation is to collect information about enemy command posts, strike facilities, and detection facilities. Such missions may risk drones being exposed to a hostile environment. If they are detected, drones can be shot down by enemies.

Suppose that optimal drone flight paths have already been developed and the sequence of visits to corresponding waypoints is given prior to flight commencement. These waypoints consist of two types: 1) target waypoints and 2) intermediate waypoints. The target waypoints are the locations that must be visited to achieve the purpose of the flight, and the intermediate waypoints are the locations on the path to the target waypoints that are chosen to minimize the risk of colliding with ground objects located on a particular flight segment. If the target waypoints are not visited, a penalty cost is incurred, while the intermediate waypoints do not incur any cost for not being visited. A threshold value of the remaining battery for a safe return to a depot is given as an input. This will ensure that a drone will have sufficient battery runtime to fly back to the initial launching depot or the nearest depot from the current waypoint. As the drone commences its flight following the given path, the actual flight environment will be unfolded, and its associated data can be collected during the flight. The updated information can be useful in revising the flight path, if necessary. Figure 3 shows the proposed real-time rerouting strategy. This rerouting process is used to provide an alternative route to ensure a safe return to a depot, while minimizing unmet demand when the remaining drone battery runtime falls below a particular value (i.e., a threshold value at each waypoint to guarantee a safe return to a depot).

Using commercially available monitoring technologies, the real-time telemetry data of the drone flight, such as the remaining battery (flight) runtime of a drone and its location can be obtained. If the remaining battery runtime reaches a threshold value, then it must first be decided whether to immediately return to the launching point, seeing as the drone may not be able to visit at least one of the remaining waypoints (demands) with the remaining battery runtime. Figure 4 shows two possible strategies when a drone has a remaining battery runtime that is less than planned.

– Strategy 1 is to directly return to depot r in which a rerouting process is not needed to get an alternative path. If the remaining battery runtime (b_λ) is not sufficient to visit at least one of the remaining target waypoints, then the drone should directly return to depot r.

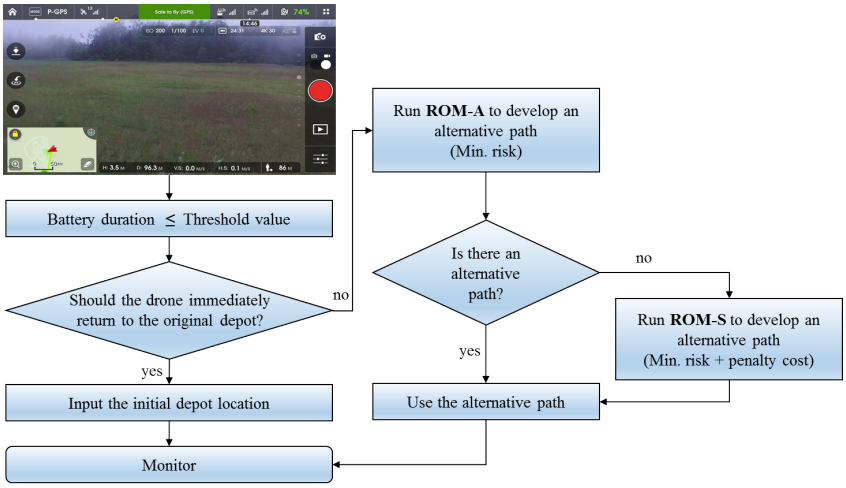


Fig. 3. Proposed rerouting process

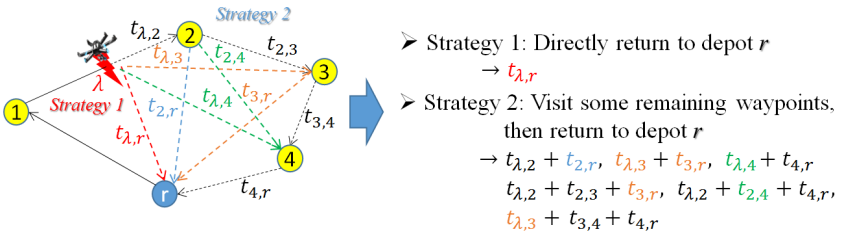


Fig. 4. Comparison of two strategies for insufficient battery runtime

– Strategy 2 is utilized when the drone is able to visit at least one of the remaining target waypoints before returning to depot r. In this case, the drone can visit target waypoint 3 ($t_{\lambda,3}$+$t_{3,r}$) or target waypoints 2 and 3 ($t_{\lambda,2}$+$t_{2,3}$+$t_{3,r}$). Hence, the rerouting process will be used to find a feasible alternative path.

The two strategies are expressed as below:

Algorithm 1 To decide whether a drone should return to its depot r directly

Inputs:
Initial launching depot: r
Location of a drone when a potential battery shortage is noticed (event): λ
Remaining waypoints when the event occurs: $j \in I$
Remaining battery runtime of drone k when the event occurs: b_λ
Flight distance (time) information between waypoints including λ: $t_{\lambda,j}$, $t_{\lambda,r}$, and $t_{j,r}$.
for all (remaining waypoints $j \in I$) **do**
  if $t_{\lambda,r} \leq b_\lambda < min(t_{\lambda,j} + t_{j,r})$ **then**
    Return to its initial depot r directly.
  else
    Run optimization model, ROM-A
  end if
  if ROM-A : infeasible **then**
    Run optimization model, ROM-S
  end if
end for

Given a set of input data, the algorithm checks if the drone can visit only one of the remaining target waypoints and decides whether to implement the rerouting process ($t_{\lambda,r} \leq b_\lambda < min(t_{\lambda,j} + t_{j,r})$). As a result, we can avoid unnecessary efforts to find an alternative path when the remaining battery runtime is not sufficient enough to do so.

If the first decision leads to Strategy 2 in Figure 4, then the proposed rerouting process is triggered and runs the rerouting optimization model for visiting all remaining target waypoints (ROM-A). The goal of ROM-A is

to find an optimal alternative flight path for the drone to visit all remaining target waypoints while minimizing the risk of the alternative path.

The ROM-A, however, might not be able to get a feasible alternative flight path to visit all remaining target waypoints due to insufficient battery runtime. For this case, we implement another mathematical optimization model, which is the rerouting optimization model for visiting some of remaining target waypoints (ROM-S). The ROM-S is used to develop an optimal alternative flight path for the drone for a given battery runtime, in which the optimal flight path minimizes the penalty cost for not visiting some target waypoints as well as the risk of the alternative path. Both of these models are constructed using the CCP method. This proposed process ensures that the drone returns safely to the initial launching point (or the nearest depot) after finishing its updated flight schedule when any sort of unforeseen circumstance arises.

3 Problem Formulation

This section explains, in detail, the two optimization models discussed in Section 2. Section 3.1 describes ROM-A, and Section 3.2 describes ROM-S. In Section 3.3, the proposed CCP method is developed to deal with the random uncertainty in the flight times of drones. To capture the properties of uncertain flight times, a set of data was obtained by flying a drone in a real flight environment.

The two optimization models are developed using the following notation.

Indices

I	Set of all waypoints (i.e, $i, j, u, \lambda \in I$),
R	Set of center (depot) waypoints ($r \in R$),
IW	Set of remaining intermediate waypoints (iw),
TW	Set of remaining target waypoints (tw).

Parameters

b_λ	Remaining battery runtime at location λ,
$\tilde{t}_{i,j}$	Random flight time between waypoints (i, j),
$R_{i,j}$	Risk when flying from waypoint i to j,
P_{tw}	Penalty cost when tw is not visited,
a	Weight factor associated with the risk level,
b	Weight factor associated with the penalty cost.

Decision Variables Our aim is to determine a near optimal flight path minimizing the risk (*and penalty cost*) on the path to visit all (*or some*) of the remaining target waypoints within flight duration b_λ. Accordingly, we define decision variables as follows:

$x_{i,j}$	1 if a drone flies from waypoint i to j, 0 otherwise,
h_{tw}	1 if target waypoint tw is not visited, 0 otherwise,
μ_i	The visiting sequence of waypoint i on a path.

3.1 Rerouting Optimization Model for visiting "all" remaining target waypoints: ROM-A

The ROM-A is expressed as follows:

$$\text{Minimize} \quad z = \sum_{i\in(I\cup\{\lambda\})}\sum_{j\in(I\cup R)} x_{i,j}\times R_{i,j}, \tag{1}$$

$$\text{Subject to:} \quad \sum_{i\in(I\cup\{\lambda\})} x_{i,tw} = 1,\ \ \forall tw \in TW \tag{2}$$

$$\sum_{j\in(I\cup R)} x_{tw,j} = 1,\ \ \forall tw \in TW \tag{3}$$

$$\sum_{j\in I} x_{\lambda,j} = 1, \lambda \in I \tag{4}$$

$$\sum_{i\in I} x_{i,r} \le 1,\ \ \forall r \in R \tag{5}$$

$$\sum_{i\in(I\cup\{\lambda\})} x_{i,j} \le 1,\ \ \forall j \in IW \tag{6}$$

$$\sum_{j\in(I\cup R)} x_{i,j} \le 1,\ \ \forall i \in IW \tag{7}$$

$$\sum_{i\in(I\cup\{\lambda\})} x_{i,u} - \sum_{j\in(I\cup R)} x_{u,j} = 0,\ \ \forall u \in I \tag{8}$$

$$\sum_{i\in(I\cup\{\lambda\})}\sum_{j\in(I\cup R)} x_{i,j}\times \tilde{t}_{i,j} \le\ b_\lambda, \lambda \in I \tag{9}$$

$$\mu_i - \mu_j + n\times x_{i,j} \le\ n-1,\ \ \forall i,j \in I \tag{10}$$

$$x_{i,i} = 0, x_{i,j} \in \{0,1\}, \mu_i \ge 0, n = |I|, m = |R|.$$

The objective function (1) is to minimize the total risk ($\sum_{i\in(I\cup\{\lambda\})}\sum_{j\in(I\cup R)} x_{i,j}\times R_{i,j}$) of the alternative path. Constraints (2) and (3) ensure that all remaining target waypoints (tw) are visited once by the drone. Constraint (4) defines the initial launching point of the alternative flight path that should be location λ, where the remaining battery runtime reaches a threshold value (i.e., the drone has insufficient battery runtime). After finishing the flight mission, the drone should return to the initial launching point or to the nearest depot (Constraint (5)). The intermediate waypoints (iw) may or may not be visited by the drone depending on the remaining battery runtime and the risk (constraints (6) and (7)). Constraint (8) defines the flow conservation of the drone flight, and Constraint (9) ensures that flight time of the alternative path should be within the remaining battery runtime (b_λ). Constraint (10) is used to eliminate sub-tours, which are incomplete flight paths [25–27].

3.2 Rerouting Optimization Model for visiting "some" remaining target waypoints: ROM-S

The ROM-A may not result in a feasible solution for visiting all remaining waypoints within the remaining battery runtime (b_λ). For this case, the ROM-S is proposed to maximize the utilization of remaining battery runtime b_λ, which means to minimize the total penalty cost caused by not visiting some of the remaining target waypoints.

The ROM-S is expressed as follows:

$$\text{Minimize} \quad z = a\times \sum_{i\in(I\cup\{\lambda\})}\sum_{j\in(I\cup R)} x_{i,j}\times R_{i,j} + b\times \sum_{tw\in TW} P_{tw}\times h_{tw}, \tag{11}$$

$$\text{Subject to:} \quad \sum_{i\in(I\cup\{\lambda\})} x_{i,tw} \le 1,\ \ \forall tw \in TW \tag{12}$$

$$\sum_{j\in(I\cup R)} x_{tw,j} \le 1,\ \ \forall tw \in TW \tag{13}$$

$$\sum_{i\in(I\cup\{\lambda\})} x_{i,tw} = 1 - h_{tw},\ \ \forall tw \in TW \tag{14}$$

and *constraints* (4) – (10).

The objective function (11) is to minimize the total penalty cost ($\sum_{tw \in TW} P_{tw} \times h_{tw}$) as well as the total risk of the alternative flight path, in which a is a parameter that represents the importance of the risk and b is a parameter that represents the importance of the penalty cost. These two parameter values are determined by the decision makers' preference. In this model, constraints (12) and (13) are used rather than constraints (2) and (3) because the drone visits some, but not all, remaining target waypoints. Constraint (14) defines which remaining target waypoints are visited by the drone. The rest of the constraints in the ROM-S correspond to constraints (4)-(10) in the ROM-A.

3.3 A CCP method

Constraint (9) in both the ROM-A and the ROM-S models has a random parameter, $\tilde{t}_{i,j}$, which is the flight time between waypoints. The random uncertainty in flight time results from the external flight environment. As described in Section 1, the CCP method is used to mitigate the random uncertainty in the flight time. From [20] and [21], we can reformulate Constraint (9) as a chance constraint with constraint violation level ϵ as below,

$$P\Big(\sum_{i \in (I \cup \{\lambda\})} \sum_{j \in (I \cup R)} x_{i,j}\tilde{t}_{i,j} - \ b_\lambda \leq 0\Big) \geq 1 - \epsilon. \tag{15}$$

This chance constraint would guarantee that the solution is feasible at the $1-\epsilon$ confidence level (i.e., the minimum probability that total flight time of a path would not exceed the remaining battery runtime).

To get the information and patterns on the probability distribution P on flight times, experiments are carried out with a Phantom 4 Pro by DJI [28] (Section 3.3.1). In Section 3.3.2, we constructed deterministic counterparts of Constraint (15) for each probability distribution obtained from the experiments. The deterministic counterpart is used to reformulate the CCP model (Constraint (15)) with an equivalent deterministic constraint, which is tractable. Section 3.3.3 explores robust chance constraints for cases where this probability distribution information is not specified [29, 30].

3.3.1 Data Collection

Experiments were performed to collect flight times of a drone (Phantom 4 Pro) in a real environment (Figure 5). The experimental conditions are shown in Table 1.

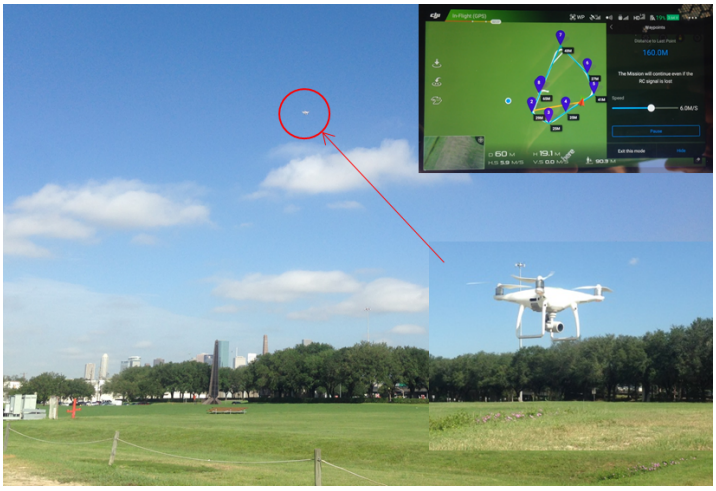


Fig. 5. Experiments to obtain real flight time data

A drone flight test was carried out for three days (including multiple flights per day) to obtain the necessary data. The time required to fly 264 m while keeping the drone's flight speed of 6 m/s and altitude (19 m) constant was measured. Based on the data collected, we analyzed the probability distribution of the flight time (by date) as shown in Table 2.

The data on Day 1 shows that the flight time is normally distributed (with mean 48 *seconds* and standard deviation 0.715) whereas the data on Day 2 and Day 3 follow a *Beta* distribution. Both of the *Beta* distributions

Table 1. Experiment setup and environment

	Day 1 (Oct. 4, 2017)	Day 2 (Oct. 5, 2017)	Day 3 (Oct. 6, 2017)
Flight speed	6 m/s	6 m/s	6 m/s
Flight altitude	19 m	19 m	19 m
Flight distance	264 m	264 m	264 m
Temperature	27.1 - 28.5 °C	26.7 - 28.8 °C	27.3 - 29.0 °C
Maximum wind speed	4.2 m/s	2.7 m/s	2.1 m/s
Number of experiments	30	58	56

Table 2. Experimental results

	Day 1 (Oct. 4, 2017)	Day 2 (Oct. 5, 2017)	Day 3 (Oct. 6, 2017)	Overall
*Distribution**	*Normal*	*Beta*	*Beta*	*Normal*
Mean	48 seconds	47.5 seconds	47.4 seconds	47.6 seconds
Standard deviation	0.715	0.355	0.361	0.51

are asymmetric, and the skewness values are -0.0697 (*Beta I*: skewed to the left with α_I=10.8, β_I=9.14) on Day 2 and 0.0827 (*Beta II*: skewed to the right with α_{II}=2.2, β_{II}=2.47) on Day 3, respectively. The result of the analysis of overall data shows that the flight time follows $Normal$ distribution. Through the experiments, the flight time distribution or fluctuation in the travel time is specified for the CCP method.

3.3.2 Under known distributions

First, by using the distribution information shown in Table 2, the flight time ($\tilde{t}_{i,j}$) in Constraint (15) is assumed to be distributed according to the known distributions ($Normal$ and $Beta$). We refer to the deviation in the *Normal* distribution and the shapes of the two *Beta* distributions (α_I & β_I and α_{II} & β_{II}). Under the assumption that the flight time distribution of a drone follows a $Normal$ or $Beta$ distribution, the Constraint (15) can be converted into a deterministic approximation for computational efficiency [22, 30].

Proposition 1 *Assume the flight time ($\tilde{t}_{i,j}$) is normally distributed and $\hat{t}_{i,j}$ is the expected value of $\tilde{t}_{i,j}$, then Constraint (15) is equivalent to the following inequality:*

$$\Phi^{-1}(1-\epsilon) \times \sqrt{\sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}^2 Var(\tilde{t}_{i,j})} + \sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}\hat{t}_{i,j} \le b_\lambda, \tag{16}$$

Proof See [22, 30] for details.

Proposition 2 *Assume the flight time ($\tilde{t}_{i,j}$) is distributed according to the Beta distribution and $\hat{t}_{i,j}$ is the expected value of $\tilde{t}_{i,j}$, then Constraint (15) is equivalent to the following inequality:*

$$\sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} [(1-\Psi_{beta}^{-1}(\epsilon)) \times \underline{\tilde{t}_{i,j}} + \Psi_{beta}^{-1}(\epsilon) \times \overline{\tilde{t}_{i,j}}] \times x_{i,j} \le b_\lambda, \quad \underline{\tilde{t}_{i,j}} \le \tilde{t}_{i,j} \le \overline{\tilde{t}_{i,j}} \tag{17}$$

Proof See [31] for details.

3.3.3 Under Uncertain Distributions

In practice, it is very challenging to evaluate the exact probability distribution P. The probability distribution of flight time data may not follow the two distributions outlined above, nor be specified. This is one of the disadvantages in using the CCP method [29]. In these cases, finite empirical data are used to estimate the mean and variance of the flight time and assumed to belong to an entire family of probability distributions [29, 30]. We refer to the deviation and the intervals of collected data. Constraint (15) can be expressed as below:

Table 3. Parameters applied in the case study

Parameter	Value	Unit
Penalty cost (P_{tw})	$tw_1 = 20$ $tw_2 = 30$ $tw_3 = 50$ $tw_4 = 70$ $tw_5 = 60$	$(U.S. Dollar)
Remaining flight duration at waypoint λ	35	minute
Threshold value at waypoint λ	35	minute

Proposition 3 *When the distribution is unknown, but the first two moments (mean $\hat{t}_{i,j}$ and variance $Var(\tilde{t}_{i,j})$) are known, then Constraint (15) is equivalent to the following inequality:*

$$\sqrt{(1-\epsilon)/\epsilon} \times \sqrt{\sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}^2 Var(\tilde{t}_{i,j})} + \sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}\hat{t}_{i,j} \leq b_\lambda, \tag{18}$$

Proof See [30] for details.

Proposition 4 *Assume random data have known mean ($\hat{t}_{i,j}$) and the individual data are only known to be placed independent bounded intervals ($\in [\gamma_{i,j}^-, \gamma_{i,j}^+]$), then Constraint (15) is equivalent to the following inequality:*

$$\sqrt{(1/2)ln(1/\epsilon)} \times \sqrt{\sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}^2 (\gamma_{i,j}^+ - \gamma_{i,j}^-)^2} + \sum_{i\in(I\cup\{\lambda\})} \sum_{j\in(I\cup R)} x_{i,j}\hat{t}_{i,j} \leq b_\lambda, \tag{19}$$

Proof See [30] for details.

4 Numerical Experiments

This section consists of three parts. In Section 4.1, we describe a case study to demonstrate how the proposed rerouting process works. Using the case study, we test the proposed rerouting models and analyze the results in Section 4.2. In Section 4.3 various scenarios on the weight parameters (a and b) in the objective function of the ROM-S are explored to analyze the sensitivity of the parameters to the optimal solution and the objective value. The two proposed models are implemented in GAMS [32] and solved by CPLEX 12.6.2 [33], and all experiments are performed on a server running RedHat Linux 64-bit with an Intel Xeon processor and 16GB of RAM.

4.1 A Case Study

For numerical experiments, we introduce the case study shown in Figure 6. Suppose that a drone is supposed to visit target waypoints following the black dashed line, as shown, and return to the initial launching depot (c_1). However, during its flight, it encountered an some sort of unforeseen circumstance (i.e., the current remaining battery runtime reaches the corresponding threshold value) at location w_λ with five remaining target waypoints (tw_1 - tw_5) to be visited. In addition, there are two more possible returning waypoints (depots c_2 and c_3) as well as seven intermediate waypoints (iw_1 - iw_7) to reduce the collision risk with ground objects in flight segments (Section 2).

As described in Section 2, each target waypoint is associated with a penalty cost if it is not visited during the flight mission. The specific penalty cost values are summarized in Table 3. When the drone has 35 *minutes* of battery runtime remaining (which is within the threshold value), an alternative flight path must be developed to ensure the safe return of the drone and minimize the collision risk level of the flight path and the penalty cost for not visiting target waypoints.

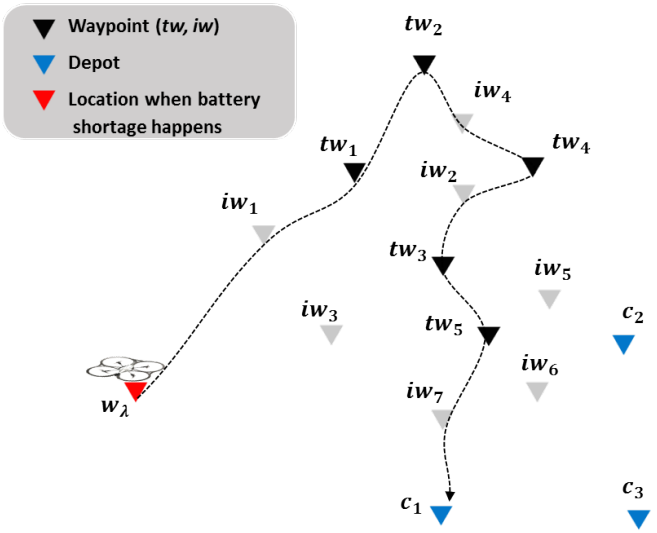


Fig. 6. A case study

4.2 Numerical results and discussions

Using the case study, the proposed four deterministic counterparts of the CCP model (i.e., *Normal* distribution, *Beta* distribution, only known mean and variance, and only known and its intervals in Section 3.3) were solved. Actual solving time for all cases is negligible. The detail results are shown in Table 4 and Figure 7.

First, under the *Normal* distribution assumption, the ROM-A found an optimal alternative flight path for the drone at 10% and 5% of ϵ. The risk of the developed alternative path increases from 1.5 to 1.7 as the violation level ϵ decreases from 10% to 5%. To ensure a higher satisfaction level for the safe return of the drone after completing its flight mission, the drone is required to fly with a shorter path without visiting iw_4 as shown in Figure 7(a). However, at violation level $\epsilon = 1\%$ (satisfaction level 99%), there is no feasible alternative flight path to guarantee the safe return of the drone after visiting all five remaining target waypoints with at least 99% confidence level. Therefore, the ROM-S instead of the ROM-A was utilized to find an optimal alternative flight path. The drone only visits four remaining target waypoints except tw_2. The corresponding risk of the developed flight path by the ROM-S is 1.3 and the corresponding penalty cost for not visiting tw_2 is \$30.

Second, the proposed process was tested under the *Beta* distribution, in which the *Beta I* and *Beta II* distributions were applied (see Section 3.3.1). Under the *Beta I* distribution, the risk of all the developed alternative flight paths is 1.5 at $\epsilon = 10\%$, 5%, and 1% (Figure 7(b)), whereas under the *Beta II* distribution, the risk increases from 1.5 to 1.8 as the satisfaction level (1 - ϵ) increases from 90% to 99% (Figure 7(c)). The reason for the objective values under the *Beta I* distribution remaining constant for the three different settings is because of the different shapes of the distributions. As shown in Figure 8, the probability density function (PDF) of the *Beta I* distribution (Figure 8(a)) is narrower in width than the PDF of the *Beta II* distribution (Figure 8(b)). The *Beta I* distribution has a lower variance than the *Beta II* distribution. The values of $\Psi^{-1}_{beta}(\epsilon)$ under the *Beta I* distribution range from 0.29 to 0.40 in the different values of ϵ (0.1, 0.05, and 0.01) as shown in Figure 8(c), whereas the values of $\Psi^{-1}_{beta}(\epsilon)$ under the *Beta II* distribution range from 0.06 to 0.19 as shown in Figure 8(d). Hence, the optimal values under the *Beta I* do not vary in the different values of ϵ (0.1, 0.05, and 0.01).

Next, we tested the proposed process under the assumption that the mean and variance of the flight time are known instead of the specified distribution information. To visit all five remaining target waypoints under this assumption, the allowable minimum violation level (ϵ) is 17%, which is the highest violation level of all experiments. At violation levels below 17% ($\epsilon < 17\%$), the ROM-S was implemented to develop an optimal alternative flight path which minimizes the penalty cost as well as the risk. At $\epsilon = 1\%$, the drone only visits two target waypoints (tw_4 and tw_5) and the corresponding penalty cost for not visiting target waypoints (tw_1, tw_2, and tw_3) is \$100 (Figure 7(d)).

Last, under the assumption that the independent intervals of the flight times, as well as the mean value are known, the allowable minimum violation level of (ϵ) is 12% to visit the five remaining target waypoints and safely return to a depot (c_2). The ROM-S was also triggered to get an optimal alternative flight path when the violation level is less than 12% ($\epsilon < 12\%$). The specific paths are shown in Figure 7(e).

Table 4. Results from case study

Violation level: ϵ (Satisfaction level: 1-ϵ)	Results	
	Risk (Penalty cost)	Optimal flight path
Under *Normal* distribution		
10% (90%)	1.5	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
5% (95%)	1.7	$w_\lambda \to tw_1 \to tw_2 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1% (99%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.3 (30)	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
Under *Beta I* distribution		
10% (90%)	1.5	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
5% (95%)	1.5	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1% (99%)	1.5	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
Under *Beta II* distribution		
10% (90%)	1.5	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
5% (95%)	1.7	$w_\lambda \to tw_1 \to tw_2 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1% (99%)	1.8	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to tw_3 \to tw_5 \to c_2$
Under uncertain distribution with known mean and variance		
17% (83%)	2.0	$w_\lambda \to tw_1 \to tw_2 \to tw_4 \to tw_3 \to tw_5 \to c_2$
10% (90%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.4 (30)	$w_\lambda \to tw_1 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
5% (95%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.5 (30)	$w_\lambda \to tw_1 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1% (99%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	0.8 (100)	$w_\lambda \to iw_3 \to tw_5 \to iw_5 \to tw_4 \to c_2$
Under uncertain distribution with known independent intervals		
12% (88%)	2.0	$w_\lambda \to tw_1 \to tw_2 \to tw_4 \to tw_3 \to tw_5 \to c_2$
10% (90%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.3 (30)	$w_\lambda \to tw_1 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to iw_7 \to c_1$
5% (95%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.4 (30)	$w_\lambda \to tw_1 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1% (99%)	*Infeasible*	*No solution*
$\Rightarrow$ *ROM-S**	1.4 (30)	$w_\lambda \to tw_1 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
Under deterministic assumption		
	1.3	$w_\lambda \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to iw_7 \to c_1$
Initial flight path		
	1.2	$w_\lambda \to iw_1 \to tw_1 \to tw_2 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to iw_7 \to c_1$

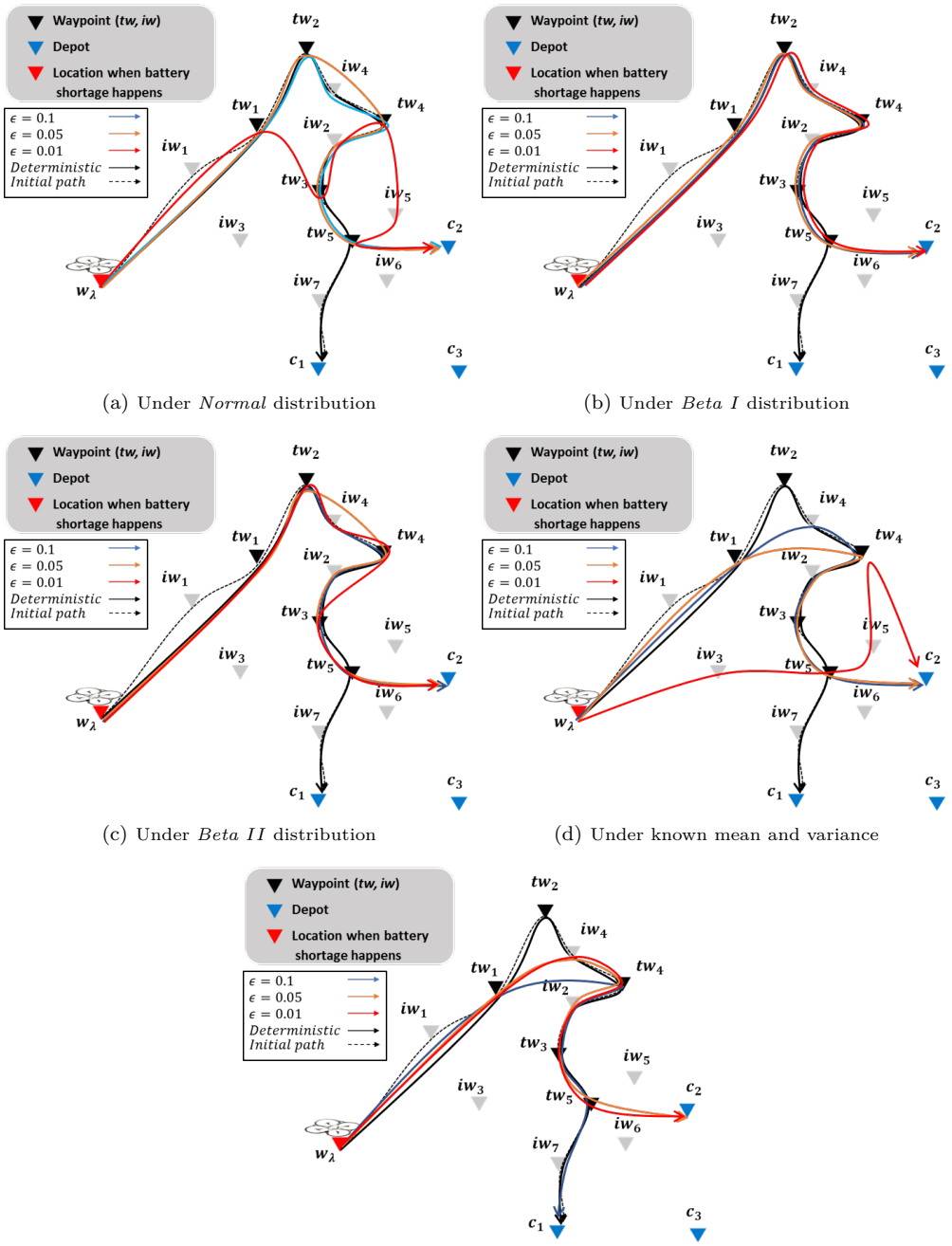
* : a=b=1

Through these numerical results, the CCP method provides more conservative solutions compared to the deterministic assumption. The results from the ROM-S were obtained under the assumption that the weighted importance value for the risk in the objective function of the ROM-S is 1 ($a = 1$) and the value for the penalty cost is also 1 ($b = 1$). In Section 4.3, we will analyze the optimal values under various values of the two parameters (a and b).

4.3 Sensitivity analysis on weight parameters

The objective function (11) in the ROM-S (Section 3.2) consists of two terms: the first term is to minimize the risk, and the second term is to minimize the penalty cost of the developed alternative flight path. These are conflicting goals, in principle. If minimizing the risk is the sole focus, it may not be feasible to accomplish the flight mission due to there being a significant number of unvisited target waypoints. Conversely, if minimizing the penalty cost is the sole focus (visiting as many waypoints as possible), the alternative flight path may be at a greater risk of a collision or a detection.

The sensitivity analysis was conducted by changing the weighted importance values on the objective function in the ROM-S in the case of satisfaction level 99% (violation level: $\epsilon = 1\%$) under the *Normal* distribution was conducted in this section. The results are shown in Table 5, in which we first changed the value of a under the condition that the value of b is fixed at 1 ($b = 1$), and vice versa.



(a) Under *Normal* distribution

(b) Under *Beta I* distribution

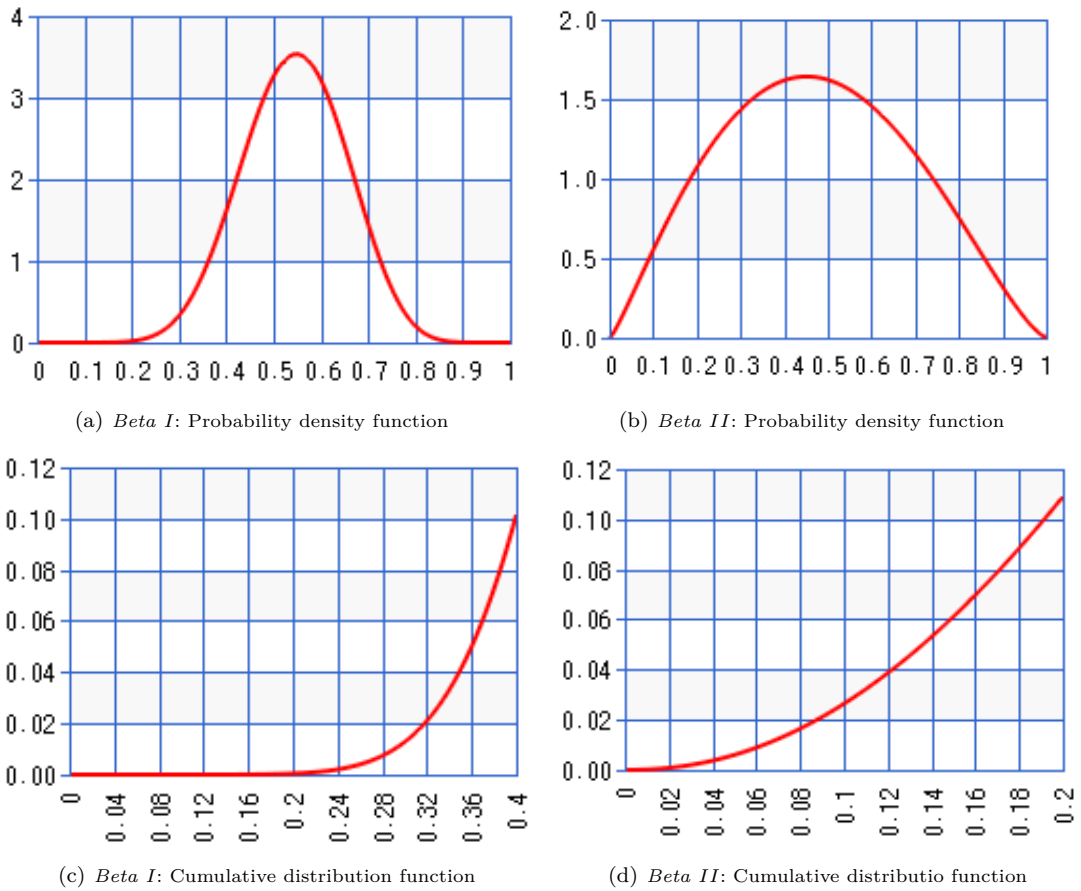
(c) Under *Beta II* distribution

(d) Under known mean and variance

(e) Under known independent intervals

Fig. 7. Specific flight paths from case study

When the value of b was set to 1, the risk value corresponding to alternative flight paths remained the same (*i.e.*, 1.3) until the value of a was increased beyond 40. Between the value of 50 and 100, the risk value dropped to 0.8, whereas the penalty cost increased from 30 to 50. To minimize the weighted sum (Weighted sum = $a \times$ Risk + $b \times$ Penalty cost) when $a \geq 40$, corresponding alternative paths dropped to visiting only two target waypoints (tw_1 and tw_2) from the original plan. If the alternative path with $a = 50$ were to be the same as the path developed at $a < 40$ (i.e., not visiting only tw_2), the corresponding weighted sum would have been 95 ($95 = 50 \times 1.3 + 1 \times 30$), which is greater than the weighed sum of the optimal alternative path as shown in Figure 9 (Projected objective value = 95 *vs* Realized objective value = 90). This means that when the value of a is 40 or higher, the risk is relatively more important than the penalty cost in the remaining flight mission. In other words, the flight safety of the drone is more important than completing the remaining flight mission. This was expected due to the scale difference between the risk value and the penalty cost.



(a) *Beta I*: Probability density function

(b) *Beta II*: Probability density function

(c) *Beta I*: Cumulative distribution function

(d) *Beta II*: Cumulative distributio function

Fig. 8. Different shapes of the *Beta* distribution

Table 5. Results from ROM-S at $\epsilon = 1\%$ under a *Normal* distribution

Weights		Objective function value			Optimal flight path
a	b	Risk	Penalty cost	Weighted sum	
1	1	1.3	30	31.3	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
2	1	1.3	30	32.6	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
5	1	1.3	30	36.5	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
10	1	1.3	30	43	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
20	1	1.3	30	56	$w_\lambda \to iw_1 \to tw_1 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
30	1	1.3	30	69	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
40	1	1.3	30	82	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
50	1	0.8	50	90	$w_\lambda \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
60	1	0.8	50	98	$w_\lambda \to iw_3 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
70	1	0.8	50	106	$w_\lambda \to iw_3 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
80	1	0.8	50	114	$w_\lambda \to iw_3 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
90	1	0.8	50	122	$w_\lambda \to iw_3 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
100	1	0.8	50	130	$w_\lambda \to iw_3 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to iw_6 \to c_1$
1	2	1.3	30	61.3	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$
1	5	1.3	30	151.3	$w_\lambda \to iw_1 \to tw_1 \to iw_4 \to tw_4 \to iw_2 \to tw_3 \to tw_5 \to c_2$
1	10	1.3	30	301.3	$w_\lambda \to tw_1 \to tw_3 \to iw_2 \to tw_4 \to iw_5 \to tw_5 \to c_2$

On the other hand, when the case is that the value of b is altered with a fixed value of a ($a = 1$), the risk and the penalty cost did not change even when the value of b increased beyond 10. Only the weighted sum increased proportionately to the value of b. In this case study, the scale of the penalty cost values dominated the scale of the risk values when $a = b = 1$. Based on the scales of the given values for the collision risk and the penalty cost, this approach can help the decision makers find appropriate values for a and b considering their decision criteria.

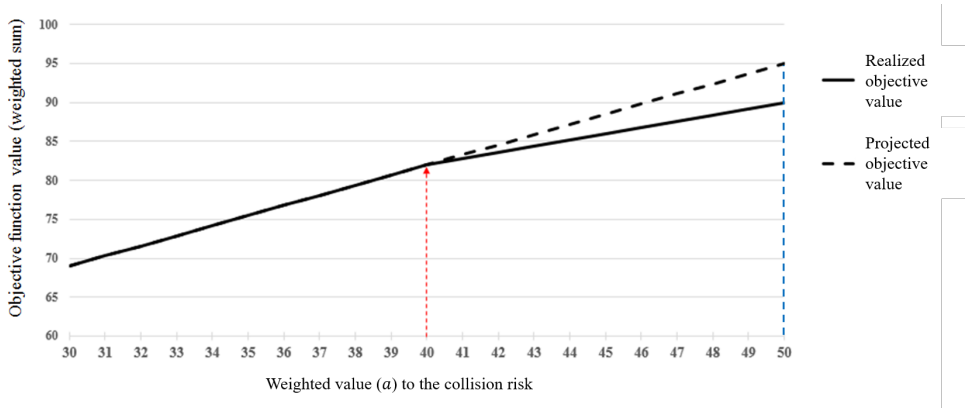


Fig. 9. Changes of the objective function value in the ROM-S when b=1

5 Conclusion

This paper introduced a method for the real-time rerouting of drone flights under uncertain flight time. To ensure the safe return of the drone and minimize the risk of setback of the purpose of the drones' missions in the case of an unforeseen circumstance (insufficient remaining battery runtime), the proposed rerouting process features two rerouting optimization models (ROM-A and ROM-S) used to develop an optimal alternative flight path for completing the remaining flight. Once the drone starts its flight and a rerouting decision is triggered (due to the occurrence of an unexpected event), the ROM-A was used to find an alternative flight path for visiting all remaining target waypoints based on the original plan that was given. If the ROM-A fails to find a feasible solution, the ROM-S was triggered to generate an alternative flight path, minimizing the unmet demand (not visiting target waypoints). Both models include the constraint to consider uncertain flight (travel) times between waypoints. To capture the flight time uncertainty, a CCP method was developed and its deterministic counterparts were derived to lower the computational burden for solving the CCP model under the *Normal* and *Beta* distributions as well as under unknown distribution information (solely known mean, variance, or intervals of the flight times). The proposed models were tested using a case study. The results showed that the flight paths developed based on CCP carries a higher risk value than the deterministic model when all remaining target waypoints were forced to be visited. When the risk value was fixed at 1.2, which is the value obtained from the deterministic model, the CCP-based flight path skipped some waypoints to reduce the risk of being detected. Overall, the CCP produced more conservative plans than the deterministic approach due to the travel time uncertainty and the detection risk. The CCP has the flexibility of adjusting the conservatism of the generated flight paths using the value of ϵ. If the user is risk-averse (very sensitive to the uncertainty), a very small value of ϵ could be specified to develop a flight path. If the user is risk-prone (less sensitive to the uncertainty), a larger value of ϵ could be used instead. Hence, the CCP method provides a flexible way to consider uncertain factors in developing alternative flight paths as well as the decision maker's risk tolerance with regard to those uncertainties.

An extension of this work may include a method to determine an appropriate threshold value at each waypoint. This value may affect the timing of the decision to reroute the current flight path and, thus, the amount of unmet demand. Furthermore, there is a need to collect more comprehensive data to understand better how uncertain factors affect drone flight performance in real-time under various flight environments, such as extreme weather conditions, and different drone specifications.

References

1. S. J. Kim, G. J. Lim, J. Cho, M. J. Côté, Drone-aided healthcare services for patients with chronic diseases in rural areas, Journal of Intelligent & Robotic Systems 88 (1) (2017) 163–180. doi:https://doi.org/10.1007/s10846-017-0548-z.
2. S. J. Kim, G. J. Lim, Drone-aided border surveillance with an electrification line battery charging system, Journal of Intelligent & Robotic Systems 92 (3-4) (2018) 657–670. doi:10.1007/s10846-017-0767-3. URL `https://doi.org/10.1007/s10846-017-0767-3`
3. G. J. Lim, S. Kim, J. Cho, Y. Gong, A. Khodaei, Multi-UAV pre-positioning and routing for power network damage assessment, IEEE Transactions on Smart Grid 9 (4) (2018) 3643–3651. doi:10.1109/TSG.2016.2637408.

4. S. J. Kim, G. J. Lim, A hybrid battery charging approach for drone-aided border surveillance scheduling, Drones 2 (4) (2018) 38.
5. C. C. Murray, A. G. Chu, The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, Transportation Research Part C: Emerging Technologies 54 (2015) 86–109.
6. J. J. Boutilier, S. C. Brooks, A. Janmohamed, A. Byers, J. E. Buick, C. Zhan, A. P. Schoellig, S. Cheskes, L. J. Morrison, T. C. Chan, Optimizing a drone network to deliver automated external defibrillators, Circulation (2017) CIRCULATIONAHA–116.
7. S. R. R. Singireddy, T. U. Daim, Technology roadmap: Drone Delivery–Amazon Prime Air, in: Infrastructure and Technology Management, Springer, 2018, pp. 387–412.
8. M. Torabbeigi, G. J. Lim, S. J. Kim, Drone delivery scheduling optimization considering payload-induced battery consumption rates, Journal of Intelligent & Robotic Systems (2019) 1–17doi:https://doi.org/10.1007/s10846-019-01034-w.
9. I. Colomina, P. Molina, Unmanned aerial systems for photogrammetry and remote sensing: A review, ISPRS Journal of Photogrammetry and Remote Sensing 92 (2014) 79–97.
10. S. Amici, M. Turci, F. Giulietti, S. Giammanco, M. Buongiorno, A. La Spina, L. Spampinato, Volcanic environments monitoring by drones mud volcano case study, Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci (2013) 5–10.
11. H. Cruz, M. Eckert, J. Meneses, J.-F. Martínez, Efficient forest fire detection index for application in Unmanned Aerial Systems (UASs), Sensors 16 (6) (2016) 893.
12. A. Dabros, M. Pyper, G. Castilla, Seismic lines in the boreal and arctic ecosystems of North America: environmental impacts, challenges, and opportunities, Environmental Reviews 26 (2) (2018) 214–229.
13. N. Ceccarelli, J. J. Enright, E. Frazzoli, S. J. Rasmussen, C. J. Schumacher, Micro UAV path planning for reconnaissance in wind, in: American Control Conference, 2007. ACC'07, IEEE, 2007, pp. 5310–5315.
14. DJI, DJI GS pro, `https://www.dji.com/ (Last accessed on July 11, 2018)` (2018).
15. J. McHale, Ground control stations for Unmanned Aerial Vehicles (UAVs) are becoming networking-hub cockpits on the ground for U.S. unmanned forces, `https://www.militaryaerospace.com/ (Last accessed on July 11, 2018)` (2010).
16. C. Howard, UAV command, control & communications, Military & Aerospace Electronics, militaryaerospace. com.
17. S. J. Kim, G. J. Lim, J. Cho, Drone flight scheduling under uncertainty on battery duration and air temperature, Computers & Industrial Engineering 117 (2018) 291–302. doi:https://doi.org/10.1016/j.cie.2018.02.005.
18. G. S. Avellar, G. A. Pereira, L. C. Pimenta, P. Iscold, Multi-UAV routing for area coverage and remote sensing with minimum time, Sensors 15 (11) (2015) 27783–27803.
19. A. Kundu, T. I. Matis, A delivery time reduction heuristic using drones under windy conditions, in: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2017, pp. 1864–1869.
20. A. Charnes, W. W. Cooper, Chance-constrained programming, Management science 6 (1) (1959) 73–79.
21. A. Nemirovski, A. Shapiro, Convex approximations of chance constrained programs, SIAM Journal on Optimization 17 (4) (2006) 969–996.
22. M. Zaghian, G. J. Lim, A. Khabazian, A chance-constrained programming framework to handle uncertainties in radiation therapy treatment planning, European Journal of Operational Research 266 (2) (2018) 736–745.
23. Y. Lin, S. Saripalli, Sampling-based path planning for uav collision avoidance, IEEE Transactions on Intelligent Transportation Systems 18 (11) (2017) 3179–3192.
24. P. Yao, H. Wang, Z. Su, Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment, Aerospace Science and Technology 47 (2015) 269–279.
25. G. Gordon, R. Tibshirani, Karush-kuhn-tucker conditions, Optimization 10 (725/36) (2012) 725.
26. T.-H. Wu, C. Low, J.-W. Bai, Heuristic solutions to multi-depot location-routing problems, Computers & Operations Research 29 (10) (2002) 1393–1415.
27. R. Kulkarni, P. R. Bhave, Integer programming formulations of vehicle routing problems, European Journal of Operational Research 20 (1) (1985) 58–67.
28. DJI, Phantom 4 Pro, `https://www.dji.com/phantom-4-pro` (Last accessed on December 14, 2016) (2017).
29. W. Yang, H. Xu, Distributionally robust chance constraints for non-linear uncertainties, Mathematical Programming 155 (1-2) (2016) 231–265.
30. G. C. Calafiore, L. El Ghaoui, On distributionally robust chance-constrained linear programs, Journal of Optimization Theory and Applications 130 (1) (2006) 1–22.
31. L. Friedman, D. Golenko-Ginzburg, Z. Sinuany-Stern, Determining control points of a production system under a chance constraint, Engineering costs and production economics 18 (2) (1989) 139–144.

32. GAMS Development, Corporation, General Algebraic Modeling System (GAMS) Release 24.9.2, Washington, DC, USA, http://www.gams.com/ (2017).
33. IBM, CPLEX Optimizer, http://www.ibm.com/ (2015).