Reliable Path Planning for Drone Delivery Using a Stochastic Time-Dependent Public Transportation Network

Hailong Huang, Andrey V. Savkin, and Chao Huang

***Abstract*—Drones have been regarded as a promising means for future delivery industry by many logistics companies. Several drone-based delivery systems have been proposed but they generally have a drawback in delivering customers locating far from warehouses. This paper proposes an alternative system based on a public transportation network. This system has the merit of enlarging the delivery range. As the public transportation network is actually a stochastic time-dependent network, we focus on the reliable drone path planning problem (RDPP). We present a stochastic model to characterize the path traversal time and develop a label setting algorithm to construct the reliable drone path. Furthermore, we consider the limited battery lifetime of the drone to determine whether a path is feasible, and we account this as a constraint in the optimization model. To accommodate the feasibility, the developed label setting algorithm is extended by adding a simple operation. The complexity of the developed algorithm is analyzed and how it works is demonstrated via a case study.**

***Index Terms*—Parcel delivery, drones, unmanned aerial vehicles (UAVs), path planning, public transportation network, stochastic time-dependent network.**

I. Introduction

DRONES, also known as Unmanned Aerial Vehicles (UAVs), have advanced significantly and been applied widely in civilian domains such as wireless communication service [1], surveillance [2], parcel delivery [3], etc. In recent years, many logistics companies have started to test their drone delivery systems [4]–[7]. In the research community, two schemes relating to drone delivery have been proposed. The first one focuses on only using drones for delivery: a warehouse is equipped with a fleet of drones, and each drone generally follows the routine of recharging, flying to a customer, dropping off the parcel, and returning to the warehouse. The problem considered for this scheme is the scheduling problem [8]–[10]. The second one is called the drone-truck collaboration scheme [3], [11]–[13]. Departing from the warehouse, a truck carries a set of parcels and one or several drones. The drones can launch from the truck, deliver customers, and then dock with the truck at some position after completing the deliveries. In the meanwhile, the truck can deliver other customers, and then move to the scheduled position to rendezvous with the drones.

These two strategies both have merits and demerits. The first method can deliver a customer quickly, and as there is little human labour involved, the system running cost is low. However, the drones suffer from the battery constraint, which results in a limited delivery range. The second mode guarantees the same delivery area as the conventional ground delivery by trucks. However, it leads to a high running cost, since a driver is still needed. Besides, the maintaining cost and the fuel cost on trucks are relatively higher than those of drones. To avoid involving trucks in parcel delivery, researches have proposed two approaches to increase the delivery area of drones. The first one is to locate warehouses or install charging stations [14]–[16] where a drone can replace or recharge the battery if necessary. The second approach is to explore autonomous mobility such as public transportation vehicles [17], [18]. A drone can land on the roof of a public transportation vehicle, travel with it, and then leave the vehicle at some position near the customer. If necessary, the drone can also transfer to another public transportation vehicle. It is easy to see that the first approach requires the supplier to invest on constructing infrastructures, while the second approach is promising to enable a drone to reach a customer that locates far from the warehouse without much extra cost since the public transportation vehicles are natural mobile platforms traversing our living area.

Since the second approach is more cost-efficient, which may be more favorable by suppliers, the current paper focuses on this approach, and we investigate a fundamental problem of how to make use of the public transportation vehicles to transport a drone from the warehouse to a customer. Different from the normal delivery vehicles such as trucks, the public transportation vehicles are not controlled by logistics companies. They have fixed routes and predefined timetables. Besides, these vehicles cannot precisely follow the timetables. The travel time of a vehicle between two stops is impacted by the departure time (in rush hours or off-peak hours) as well as some uncertainty like traffic congestion, passenger demand,

Manuscript received July 16, 2019; revised November 17, 2019 and January 26, 2020; accepted March 23, 2020. Date of publication April 7, 2020; date of current version August 9, 2021. This work was supported by Australian Research Council. The Associate Editor for this article was M. Menendez.
(Corresponding author: Hailong Huang.)

Hailong Huang and Andrey V. Savkin are with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: hailong.huang@unsw.edu.au; a.savkin@unsw.edu.au).

Chao Huang is with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: ch449@uowmail.edu.au).

Digital Object Identifier 10.1109/TITS.2020.2983491

vehicle breakdowns, etc. Therefore, the public transportation network is actually a stochastic time-dependent network. As a delivery drone may also have limited ability of computation, it is sensible to construct a path before the drone leaves the warehouse. Then, the drone can only memorize some key information about the path and does not need to re-plan the path during the trip. Therefore, a supplier may wish to construct a reliable path prior to departure, and we name this problem the reliable drone path planning problem (RDPP).

Without the term of drone, the reliable path planning problem in a stochastic time-dependent network is extended from the path planning problem in deterministic non-time-dependent networks [19], [20], stochastic non-time-dependent networks [21], and deterministic time-dependent networks [22], [23]. With uncertainty and time-dependency accounted together, the problem becomes more complex [24], [25]. Some publications have proposed methods for finding the path with the least expected time [26]–[28]. The path obtained by these approaches can have the least expected travel time, but it is with some risk or, in other words, it is not reliable. A common way to incorporate the travel time variability is to further account some other metrics such as standard deviation. Then, the reliable path can be redefined as the one that maximizes the probability of guaranteeing a travel time no greater than a given threshold [29]–[32].

Returning to reliable drone path planning in a stochastic time-dependent network, we need to model the new problem, which is different from the existing problems in several aspects. The considered stochastic time-dependent network is actually a combination of the public transportation network and the drone flight network. We can also regard it as a multimodal network. The node set of this network consists of the vehicle stops and two extra nodes: the warehouse and the customer. The edge set consists of the links in the public transportation network and some extra links. The extra links connect the two extra nodes with some stop nodes and may also connect two stop nodes if they are not linked yet. The number of extra links should be sufficient such that the two extra nodes are connected in the combined network. Clearly, this network inherits the stochastic time-dependent feature of the public transportation network. In this network, the drone should fly by itself and travel with some vehicles, and these two operations should appear alternately on a drone path. Thus, the drone many need to wait for the vehicle if it arrives earlier. This is different from many existing publications focusing on road networks where waiting at an intermediate node is not permitted [26]–[28], [30].

Following the network flow framework [33], the considered RDPP problem is formulated as an optimization problem with the objective of maximizing the probability of delivering the customer on time. The so-called delivering on time requires the drone to arrive at the customer within a given time. We characterize the path traversal time for different cases to extend a path. Specifically, we investigate the probability density functions of the drone path when it is extended by adding a drone flight and travelling with a vehicle. In the latter case, the drone has to arrive at the stop node earlier than the vehicle with which the drone plans to travel, and this involves a waiting time. Besides, with which vehicle a drone can travel depends on the arrival instant of the drone and the departure instant of the vehicle. (Regarding this point, the current paper differs from [31], which considers the reliable path planning in a bus network but assumes that the waiting time only depends on the travel time of the predecessor link.) Attention is paid to the computation of the probability density function of the drone path in these cases. Following the concept of reliability in stochastic networks [34], the original probability maximization problem is converted to an equivalent one that minimizes the path traversal time under a given confidence level. Following the concept of stochastic dominance, we develop a label setting algorithm to find non-dominant paths from the warehouse to the customer. Moreover, we take into account the battery lifetime of the drone. The limited battery lifetime determines whether a path is feasible or not. Then, we have an extra constraint in both the original problem and the converted problem. The developed label setting algorithm is extended by taking one more operation to verify whether a path is feasible. The complexity of the proposed algorithm is analyzed and how it works is demonstrated by an extensive case study.

TABLE I
MAIN NOTATIONS AND MEANINGS

Scheme	Delivery time
a_{ij}^l	The link from node i to node j by line l
r_{ij}^l	The route from node i to node j by line l
$\mathcal{R}$	The set of all the routes
p_{ij}	The path from node i to node j
p_i	The path from W to node i
b_{ij}^l	The time cost of the route r_{ij}^l
c_i^l	The time cost at node i
$\Gamma(i)$ $(\Gamma^{-1}(i))$	The set of successors (predecessors) of node i
τ_C	The traversal time of path p_C
$\phi_{b_{ij}^l}$ (ϕ_{τ_i})	The PDF of b_{ij}^l (τ_i)
$\Phi_{\tau_C}(t)$	The CDF of τ_C

The main contributions of this paper are as follows:

- We propose a new mode for drone parcel delivery which explores the public transportation network. This mode is promising to reduce the system running cost and enlarge the delivery area significantly.
- We formulate the reliable drone path planning problem (RDPP) following the network flow framework to maximize the on time delivery probability. To address RDPP, we convert it equivalently to another problem that minimizes the path traversal time under a given confidence level.
- We propose a label setting algorithm to find the optimal solution.
- Moreover, we extend the model to account the battery lifetime of the drone. The proposed algorithm is also extended by taking an extra operation to verify the feasibility of a path.

The rest of the paper is organized as follows. Section II formally states the studied problem and Section III presents the proposed algorithm. A case study is demonstrated in Section IV and a conclusion is given in Section V.

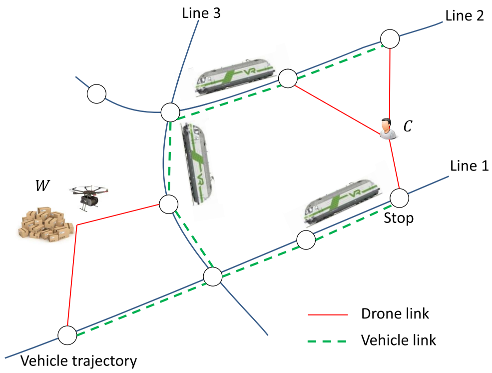


Fig. 1. An illustration of the public transportation network and the drone flight network.

II. System Model and Problem Statement

Let $\mathcal{G}(\mathcal{N}_1, \mathcal{A}_1, \mathcal{L}_1)$ denote the public transportation network, where $\mathcal{N}_1$ is a set of nodes consisting of the vehicle stops, $\mathcal{A}_1$ is a set of links between pairs of vehicle stops: $\mathcal{A}_1 = \{a^l_{ij} | i, j \in \mathcal{N}_1, l \in \mathcal{L}_1\}$, and $\mathcal{L}_1 = \{l\}$ is the set of vehicle line labels; see Fig. 1. Let $\mathcal{G}(\mathcal{N}_2, \mathcal{A}_2, \mathcal{L}_2)$ denote a drone flight network. $\mathcal{N}_2 = \mathcal{N}_1 \cup W \cup C$, where W represents the warehouse and C represents the customer. We assume that the warehouse W and the customer C do not locate at any vehicle stops. This is a reasonable assumption because in practice the warehouse and customers cannot be at any stop, although they may be close to some stops. $\mathcal{A}_2$ consists of some links connecting two nodes in $\mathcal{N}_2$: $\mathcal{A}_2 = \{a^0_{ij} | i, j \in \mathcal{N}_2\}$, where the label 0 is used to represent drone flight, i.e., $\mathcal{L}_2 = \{0\}$. They connect the warehouse W and the customer C with some stop nodes and two stop nodes that are within a certain range but not linked in $\mathcal{A}_1$; see Fig. 1. The second group enables transfers between stops, i.e., the drone can fly to another stop to catch a vehicle. Let $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$ be a directed network or graph that combines $\mathcal{G}(\mathcal{N}_1, \mathcal{A}_1, \mathcal{L}_1)$ and $\mathcal{G}(\mathcal{N}_2, \mathcal{A}_2, \mathcal{L}_2)$. Specifically, $\mathcal{N} = \mathcal{N}_2$, $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, and $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$. The set $\mathcal{A}_2$ can be added with more drone flights to make the warehouse W and the customer C connected in $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$. Note that if the warehouse W and the customer C are close to some vehicle lines, some virtual stops (which are the positions on some vehicle lines and are closest to the warehouse W or the customer C) can be added into $\mathcal{N}_1$ at which a vehicle can catch or leave a vehicle without flying too far. In this case, some related links in $\mathcal{A}_1$ are divided into two links. This will increase the number of nodes and links in $\mathcal{G}$ slightly, but it does not impact on the presented model.

The links in $\mathcal{A}$ are the basis of the system model. Additionally, two other terms also play a pivotal role, i.e., route and path. A route is a series of links that satisfy two conditions: 1) their labels are identical, and 2) one link and its successor in the series share the same node. The term route is used to describe the travel method between two stops using the same line. Let r^l_{ij} denote a route from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$ via line $l \in \mathcal{L}$. According to the definition of the route, we can represent a route by $r^l_{ij} = a^l_{ik_1} \oplus a^l_{k_1k_2} \oplus \cdots \oplus a^l_{k_{n-1}k_n} \oplus a^l_{k_nj}$. Here, $\oplus$ is the link (route and path) concatenation operator. Notice that a route can consist of only one link. Let $\mathcal{R}$ denote the set of all the routes in $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$. The term path refers to a series of routes belonging to $\mathcal{R}$. These routes have different labels, and one route needs to share the same node with its successor in the series. Let p_{ij} denote a path from node $i \in \mathcal{N}$ to node $j \in \mathcal{N}$. If there are multiple paths between i and j, we use a superscript for distinction, e.g., p^u_{ij}. In the rest of this paper, when we mention a path from the warehouse W to some node j, we omit the starting node. In other words, we use the notation p_j when we refer to a path from W to j.

Now, we model the traversal time of a path. To do so, we start from the traversal time of a route. Let b^l_{ij} denote the traversal time over the route r^l_{ij}. If $l \in \mathcal{L}_1$, traversing this route means that the drone travels with a public transportation vehicle labelled by l. Although there is a predefined travel time for each route, b^l_{ij} is a random variable considering various traffic factors like the congestion. If $l = 0$, it means that the drone flies by itself. Although the drone flight is independent on the traffic conditions, b^l_{ij} may still be a random variable if environmental factors such as wind are taken into account. Let $\phi_{b^l_{ij}}(t)$ denote the probability density function (PDF) of b^l_{ij}. For $l \in \mathcal{L}_1$, the PDF can be estimated from empirical observations [35], and for $l = 0$, the data can be collected by conducting filed experiments in different environmental conditions.

Consider two cases where a path has two routes. One case is that the first route is with label 0 and the second route is with a label $l \in \mathcal{L}_1$. The other is that two routes are with the labels $l_1 \in \mathcal{L}_1$ and $l_2 \in \mathcal{L}_1$, respectively, but $l_1 \neq l_2$. The first case means that the drone flies to a stop and then travels with a vehicle. If the drone arrives at the stop earlier than the vehicle, the drone has to wait until the vehicle comes. The second case means that the drone needs to transfer from one route to another. To do so, the drone also needs to hover at the intermediate stop for a while. Both lead to some waiting time at the stop. Let c^l_i denote the waiting time at node i before traversing a route that leaves node i and is labelled by l.

Introduce a binary route-path incidence variable x^C_{ijl}, where $x^C_{ijl} = 1$ if the route r^l_{ij} is on the path p_C, and $x^C_{ijl} = 0$ otherwise. Let τ_C be the traversal time of the path p_C, which can be computed as follows:

$$\tau_C = \sum_{r^l_{ij} \in \mathcal{R}} x^C_{ijl}(b^l_{ij} + c^l_i). \tag{1}$$

Since b^l_{ij} and c^l_i are random variables, τ_C is also a random variable. Its cumulative distribution function (CDF) is:

$$\Phi_{\tau_C}(t) = \Pr\Big(\sum_{r^l_{ij} \in \mathcal{R}} x^C_{ijl}(b^l_{ij} + c^l_i) \leq t\Big). \tag{2}$$

Let T denote the customer specified delivery time and t_0 be the instant at which the customer makes the order. If the drone can reach the customer at or before the instant $t_0 + T$, we say the delivery is on time. The probability of fulfilling this requirement is given by $\Phi_{\tau_C}(T)$. We introduce another notation $\Gamma(i)$ to denote the set of successors of node i in the network $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$. $\Gamma^{-1}(i)$ denotes the set of predecessors of node i.

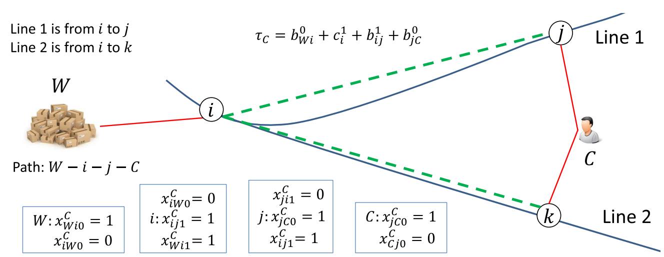


Fig. 2. A small example to illustrate the path from the warehouse W to the customer C via nodes i and j.

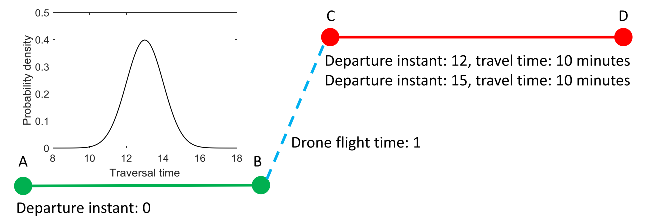


Fig. 3. An example to illustrate the dependence of travel times and waiting time. The drone departs A at instant 0 and travels with a bus. When it arrives B, the drone flies to C. From C, it takes a train to D.

Definition 1: Let P_C denote the set of paths from W to C. We say a path $p^u_C \in P_C$ is T-reliable if the following condition is satisfied [34]:

$$\Phi_{\tau^u_C}(T) \geq \Phi_{\tau^v_C}(T), \quad \forall p^v_C \in P_C,\ v \neq u. \tag{3}$$

Problem Statement: For the given $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$, T, and t_0, the RDPP problem can be stated as to find a path from the warehouse W to the customer C such that the probability of delivery on time is maximized. This problem can be formulated following the conventional network flow framework [33]:

$$\max_{x^C_{ijl}} \ \Phi_{\tau_C}(T), \tag{4}$$

subject to

$$\sum_{j \in \Gamma(W)} x^C_{Wjl} - \sum_{k \in \Gamma^{-1}(W)} x^C_{kW\hat{l}} = 1, \tag{5}$$

$$\sum_{j \in \Gamma(C)} x^C_{Cjl} - \sum_{k \in \Gamma^{-1}(C)} x^C_{kC\hat{l}} = -1 \tag{6}$$

$$\sum_{j \in \Gamma(i)} x^C_{ijl} - \sum_{k \in \Gamma^{-1}(i)} x^C_{ki\hat{l}} = 0, \quad \forall i \neq W,\ i \neq C, \tag{7}$$

$$x^C_{ijl} \in \{0, 1\}, \quad r^l_{ij} \in \mathcal{R},\ l, \hat{l} \in \mathcal{L}, \tag{8}$$

The constraints (5), (6) and (7) require the balance of income and outcome of a node. The node W has only one outcome and the node C has only one income. All the other nodes must have either one income and one outcome, or no income and no outcome.

To illustrate the considered problem, we present a small example with two lines and three stops as shown in Fig. 2. From the warehouse W, the drones may have two paths to reach the customer C. One is $W - i - j - C$ and the other is $W - i - k - C$. If the drone follows the first path, i.e., the routes r^0_{Wi}, r^1_{ij} and r^0_{jC} are selected, the constraints shown in the boxes of Fig. 2 should hold for nodes W, i, j and C. Then, the path traversal time is given by $\tau_C = b^0_{Wi} + c^1_i + b^1_{ij} + b^0_{jC}$. If the drone follows the second path, the corresponding traversal time is given by $\tau_C = b^0_{Wi} + c^2_i + b^2_{ik} + b^0_{kC}$. The better path can be further determined by evaluating the CDF of the path traversal time, i.e., (4).

III. Proposed Solution

Although the problem (4) follows the conventional network flow framework [33], the existing algorithms do not apply to this problem. The reason is that the route traversal times and waiting times are dependent. We show this feature via a simple example in Fig. 3. The drone needs to reach node D from node A via node B and C. Route r_{AB} (the green one) represents a bus route, and its traversal time distribution is shown above the route. Route r_{BC} (the blue one) represents a flight route, and we assume that the traversal time is fixed by neglecting the environmental factors such as wind. Route r_{CD} (the red one) represents a train route which is very reliable. There are two trains running from C to D, and each has the fixed departure instant and traversal time. The drone departs node A at instant 0 and the arrival instant at node B follows the given distribution. Suppose the drone arrives at node B before instant 11. After flying for 1 minute, it arrives at node C before instant 12, which enables it to reach node D at instant 22. However, if the drone reaches node B after instant 11 but before instant 14, it can reach node C before the next departure of a train. In this case, the drone needs to hover at node C for sometime. It can then arrive at node D at instant 25. It is clear that the waiting time depends on the route traversal time and the vehicle departure instant.

To address the considered problem, we first model the probability density of a path consisting of several routes in Section III-A, and then we discuss how to construct the reliable path in Section III-B.

A. Path Traversal Time

Suppose we have a path p_i from node W to node i. Consider that we extend it by adding a route such that node j is reached. There are three different extensions.

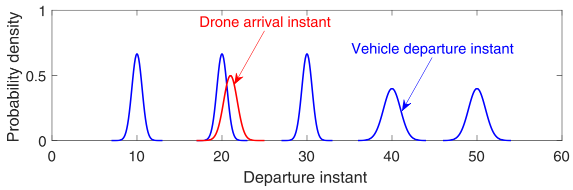
1) Vehicle Route Plus Drone Flight Route: If the end route of p_i is a vehicle route, we can add a drone flight route a^0_{ij}. The traversal time of path p_i is represented by τ_i, and its PDF is ϕ_{τ_i}. The traversal time of the path p_j is computed by:

$$\tau_j = \tau_i + b^0_{ij}. \tag{9}$$

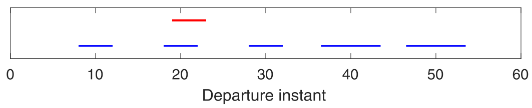
Here, the waiting time at node i is $c^0_i = 0$ since we always require that if the drone needs to leave a vehicle, it does so immediately. As the drone flight has no relationship with how long the drone travels with a vehicle, the probability densities τ_i and b^0_{ij} are independent. Then, the PDF of τ_j is given as follows:

$$\phi_{\tau_j} = \phi_{\tau_i} * \phi_{b^0_{ij}}, \tag{10}$$

where $*$ is the convolution operator.



(a)



(b)

Fig. 4. (a) There are five vehicles scheduled to depart at instant 10, 20, 30, 40, and 50, respectively. (b) The HPAR of the drone arrival instant is between the two HPARs of the vehicle departure instants, where $\alpha = 0.005$ and $\beta = 0.995$.

2) Vehicle Route Plus Vehicle Route: We can also extend the path p_i ended with a vehicle route by adding another vehicle route a_{ij}^l, $l \in \mathcal{L}_1$. Then, at node i, the drone may need to hover to wait for the vehicle and the corresponding waiting time is c_i^l. Thus, the traversal time of the path p_j is given by:

$$\tau_j = \tau_i + c_i^l + b_{ij}^l. \tag{11}$$

To characterize the waiting time c_i^l, we introduce another random variable d_i^l, which is the departure instant of a vehicle from node i labelled by l. The PDF d_i^l is denoted by $\phi_{d_i^l}$. Similar to that of the route traversal time, the PDF $\phi_{d_i^l}$ can also be extracted from empirical observations. It is easy to understand that the waiting time is the gap between the departure instant of the vehicle with which the drone plans to travel and the instant at which the drone arrives at node i, i.e.,

$$c_i^l = d_i^l - (\tau_i + t_0). \tag{12}$$

Here, $\tau_i + t_0$ is the arrival instant at node i. Clearly, the waiting time at node i depends on two random variables: the drone arrival instant at node i, i.e., $t_0 + \tau_i$, and the vehicle departure instant from node i, i.e., d_i^l. Summing up the drone arrival instant and the waiting time at node i, we obtain the drone departure instant from node i. To further add a vehicle route to the path, it is necessary to know the PDF of the drone departure instant. Since the drone departure instant is just the vehicle departure instant, i.e., d_i^l, the PDF of the drone departure instant is just that of d_i^l. As there may exist multiple vehicles labelled by l to depart from node i, the problem turns to the determination of which vehicle the drone can catch, see Fig. 4a. Let H_l denote the number of vehicles serving the line l. The departure instant of the hth (where $h = 1, 2, \ldots, H_l$) vehicle from node i is denoted by d_i^{lh}. Since τ_i and d_i^{lh} are random variables, the computation of the probability of which vehicle the drone can take is quite complex. We propose the following strategy to simplify the computation.

3) High Probability Arrival Range (HPAR): Having path p_i, the CDF of the path traversal time τ_i can be computed. Then, we can obtain the CDF of the drone arrival instant at node i (the distribution is simply shifted by a constant t_0). Introduce a probability interval $[\alpha, \beta]$, where $\alpha, \beta \in (0, 1)$ and $\alpha < \beta$. With the CDF of the drone arrival instant, we can determine the arrival instants corresponding to the probability α and β, respectively, and these two arrival instants form the High Probability Arrival Range (HPAR). In addition, for each vehicle serving line l, we do the same for the departure instant. Then, from the HPARs for vehicle departure instants, from the left side, we can find the HPAR whose right-end instant is the closest to the left-end instant of the HPAR of the drone. We call it the left-range. For example, in Fig. 4b, the HPAR of the vehicle scheduled to depart at instant 10 is the left-range. Similarly, from the right side, we can find the HPAR whose left-end instant is the closest to the right-end instant of the drone HPAR. We call it the right-range. In Fig. 4b, the HPAR of the vehicle scheduled to depart at instant 30 is the right-range. It is possible that there exist some other HPARs between the left-range and the right-range. In this paper, we take consider the worst case, i.e, the drone will take the vehicle corresponding to the right-range. The drone can travel with this vehicle with the probability $\beta - \alpha$. In the example shown in Fig. 4b, the drone will take the vehicle scheduled to depart at instant 30. Once we know the vehicle with which the drone can travel, the PDF of the drone departure instant is obtained. We use $\phi_{d_i^l|\tau_i+t_0}$ to represent such a PDF. Clearly, it is based on the knowledge of the HPAR of the random variable $\tau_i + t_0$. Furthermore, we use $\phi_{b_{ij}^l|\tau_i+t_0}$ to represent the PDF of the traversal time b_{ij}^l.

With Eq. (12), Eq. (11) becomes:

$$\tau_j = d_i^l + b_{ij}^l - t_0. \tag{13}$$

We assume that the route traversal time and vehicle departure instant are independent. This is reasonable because for a given scheduled departure instant, the route traversal time depends on the traffic condition mostly. For example, for a vehicle scheduled to depart at 12:00, the actual departure time, say 11:58 or 12:03, has little impact on the travel time to some stop. However, the travel time differs with different scheduled departure instants. For example, the travel time when departing at around 12:00 may be quite different from the one that departs at around 17:00 (which is during the rush hours). Based on this assumption, the PDF of τ_j is computed by:

$$\phi_{\tau_j} = \phi_{d_i^l|\tau_i+t_0} * \phi_{b_{ij}^l|\tau_i+t_0}. \tag{14}$$

4) Drone Flight Route Plus Vehicle Route: If the path p_i is ended with a drone flight route, it does not make sense to add another drone flight route, while it is sensible to add a vehicle route. Then, the situation is the same as the second case discussed above, and the path traversal time and the probability density are given by (13) and (14), respectively.

Remark 2: In the above cases, when the drone travels with a vehicle, it can turn off the motors to save energy. However, the controller should keep working. The energy consumed by the controller can be ignored compared to that consumed by the motors. Via GPS, the drone knows its current position. By comparing the GPS position and the pre-computed way-

points of the path, the drone knows when and where to leave the vehicle.

B. Reliable Path Construction

Let $\Phi_{\tau_i}^{-1}(\lambda)$ denote the inverse of the CDF of the traversal time of path p_i:

$$\Phi_{\tau_i}^{-1}(\lambda) = \{\xi | \Pr(\tau_i \leq \xi) \geq \lambda\}. \tag{15}$$

In words, for a confidence level $\lambda \in (0, 1)$, $\Phi_{\tau_i}^{-1}(\lambda)$ gives the path traversal time that can be achieved with the confidence level λ.

Definition 3: Let P_C denote the set of paths from W to C. We say a path $p_C^u \in P_C$ is λ-reliable if the following condition is satisfied [34]:

$$\Phi_{\tau_C^u}^{-1}(\lambda) \leq \Phi_{\tau_C^v}^{-1}(\lambda), \quad \forall p_C^v \in P_C, \ v \neq u. \tag{16}$$

With Definition 3, the original RDPP problem (4), can be reformulated as follows:

$$\min_{x_{ijl}^C} \ \Phi_{\tau_C}^{-1}(\lambda), \tag{17}$$

subject to (5), (6), (7) and (8).

Here, the objective becomes to find the λ-reliable path having the shortest traversal time.

Remark 4: Definition 3 and Definition 1 are equivalent. Thus, the problem (17) is also equivalent with the original problem (4).

To address the problem (17), we introduce the following assumption and definition.

Assumption 5: Following [24], we assume the network $\mathcal{G}(\mathcal{N}, \mathcal{A}, \mathcal{L})$ is stochastic consistent, i.e., the probability of arriving at any node by any given instant cannot be increased by departing later.

Therefore, we always require the drone to take the first departing vehicle it can catch.

Definition 6: Let P_i denote the set of paths from W to i. Consider two paths $p_i^u \in P_i$ and $p_i^v \in P_i$, and $p_i^u \neq p_i^v$. We say p_i^u stochastically dominates p_i^v if the following condition is satisfied:

$$\Phi_{\tau_i^u}^{-1}(\lambda) < \Phi_{\tau_i^v}^{-1}(\lambda), \quad \forall \lambda \in (0, 1). \tag{18}$$

This is to say that p_i^u takes shorter time than p_i^v for any confidence level $\lambda \in (0, 1)$. Besides, we say p_i^u is a non-dominant path in P_i if it is not dominated by any other path $p_i^v \in P_i$. With Assumption 5 and Definition 6, we conclude that if p_i^u dominates p_i^v and suppose $j \in \Gamma(i)$, the extended path $p_i^u \oplus r_{ij}^l$ dominates $p_i^v \oplus r_{ij}^l$. Therefore, whenever we encounter a dominated path in the path extension process, it can be discarded safely. This plays a crucial role in our method, because it allows us to prune some candidate paths without evaluating them.

Now, it is the position to present the reliable path construction algorithm. The basic idea is similar to many well-known path planning algorithms for a static graph, i.e., extending the path from the Warehouse W by adding a new route. Obviously, when uncertainty and time-dependency are accounted, some extra operations are needed to evaluate a candidate path.

Algorithm 1 Constructing the Reliable Path

```
1: Construct an initial path p_W.
2: SE ← p_W.
3: while SE has a path to a node i and i ≠ C do
4:     Select p_i^u (i ≠ C) in SE and remove it from SE.
5:     for j ∈ Γ(i) and any feasible l do
6:         if j ∉ p_i^u then
7:             p_j^u ← p_i^u ⊕ r_ij^l.
8:             Compute φ_{τ_j^u}.
9:             if p_j^u is not dominated by any path in P_j then
10:                P_j ← P_j ∪ p_j^u, SE ← SE ∪ p_j^u.
11:                Remove paths dominated by p_j^u from P_j.
12:                Remove paths dominated by p_j^u from SE.
13:            end if
14:        end if
15:    end for
16: end while
```

Let SE be the scan eligible set that stores the non-dominant paths and P_i store the non-dominant paths from W to i. Initially, SE contains only the path P_W, whose path traversal time probability density is 1. In each iteration, we select one path, say p_i^u, where $i \neq C$, and remove it from SE. Then, for each successor of node i, say $j \in \Gamma(i)$, and each feasible line l leaving node i, we construct an extended path $p_j^u = p_i^u \oplus r_{ij}^l$. To evaluate this extended path, we need to look at the route types of the ending route of the path p_i^u and the added route r_{ij}^l, based on which we compute the PDF of the extended path traversal time by either (10) or (14). If (14) is used, we further need to estimate which vehicle the drone can take by the HPAR strategy. After knowing the PDF of the traversal time of the extended path p_j^u, we conduct the following two operations to modify the set P_j and SE. Firstly, if p_j^u is not dominated by any other paths in P_j, which is checked according to Definition 6, we add it to P_j and SE. Secondly, if any path in P_j and SE is dominated by p_j^u, we remove it from P_j and SE. The algorithm repeats the above process until SE stores paths only to C. These procedures are summarized in Algorithm 1. When the algorithm terminates, the node C is associated with a set of non-dominant paths and the PDF of these paths have already been computed.

We now analyze the complexity of Algorithm 1. Let $|P|$ denote the maximum number of non-dominant paths from W to any other node in $\mathcal{N}$, $|\Gamma|$ denote the maximum number of successors of any node, $|L|$ denote the maximum number of lines between two nodes, and $|H|$ denote the maximum number of vehicles serving a line within a period of interest. Clearly, the maximum number of non-dominant paths in SE is bounded by $O(|\mathcal{N}||P|)$, which is the maximum number of iterations. For any non-dominant path, say p_i^u, the maximum number of extensions is the number of combinations of successors and lines, i.e., $O(|\Gamma||L|)$. For each extended path, say p_j^u, the HPAR strategy requires to compare the HPAR of the extended path with those of at most $O(|H|)$ scheduled departure instants (Line 8 of Algorithm 1), and compare this path with at most $O(|P|)$ paths in the set P_j (Line 9 of

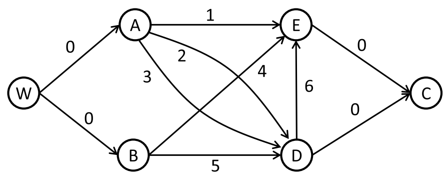


Fig. 5. The network used to demonstrate how the proposed algorithm works.

Algorithm 2 Constructing the Reliable and Feasible Path

```
Construct an initial path p_W.
SE ← p_W.
while SE has a path to a node i and i ≠ C do
    Select p_i^u (i ≠ C) in SE and remove it from SE.
    for j ∈ Γ(i) and any feasible l do
        if j ∉ p_i^u then
            p_j^u ← p_i^u ⊕ r_{ij}^l.
            Compute φ_{τ_j^u} and φ_{e_j^u}.
            if Φ_{e_C}(E_0) ≥ η then
                if p_j^u is not dominated by any other path in P_j then
                    P_j ← P_j ∪ p_j^u, SE ← SE ∪ p_j^u.
                    Remove paths dominated by p_j^u from P_j.
                    Remove paths dominated by p_j^u from SE.
                end if
            end if
        end if
    end for
end while
```

Algorithm 1). Therefore, the overall computational complexity is $O(|\mathcal{N}||\Gamma||L||H||P|^2)$ in the worst case. Clearly, for a given network, $|\Gamma|$, $|L|$, $|H|$ and $|P|$ are fixed, and they should not be large for a practical network. For example, the number of stops connecting directly by some lines to one stop may be a small number if this stop is at a traffic junction, while this number is often two for the normal stops. Besides, for a specific customer, many stops are useless since they may never appear on a feasible path. Removing these nodes from the network significantly reduces the complexity of the proposed approach. Moreover, as we focus on designing the drone flight before its departure, the computational complexity is not a bottleneck here.

C. Energy-Aware Reliable Path

We note that the problem statement in Section II and the proposed reliable path construction algorithm in Section III-B have not considered an important constraint of drones, i.e., the limited battery capacity. In this subsection, we extend the proposed approach such that an energy-aware reliable path can be constructed.

We introduce some more notations to model the energy consumption of the drone. Let E_0 denote the allowed energy to be consumed for the delivery task. Practically, it can be half of the battery capacity. In this paper, we assume that operations of flying and hovering consume approximately the constant

TABLE II

THE DISTRIBUTIONS OF ROUTE TRAVERSAL TIMES

Route	Traversal time
r^0_{WA}	3(0.1)
r^0_{WB}	5(0.2)
r^0_{DC}	6(0.2)
r^0_{EC}	3(0.3)
r^1_{AE}	23(2.0), 23(2.0), 22(2.0), 21(1.5), 21(1.5)
r^2_{AD}	10(1.2), 10(1.2), 10(1.2)
r^3_{AD}	10(1.0), 10(1.0), 10(1.0), 11(1.2), 11(1.3), 11(1.3)
r^4_{BE}	15(1.4), 15(1.4), 15(1.4), 15(1.4), 15(1.4)
r^5_{BD}	8(1.6), 8(1.6), 8(1.6), 10(1.3), 10(1.3), 10(1.3),
r^6_{DE}	6(1.5), 6(1.5), 6(1.5), 6(1.3), 6(1.3), 7(1.3), 7(1.3)

TABLE III

THE DISTRIBUTIONS OF DEPARTURE INSTANTS

	Departure instant
d^1_A	-1(1.0), 9(1.0), 19(1.0), 29(1.0), 39(1.0)
d^2_A	5(1.5), 20(1.5), 35(1.5)
d^3_A	2(1.1), 8(1.1), 14(1.1), 20(1.2), 26(1.3), 32(1.3)
d^4_B	-3(1.4), 7(1.4), 17(1.6), 27(1.6), 37(1.6)
d^5_B	-1(1.3), 6(1.3), 13(1.3), 20(1.3), 30(1.4), 40(1.4)
d^6_D	2(1.2), 8(1.2), 14(1.2), 20(1.2), 26(1.0), 31(1.0), 36(1.0)

powers, respectively; while when the drone travels with a vehicle, the energy consumption is assumed to be zero. Recall Remark 2, the energy consumption by the controller during this period is ignored. Let q_f and q_h be the consuming powers of the drone for flying and hovering, respectively. Then, we can relate the energy consumption of the drone on a path to the corresponding time. Let ϵ^l be the energy consuming power on a route with label l defined as follows:

$$\epsilon^l = \begin{cases} q_f, & \text{if } l = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Moreover, let e_i denote the energy consumption on path p_i. Similar to (1), e_C can be computed as:

$$e_C = \sum_{r^l_{ij} \in \mathcal{R}} x^C_{ijl}(\epsilon^l b^l_{ij} + q_h c^l_i). \tag{20}$$

Obviously, (20) is a weighted form of (1). Thus, e_C is also a random variable. The CDF of the energy consumption no more than E_0 is given by:

$$\Phi_{e_C}(E_0) = \Pr(\sum_{r^l_{ij} \in \mathcal{R}} x^C_{ijl}(\epsilon^l b^l_{ij} + q_h c^l_i) \le E_0). \tag{21}$$

Definition 7: Let $\eta \in (0, 1)$ be the confidence level for the total energy consumption no more than E_0. We say the path p_C is feasible if the following condition holds:

$$\Phi_{e_C}(E_0) \ge \eta. \tag{22}$$

With Definition 7, the original problem is extended to maximizing (4) subject to (5), (6), (7), (8) and (22), which is to find the feasible and shortest path in time. The corresponding converted problem becomes maximizing (17) subject to (5), (6), (7), (8) and (22). To address the latter problem, we present Algorithm 2, which is extended from Algorithm 1 by adding

TABLE IV

THE COMPUTATION PROCESS OF APPLYING ALGORITHM 1 TO THE EXAMPLE NETWORK

Iteration	Considered path	Path extension	Traversal time distribution	Vehicle to take	SE
0					$\{P_W\}$
1	P_W	$P_A^1 = P_W \oplus r_{WA}^0$	3(0.1)		$\{P_A^1\}$
		$P_B^1 = P_W \oplus r_{WB}^0$	5(0.2)		$\{P_A^1, P_B^1\}$
2	P_A^1	$P_E^1 = P_A^1 \oplus r_{AE}^1$	32(2.2)	2nd	$\{P_B^1, P_E^1\}$
		$P_D^1 = P_A^1 \oplus r_{AD}^2$	30(1.9)	2nd	$\{P_B^1, P_E^1, P_D^1\}$
		$P_D^2 = P_A^1 \oplus r_{AD}^3$	18(1.5)	2nd	$\{P_B^1, P_E^1, P_D^2\}$
3	P_B^1	$P_E^2 = P_B^1 \oplus r_{BE}^4$	32(1.9)	3rd	$\{P_E^1, P_D^2, P_E^2\}$
		$P_D^3 = P_B^1 \oplus r_{BD}^5$	21(2.1)	3rd	$\{P_E^1, P_D^2, P_E^2, P_D^3\}$
4	P_E^1	$P_C^1 = P_E^1 \oplus r_{EC}^0$	35(2.3)		$\{P_D^2, P_E^2, P_D^3, P_C^1\}$
5	P_D^2	$P_C^2 = P_D^2 \oplus r_{DC}^0$	24(1.5)		$\{P_E^2, P_D^3, P_C^2\}$
		$P_E^3 = P_D^2 \oplus r_{DE}^6$	32(1.6)	5th	$\{P_E^2, P_D^3, P_C^2, P_E^3\}$
6	P_E^2	$P_C^3 = P_E^2 \oplus r_{EC}^0$	35(2.0)		$\{P_D^3, P_C^2, P_E^3\}$
7	P_D^3	$P_E^4 = P_D^3 \oplus r_{DE}^6$	38(1.6)	6th	$\{P_C^2, P_E^3\}$
		$P_C^3 = P_D^3 \oplus r_{DC}^0$	27(2.7)		$\{P_C^2, P_E^3, P_C^3\}$
8	P_E^3	$P_C^4 = P_E^3 \oplus r_{EC}^0$	35(1.7)		$\{P_C^2, P_C^3\}$

TABLE V

THE COMPUTATION PROCESS OF APPLYING ALGORITHM 2 TO THE EXAMPLE NETWORK

Iteration	Considered path	Path extension	Traversal time distribution	Energy consumption distribution	Vehicle to take	SE
0						$\{P_W\}$
1	P_W	$P_A^1 = P_W \oplus r_{WA}^0$	3(0.1)	3(0.1)		$\{P_A^1\}$
		$P_B^1 = P_W \oplus r_{WB}^0$	5(0.2)	5(0.2)		$\{P_A^1, P_B^1\}$
2	P_A^1	$P_E^1 = P_A^1 \oplus r_{AE}^1$	32(2.2)	9(1.0)	2nd	$\{P_B^1, P_E^1\}$
		$P_D^1 = P_A^1 \oplus r_{AD}^2$	30(1.9)	20(1.5)	2nd	$\{P_B^1, P_E^1\}$
		$P_D^2 = P_A^1 \oplus r_{AD}^3$	18(1.5)	8(1.1)	2nd	$\{P_B^1, P_E^1, P_D^2\}$
3	P_B^1	$P_E^2 = P_B^1 \oplus r_{BE}^4$	32(1.9)	17(1.6)	3rd	$\{P_E^1, P_D^2\}$
		$P_D^3 = P_B^1 \oplus r_{BD}^5$	21(2.1)	13(1.3)	3rd	$\{P_E^1, P_D^2, P_D^3\}$
4	P_E^1	$P_C^1 = P_E^1 \oplus r_{EC}^0$	35(2.3)	12(1.0)		$\{P_D^2, P_D^3, P_C^1\}$
5	P_D^2	$P_C^2 = P_D^2 \oplus r_{DC}^0$	24(1.5)	14(1.1)		$\{P_D^3, P_C^2\}$
		$P_E^3 = P_D^2 \oplus r_{DE}^6$	32(1.6)	16(2.1)	5th	$\{P_D^3, P_C^2\}$
6	P_D^3	$P_E^4 = P_D^3 \oplus r_{DE}^6$	38(1.6)	23(2.7)	6th	$\{P_C^2\}$
		$P_C^3 = P_D^3 \oplus r_{DC}^0$	27(2.7)	19(1.3)		$\{P_C^2\}$

one extra procedure only, i.e., verifying constraint (22). Specifically, before verifying whether a path is dominated or not, we check whether this path is feasible; see Line 9. If not, this path is discarded. Such a procedure not only ensures that the stored paths in the set SE are feasible but also reduces the number of iterations for running the algorithm.

IV. CASE STUDY

A simple network is adopted to illustrate the proposed algorithm. As shown in Fig. 5, this network consists of 6 nodes and 10 routes. The route labels are shown along each route. The route traversal times are shown in TABLE II and the departure instants from 4 stop nodes are shown in TABLE III. As all the route traversal times and departure instants are assumed to follow normal distributions in this case, only their mean and standard deviation are provided. The drone flight routes are more reliable than the vehicle routes, i.e., with smaller deviations. We set $t_0 = 0$, $\alpha = 0.0015$ and $\beta = 0.9985$.

We apply Algorithm 1 on the example network, and how it works is shown in TABLE IV. In Iteration 0, SE only contains P_W. In Iteration 1, P_W is the currently considered path. As $\Gamma(W) = \{A, B\}$, we can extend P_W to P_A^1 and P_B^1. Both extensions add drone flight routes. Since the departure instant is $t_0 = 0$, the path traversal time distributions are just those of the flight routes, shown in TABLE II. After Iteration 1, $SE = \{P_A^1, P_B^1\}$. In Iteration 2, P_A^1 is extracted from SE. This path can be extended in three ways, as there are three vehicle lines moving out of node A. For line 1, the algorithm first adopts the HPAR strategy to figure out which vehicle the drone can take. In this case, it is the second vehicle labelled by 1. Then, the path traversal time distribution of P_E^1 is computed by Eq. (14). Similarly, the path traversal time distributions of P_D^1 and P_D^2 can be obtained, which are 30(1.9) and 18(1.5), respectively. According to Definition 6, P_D^2 dominates P_D^1. Thus, after P_D^2 is added to SE, P_D^1 is removed from SE. The algorithm continues until SE contains paths to C only. Therefore, we find two paths P_C^2 and P_C^3 and they are non-dominant. To further obtain the whole path, a backtracking method is adopted, and the process of path extension, i.e., the third column in TABLE IV is used. For example, to construct the whole path of P_C^2, we look for the iteration where it is generated. It is Iteration 5, and $P_C^2 = P_D^2 \oplus r_{DC}^0$. P_D^2 is generated in Iteration 2 by $P_A^1 \oplus r_{AD}^3$. P_A^1 is generated in Iteration 1 by $P_W \oplus r_{WA}^0$. Thus, $P_C^2 = r_{WA}^0 \oplus r_{AD}^3 \oplus r_{DC}^0$, where P_W can be removed. In words, from the warehouse, the drone should first fly to stop A, take line 3 to arrive at stop D, and finally fly to the customer. Using the same

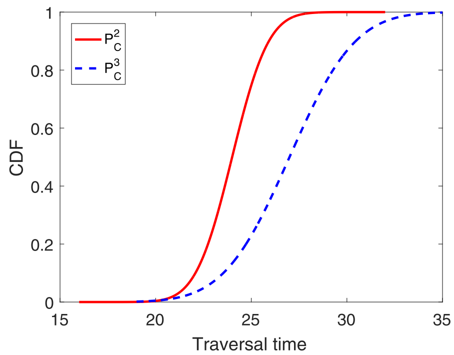


Fig. 6. The CDFs of the two found paths.

backtracking method, we find the whole path for P_C^3, i.e., $P_C^3 = r_{WB}^0 \oplus r_{BD}^5 \oplus r_{DC}^0$. The CDFs of these two paths are shown in Fig. 6. Clearly, expect for some small traversal times, the path P_C^2 is with higher probability than P_C^3 to realize a certain traversal time. For example, following P_C^2, the drone can reach the customer in 25 minutes with the probability of 0.78, while that of P_C^3 is only 0.25.

Remark 8: In the case study, the HPAR strategy is used to determine the vehicle that the drone can catch. The HPAR corresponds to a very high probability of 0.997 ($\beta - \alpha$), which is the three-sigma rule for normal distributions. If we relax this probability a bit, different paths with higher risk of not being realized can be found.

Moreover, we take into account the energy consumption of the UAV and apply Algorithm 2 to the example network. Here, $E_0 = 20$, $q_f = 1$, $q_h = 1$ and $\eta = 0.99$, and all the other parameters are the same as above. The computation process is summarized in TABLE V. As an additional constraint (22) is considered, there are some differences between TABLE V and TABLE IV. In particular, when path P_D^1 is constructed, the energy consumption distribution shown in the fifth column cannot satisfy (22). Thus, different from TABLE IV, this path is not added into the set of SE. This case occurs several times; see the paths P_E^2, P_E^3, P_E^4 and P_C^3. So, Algorithm 2 completes in fewer iterations. Finally, there is only one returned path, i.e., P_C^2, which is the feasible and shortest path in time.

V. Conclusion

This paper presented a new drone delivery system based on a public transportation network, which is promising to significantly extend the delivery range. We formulated the reliable drone path planning problem (RDPP), presented a stochastic model to characterize the path traversal time, and developed a label setting algorithm to construct a reliable path. Furthermore, we extended the model to take the battery lifetime into account, so that the constructed path is not only reliable but also feasible. The complexity of the proposed algorithm was discussed and how it works was shown through a case study. This algorithm constructs the warehouse-customer path and the return path for the drone can be further constructed in the same way.

References

[1] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, and D. Wu, "Proactive drone-cell deployment: Overload relief for a cellular network under flash crowd traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2877–2892, Oct. 2017.

[2] H. Huang and A. V. Savkin, "An algorithm of reactive collision free 3-D deployment of networked unmanned aerial vehicles for surveillance and monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 132–140, Jan. 2020.

[3] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 86–109, May 2015.

[4] Amazon.com. *Amazon Prime Air*. Accessed: Feb. 1, 2019. [Online]. Available: http://www.amazon.com/primeair

[5] *UPS Testing Drones for Use in Its Package Delivery System*. Accessed: Jun. 26, 2019. [Online]. Available: https://www.apnews.com/f34dc40191534203aa5d041c3010f6c5

[6] *DHL's Parcelcopter: Changing Shipping Forever*. Accessed: Jun. 26, 2019. [Online]. Available: https://discover.dhl.com/business/business-ethics/parcelcopter-drone-technology

[7] *SF Express Approved to Fly Drones to Deliver Goods*. Accessed: Jun. 26, 2019. [Online]. Available: https://www.caixinglobal.com/2018-03-28/sf-express-approved-to-fly-drones-to -deliver-goods-101227325.html

[8] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[9] M. Torabbeigi, G. J. Lim, and S. J. Kim, "Drone delivery schedule optimization considering the reliability of drones," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 1048–1053.

[10] P. Grippa, D. A. Behrens, F. Wall, and C. Bettstetter, "Drone delivery systems: Job assignment and dimensioning," *Auto. Robots*, vol. 43, no. 2, pp. 261–274, Feb. 2019.

[11] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1298–1308, Oct. 2015.

[12] S. Kim and I. Moon, "Traveling salesman problem with a drone station," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 1, pp. 42–52, Jan. 2019.

[13] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transp. Res. B, Methodol.*, vol. 122, pp. 350–364, Apr. 2019.

[14] S. M. Shavarani, M. G. Nejad, F. Rismanchian, and G. Izbirak, "Application of hierarchical facility location problem for optimization of a drone delivery system: A case study of Amazon prime air in the City of San Francisco," *Int. J. Adv. Manuf. Technol.*, vol. 95, nos. 9–12, pp. 3141–3153, Apr. 2018.

[15] I. Hong, M. Kuby, and A. T. Murray, "A range-restricted recharging station coverage model for drone delivery service planning," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 198–212, May 2018.

[16] S. M. Shavarani, S. Mosallaeipour, M. Golabi, and G. Izbirak, "A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system," *Comput. Oper. Res.*, vol. 108, pp. 57–68, Aug. 2019.

[17] H. D. Yoo and S. M. Chankov, "Drone-delivery using autonomous mobility: An innovative approach to future last-mile delivery problems," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2018, pp. 1216–1220.

[18] A. Trotta, F. D. Andreagiovanni, M. Di Felice, E. Natalizio, and K. R. Chowdhury, "When UAVs ride a bus: Towards energy-efficient city-scale video surveillance," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1043–1051.

[19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[20] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[21] E. Nikolova, M. Brand, and D. R. Karger, "Optimal route planning under uncertainty," in *Proc. 6th Int. Conf. Automated Planning Scheduling (ICAPS)*, vol. 6, 2006, pp. 131–141.

[22] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *J. ACM*, vol. 37, no. 3, pp. 607–625, Jul. 1990.

[23] B. Ding, J. X. Yu, and L. Qin, "Finding time-dependent shortest paths over large graphs," in *Proc. 11th Int. Conf. Extending Database Technol. Adv. Database Technol. (EDBT)*, 2008, pp. 205–216.

[24] M. P. Wellman, M. Ford, and K. Larson, "Path planning under time-dependent uncertainty," in *Proc. 11th Conf. Uncertainty Artif. Intell.* San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 532–539.

[25] W. Huang and J. Wang, "The shortest path problem on a time-dependent network with mixed uncertainty of randomness and fuzziness," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3194–3204, Nov. 2016.

[26] E. Miller-Hooks, "Adaptive least-expected time paths in stochastic, time-varying transportation and data networks," *Netw., Int. J.*, vol. 37, no. 1, pp. 35–52, 2001.

[27] B. Y. Chen, W. H. K. Lam, Q. Li, A. Sumalee, and K. Yan, "Shortest path finding problem in stochastic time-dependent road networks with stochastic First-In-First-Out property," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1907–1917, Dec. 2013.

[28] A. A. Prakash, "Pruning algorithm for the least expected travel time path on stochastic and time-dependent networks," *Transp. Res. B, Methodol.*, vol. 108, pp. 127–147, Feb. 2018.

[29] H. Frank, "Shortest paths in probabilistic graphs," *Oper. Res.*, vol. 17, no. 4, pp. 583–599, Aug. 1969.

[30] X. Wu and Y. M. Nie, "Modeling heterogeneous risk-taking behavior in route choice: A stochastic dominance approach," *Procedia-Social Behav. Sci.*, vol. 17, pp. 382–404, Jul. 2011.

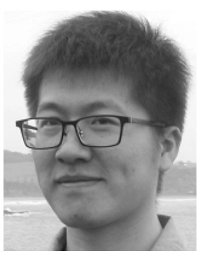
[31] Y. Zheng, Y. Zhang, and L. Li, "Reliable path planning for bus networks considering travel time uncertainty," *IEEE Intell. Transp. Syst. Mag.*, vol. 8, no. 1, pp. 35–50, Spring 2016.

[32] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Finding the shortest path in stochastic vehicle routing: A cardinality minimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1688–1702, Jun. 2016.

[33] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[34] Y. M. Nie and X. Wu, "Shortest path problem considering on-time arrival probability," *Transp. Res., B, Methodol.*, vol. 43, no. 6, pp. 597–613, Jul. 2009.

[35] Z. Dai, X. Ma, and X. Chen, "Bus travel time modelling using GPS probe and smart card data: A probabilistic approach considering link travel time and station dwell time," *J. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 175–190, Mar. 2019.



Hailong Huang was born in China, in 1988. He received the B.Sc. degree in automation from the China University of Petroleum, Beijing, China, in June 2012, and the Ph.D. degree in systems and control from the University of New South Wales, Sydney, Australia, in March 2018. He is currently a Research Associate with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. His current research interests include wireless sensor networks and guidance, navigation, and control of mobile robots.

Andrey V. Savkin was born in Norilsk, Russia, in 1965. He received the M.S. and Ph.D. degrees in mathematics from the Leningrad State University, Saint Petersburg, Russia, in 1987 and 1991, respectively. From 1987 to 1992, he was with the Television Research Institute, Leningrad, Russia. From 1992 to 1994, he held a Post-Doctoral Position with the Department of Electrical Engineering, Australian Defence Force Academy, Canberra. From 1994 to 1996, he was a Research Fellow with the Department of Electrical and Electronic Engineering and the Cooperative Research Centre for Sensor Signal and Information Processing, University of Melbourne, Australia. From 1996 to 2000, he was a Senior Lecturer, and an Associate Professor with the Department of Electrical and Electronic Engineering, The University of Western Australia, Perth. Since 2000, he has been a Professor with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW, Australia. He has authored/coauthored seven research monographs and numerous journals and conference papers on these topics. He has served as an Associate Editor for several international journals. His current research interests include robust control and state estimation, hybrid dynamical systems, guidance, navigation and control of mobile robots, applications of control and signal processing in biomedical engineering and medicine.

Chao Huang was born in China, in 1990. She received the B.Sc. degree in automation from the China University of Petroleum, Beijing, China, in June 2012, and the Ph.D. degree from the University of Wollongong, Wollongong, Australia, in December 2018. Her interests include motion planning, human-machine collaboration, fault tolerant, automotive control, and application.