

Joint Ground and Aerial Package Delivery Services: A Stochastic Optimization Approach

Suttinee Sawadsitang^{1b}, Dusit Niyato^{1b}, *Fellow, IEEE*, Puay-Siew Tan,
and Ping Wang^{1b}, *Senior Member, IEEE*

Abstract—Unmanned aerial vehicles, also known as drones, have emerged as a promising mode of fast, energy-efficient, and cost-effective package delivery. A considerable number of works have studied different aspects of drone package delivery service by a supplier, one of which is delivery planning. However, existing works addressing the planning issues consider a simple case of perfect delivery without service interruption, e.g., due to accident, which is common and realistic. Therefore, this paper introduces the joint ground and aerial delivery service optimization and planning (GADOP) framework. The framework explicitly incorporates uncertainty of drone package delivery, i.e., takeoff and breakdown conditions. The GADOP framework aims to minimize the total delivery cost, given practical constraints, e.g., travelling distance limit. Specifically, we formulate the GADOP framework as a three-stage stochastic integer programming model. To deal with the high complexity issue of the problem, a decomposition method is adopted. Then, the performance of the GADOP framework is evaluated by using two data sets including the Solomon benchmark suite and the real data from one of the Singapore logistics companies. The performance evaluation clearly shows that the GADOP framework can achieve significantly lower total payment than that of the baseline methods, which do not take uncertainty into account.

Index Terms—Ground-based delivery, drone delivery, UAV, stochastic optimization, uncertainty, breakdown.

I. INTRODUCTION

PACKAGE delivery becomes an instrumental function of logistics businesses due to the popularity of e-commerce such as online shopping. Traditionally, package delivery has been done through land transportation such as using trucks, cars, motorcycles, and bicycles, which is reasonably affordable, reliable, and accessible. However, due to rising labor costs, an alternative delivery mode by using unmanned

aerial vehicles (UAVs) such as drones becomes increasingly attractive. Thanks to the advancement of drone technologies, the cost and reliability of using drones in package delivery services have been improved significantly. Businesses start adopting the drone delivery such as Amazon Prime Air, DHL Parcelcopter, and Google Project Wing, showing the technological and economic feasibility of the services. The benefits of drone package delivery services are palpable as follows. Firstly, drone delivery is faster than land vehicles as drones are not subjected to road traffic jam. Also, it has much less service time as the drones can drop a package directly to the customers with minimal human involved. Secondly, drone delivery reduces resource usage in terms of manpower and energy. Thirdly, drone delivery is applicable to the areas which are difficult to access by land transportation.

Although drone delivery has many advantages over traditional ground-based delivery, using drones still faces many challenges.¹ For example, the reliability of drone delivery is lower than that of the ground-based delivery. Furthermore, drones have significantly less service coverage and capacity. As a result, using the drones for package delivery is limited to a certain area near the depot. Nevertheless, recently, an integration of ground-based delivery and drone delivery has been introduced [1]. It has been reported that these two delivery modes can complement each other because of their unique features and advantages. For example, packages can be optimally assigned to be delivered by trucks or drones to maximize service quality and to minimize cost. In this regard, a joint ground and aerial delivery service optimization and planning emerge as an important issue.

In this paper, we introduce the joint ground and aerial delivery service optimization and planning (GADOP) framework for a supplier. In particular, customers' packages can be delivered by trucks or drones. We consider the case that the supplier needs to make reservations for both trucks and drones in advance. The supplier, after deciding packages to be delivered by trucks, determines route for each truck. Similarly, the supplier determines package serving order of the drones. We model the uncertainty of the drone delivery in terms of takeoff and breakdown conditions. Specifically, the drone may not be able to take off, e.g., due to bad weather, and if it can take off, the drone may experience accident, e.g., due to technical issues. Therefore, we adopt the stochastic programming technique to formulate the joint ground and

Manuscript received March 6, 2018; revised June 28, 2018; accepted August 12, 2018. Date of publication October 5, 2018; date of current version May 29, 2019. This work was supported by the Singapore Institute of Manufacturing Technology, Nanyang Technological University Joint Laboratory, through the Collaborative Research Programme on Complex Systems. The Associate Editor for this paper was D. Sun. (*Corresponding author: Suttinee Sawadsitang.*)

S. Sawadsitang and D. Niyato are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: suttinee002@e.ntu.edu.sg; dniyato@ntu.edu.sg).

P.-S. Tan is with the Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research (A*STAR), Singapore 638075 (e-mail: pstan@simtech.a-star.edu.sg).

P. Wang is with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada (e-mail: pingw@yorku.ca).

Digital Object Identifier 10.1109/TITS.2018.2865893

¹In this paper, ground-based delivery refers to the delivery by trucks.

aerial delivery service optimization and planning in which the decisions are made in different stages. The contributions of this paper are summarized as follows.

- The proposed joint optimization is able to utilize and trade off benefits and limitations of truck and drone delivery. In particular, the proposed optimization can yield the minimum total payment² for the supplier by considering the routing path of trucks and the serving order of drones.
- The formulated stochastic programming model can find and achieve an optimal solution taking the uncertainty of the drone delivery into account. Consequently, the supplier can obtain the better planning for the truck and drone usage given certain constraints such as capacity limit and traveling distance limit.
- To reduce the computational time of obtaining the solution of the stochastic programming model, we introduce the decomposed GADOP by adopting the L-shape method. The decomposition allows the model to be split into multiple smaller sub-problems, which are more computationally tractable.

This paper is structured as follows. Section II presents the literature review of vehicle routing problem, ground and aerial delivery. The system model and the formulations of the proposed joint optimization are presented in Section III and Section IV, respectively. Section V presents the decomposition of the proposed joint optimization. Section VI presents the experiment setting and the numerical results of the proposed joint optimization. At the end, the conclusion and future works are discussed in Section VII.

II. RELATED WORK

Vehicle Routing Problem (VRP) has played a significant role in goods distribution for half century. Hundreds of different models and solution algorithms have been proposed since then. Many surveys on a variety of issues in VRP exist in the literature, e.g., [2] and [3]. Some of them consider specific aspects of the problem, e.g., VRP with time windows [4], pick-up and delivery VRP [5], dynamic VRP [6] and stochastic VRP [7], and solution algorithms [8], [9]. However, almost all of the existing VRP works consider only cars or trucks because they are the most popular mode of transportation. Recently, drones have been introduced which has a great potential to be used in businesses. A number of researchers and engineers studied and improved the technical aspects of drones, i.e., enhance endurance, safety, and features. Recently, planning and scheduling problems of drones have been studied in [10] and [11]. Sabo *et al.* [10] foresaw that drones could be used as mobile wireless communication nodes. In particular, the drones fly to collect data and return to transmit the data to the base station. The authors formulated the problem based on the pick-up and delivery VRP and added the communication constraints, which were not considered in the previous works. Sundar and Rathinam *et al.* [11] addressed the drone path planning problem, which is also one type

of VRP, by minimizing the total drone resource requirement, i.e., fuel or battery. In the problem, all the customers need to be visited once, and multiple depots are available for the drones to refuel or change their battery. The authors adopted resource constraint and developed a heuristic algorithm for the problem.

However, Sabo *et al.* [10] and Sundar and Rathinam [11] did not consider uncertainty, which is unavoidable in a real environment. A stochastic drone mission planning is presented in [12] and [13]. Evers *et al.* [12] proposed a drone planning model with the stochastic resource consumption, i.e., fuel or energy consumption, traveling from one location to another location. The authors presented the comparison between the robust solutions and the solution with average resource consumption. Instead of considering stochastic resource consumption, Evers *et al.* [13] considered the uncertainty in recording time and traveling time from one location to another location, as well as a visiting time interval of each location, which is known as time-windows. The authors combined two objectives together by adding and balancing two weight parameters. The objectives are to maximize (i) the expected profits of visiting the locations which are already known before taking off and (ii) the percentage of reaching new locations, which are not known before taking off, on time. The new locations are emergency and unplanned cases. The authors proposed a re-planning approach as integer programming and developed a heuristic algorithm to solve it. However, Evers *et al.* [12], [13] formulated the problems as orienteering problems. In particular, a solution of the orienteering problem is to visit only some locations while the VRP requires all locations to be visited.

For drone-related research, most of the existing works consider surveillance applications. Only few works are dedicated to parcel delivery [14]. The drone parcel delivery can be found in [1] and [14]. Dorling *et al.* [14] proposed the drone delivery planning considering drones' capacity, battery weights, and changing payload weights. The authors modeled multiple drones, and each drone is able to serve multiple customers before returning to the depot. Drones are also allowed to come back to refuel or charge a battery at the depot for multiple times. Murray and Chu [1] considered the heterogeneous VRP for package delivery using both trucks and drones. The authors proposed the Flying Sidekick Traveling Salesman Problem (FSTSP) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). In the FSTSP, the truck driver can set up a drone and let it visit different customers. After visiting and dropping packages to the assigned customers, the drone returns to the truck. Hence, the truck is referred to as a mobile depot for the drone separation and re-connection. On the other hand, in the PDSTSP, the authors formulated the problem with only one depot, but with multiple drones. Both the problems were formulated as a mix-integer programming model, where the objectives of the FSTSP and the PDSTSP are to minimize the serving time for both vehicles, i.e., one truck and one drone, and to minimize the traveling time of the truck, respectively.

However, all the above works ignored the random breakdown of the drone delivery which can happen more frequently

²In this paper, total cost and total payment refer to the same.

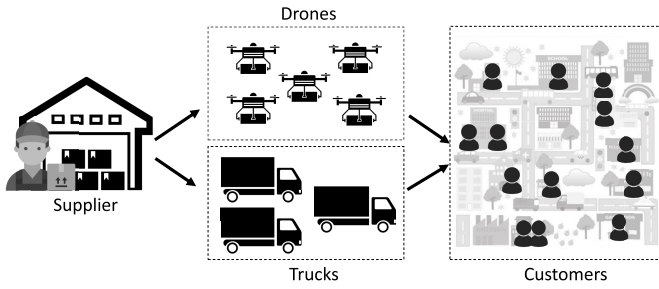


Fig. 1. The joint ground and aerial delivery service optimization and planning (GADOP) framework

than land transportation, e.g., truck. Therefore, we propose the joint ground and aerial delivery service optimization and planning (GADOP) framework considering realistic drone operations. For example, (i) drones cannot take off due to weather conditions such as raining and strong wind, (ii) drones may have technical problems during flying, and (iii) drones can have an accident such as a bird strike. Therefore, we formulate an optimization of the GADOP framework as a stochastic integer programming problem. We model the uncertainty of drones including takeoff and breakdown conditions. Furthermore, to accelerate the computational time of solving the problem, we adopt L-shape [15] to decompose the GADOP optimization.

III. SYSTEM MODEL AND ASSUMPTIONS

We consider package delivery service of a supplier. In this paper, we consider two delivery modes, i.e., truck delivery mode and drone delivery mode (Figure 1). The truck delivery mode is for the supplier to rent vehicles, e.g., lorry trucks, and use the vehicles to deliver packages to customers. The drone delivery mode is for the supplier to rent drones and use the drones to deliver packages to customers. While trucks have a higher initial cost, e.g., rental fee and driver wage, and a higher traveling cost, i.e., fuel, than those of the drones, the trucks have larger capacity and are more reliable.

Drone delivery is less reliable than the truck delivery as the drones can be affected by weather and other technical problems. Consequently, the drone delivery has a high breakdown probability. In this paper, we specifically consider two operability conditions of the drone, i.e., a takeoff condition and a breakdown condition. The takeoff condition indicates whether the drone can take off from the depot or not, which is observable before the drone departs. For example, the drone cannot takeoff from the depot if there is rain or strong wind. The breakdown condition happens after the drone already takes off from the depot, e.g., due to an accident. Therefore, the malfunctioned drone needs to be collected from the site and repaired, which also incurs a certain cost. Furthermore, the supplier will pay a penalty composed of the fee for outsourcing the delivery to a carrier and the compensation to the customer due to the delayed delivery.

The supplier faces a problem of utilizing trucks and drones for package delivery for their customers. Trucks and drones need to be reserved in advance. Furthermore, the supplier

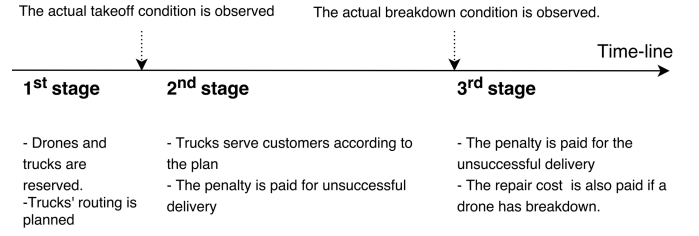


Fig. 2. The time-line of the scenarios.

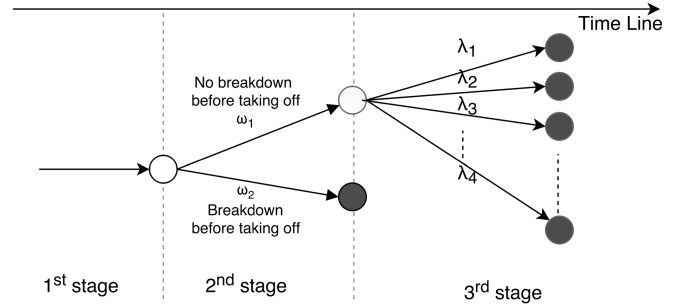


Fig. 3. An example of scenario tree with one drone.

has to take the uncertainties including drone takeoff and breakdown conditions into account. Therefore, we formulate the GADOP framework as a three-stage stochastic integer programming model.

- **First Stage:** the supplier decides (i) whether customers will be served by either trucks or drones, (ii) how many trucks and drones need to be reserved, (iii) the routing path of trucks, and (iv) the serving order of drones' delivery. The decisions are made based on the probability of the uncertainties.
- **Second Stage:** the supplier calculates the penalty payment for customers which cannot be served due to the drone takeoff condition.
- **Third Stage:** the supplier calculates the penalty payment for customers which cannot be served due to the drone breakdown condition. The breakdown condition is associated with each drone and each customer. Additionally, the supplier takes the repair cost into account when the breakdown happens.

Figure 2 shows the stages of the GADOP framework. In the first stage, the decisions are made before the actual drone take-off condition is observed. In the second stage, the decisions are made after the takeoff condition is observed and before the delivery begins. The third stage decisions are made when an actual breakdown of the drone is observed. Figure 3 shows an example of a scenario tree with one drone for the supplier. The formal definition of the scenarios is presented in Section III-A.

The supplier has a set of customers to be served denoted by $\mathcal{C} = \{C_1, C_2, \dots, C_{c'}\}$, where c' denotes the total number of customers. Without loss of generality, each customer in \mathcal{C} has one package to be delivered, and g_i denotes the package weight of customer i . The supplier will assign customers' packages to either trucks or drones in the first stage.

We assume that all the customers have a dropoff station for the drone delivery. The supplier can choose to reserve and use trucks from the set $\mathcal{T} = \{T_1, T_2, \dots, T_{t'}\}$ and drones from the set $\mathcal{D} = \{D_1, D_2, \dots, D_{d'}\}$. t' and d' denote the total numbers of available trucks and drones, respectively. Every truck in the set \mathcal{T} has a capacity limit (\bar{f}_t), daily distance traveling limit (\bar{l}_t), and daily traveling time limit (\bar{h}_t). The package dropoff time of a truck is denoted by \bar{r} . Similarly, every drone in the set \mathcal{D} has a capacity limit (\bar{f}_d), daily flying distance limit (\bar{l}_d), and flying distance limit per trip (\bar{e}_d). The package dropoff time of a drone is assumed to be negligible.

The supplier has one depot. The traveling distance from location i' to location j' is represented as $k_{i',j'}$, where i' and j' are the locations being members of the set $\mathcal{C} \cup \{0\}$, where 0 represents the depot. The average truck driving speed and the drone flying speed are denoted by $\bar{q}_{i',j'}$ and $\hat{q}_{i',j'}$, respectively. We assume that a drone can carry only one package per trip, and therefore the drone needs to return to the depot after delivering one package. A similar assumption is made in the literature, e.g., [1] and [16].

A. Uncertainty

The drone delivery is much less reliable than the truck delivery [18]. Therefore, we consider the uncertainty of the drones including takeoff and breakdown conditions. Again, the takeoff condition and breakdown condition are observable in the second and third stages, i.e., before and after taking off, respectively. Let $\Omega = \{\omega_1, \omega_2, \dots, \omega_{\omega'}\}$ be the set of the takeoff scenarios, where ω' represents the total number of the scenarios. The takeoff scenario is defined as $\omega = (\mathbb{R}_1, \mathbb{R}_2, \dots, \mathbb{R}_{d'})$, the maximum number of takeoff scenarios is based on the total number of drones, i.e., d' . $\mathbb{R}_d = 1$ when drone D_d cannot take off from the depot, and $\mathbb{R}_d = 0$ otherwise. For example, $\omega = (0, 0, 1)$ means that the supplier has three drones, and drones D_1 and D_2 can take off while drone D_3 cannot take off from the depot. Let $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{\lambda'}\}$ denote the set of breakdown scenarios, where $\lambda_{\lambda'}$ represents the total number of the scenario. The breakdown scenario λ contains the breakdown parameters indicating which drones are broken during the delivery. The scenario is defined as follows:

$$\lambda = \begin{bmatrix} \mathbb{B}_{1,1} & \mathbb{B}_{1,2} & \dots & \mathbb{B}_{1,d'} \\ \mathbb{B}_{2,1} & \mathbb{B}_{2,2} & \dots & \mathbb{B}_{2,d'} \\ \vdots & \vdots & \dots & \vdots \\ \mathbb{B}_{c',1} & \mathbb{B}_{c',2} & \dots & \mathbb{B}_{c',d'} \end{bmatrix}, \quad (1)$$

where $\mathbb{B}_{i,d} = 1$ when drone D_d breaks down while serving customer C_i , and $\mathbb{B}_{i,d} = 0$ otherwise. Additionally, $\mathbb{P}(\omega)$ and $\mathbb{P}(\lambda)$ denote the probabilities that scenario ω and scenario λ will happen, respectively. In reality, the probabilities can be obtained from the history records.

B. Payments

The supplier involves six types of payments, i.e., costs, including (i) the initial costs of trucks, (ii) the initial costs of drones, (iii) the traveling costs of trucks, (iv) the traveling costs of drones, (v) penalty of unsuccessful drone delivery,

and (vi) drone repair cost. The initial costs are the fixed costs such as rental fee, driver or staff wages. $\bar{c}_t^{(i)}$ and $\hat{c}_d^{(i)}$ represent the initial costs of trucks and the initial costs of drones, respectively. The traveling costs are functions of traveling distance, fuel cost, or battery recharge cost. $\bar{c}_t^{(r)}$ and $\hat{c}_d^{(r)}$ represent the traveling costs of trucks and the traveling costs of drones, respectively. The penalty and the repair cost, denoted as p and m , respectively, are incurred when a drone does not successfully deliver a package.

IV. PROBLEM FORMULATION

Minimize:

$$\sum_{t \in \mathcal{T}} \bar{c}_t^{(i)} \bar{W}_t + \sum_{d \in \mathcal{D}} \hat{c}_d^{(i)} \hat{W}_d + \sum_{i', j' \in \mathcal{C} \cup \{0\}} \sum_{t \in \mathcal{T}} \bar{c}_t^{(r)} V_{i',j',t} + \mathbb{E}(\mathcal{L}(\hat{X}_{i,d})), \quad (2)$$

where

$$\begin{aligned} \mathcal{L}(\hat{X}_{i,d}) = & \sum_{\omega \in \Omega} \mathbb{P}(\omega) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \bar{c}_i^{(r)} \hat{X}_{i,d} (1 - \mathbb{R}_d(\omega)) \right. \\ & \left. + \sum_{d \in \mathcal{D}} p Z_{i,d}^{(b)}(\omega) + \mathbb{E}(\mathcal{M}(\hat{X}_{i,d}, \omega)) \right), \end{aligned} \quad (3)$$

$$\begin{aligned} \mathcal{M}(\hat{X}_{i,d}, \omega) = & \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p Z_{i,d}^{(a)}(\omega, \lambda) \right. \\ & \left. + \sum_{d \in \mathcal{D}} m Z_d^{(m)}(\omega, \lambda) \right), \end{aligned} \quad (4)$$

subject to: (5) to (37).

The GADOP framework is a three-stage stochastic integer programming as modeled in (2) for the first stage, (3) for the second stage, and (4) for the third stage. $\mathbb{E}(\mathcal{L}(\hat{X}_{i,d}))$ denotes the expectation of the traveling cost and the penalty of drones over random takeoff conditions occurred in the second stage. $\mathbb{E}(\mathcal{M}(\hat{X}_{i,d}, \omega))$ denotes the expectation of the penalty cost and the repair cost of drones over random takeoff conditions and random breakdown conditions occurred in the third stage. The objective is to minimize the total delivery payments as discussed in Section III-B. The decision variables in the framework are listed below.

- \bar{W}_t is an indicator whether truck t will be used or not. When $\bar{W}_t = 1$, truck t will be used, and $\bar{W}_t = 0$ otherwise.
- \hat{W}_d is an indicator whether drone d will be used or not. When $\hat{W}_d = 1$, drone d will be used, and $\hat{W}_d = 0$ otherwise.
- $V_{i',j',t}$ is a routing variable in which $V_{i',j',t} = 1$ if the route from location i' to location j' will be used by truck t , and $V_{i',j',t} = 0$ otherwise.
- $\hat{X}_{i,d}$ is an allocation variable of drone d . $\hat{X}_{i,d} = 1$ if drone d will serve customer i , and $\hat{X}_{i,d} = 0$ otherwise.
- $\bar{X}_{i,t}$ is an allocation variable of truck t . $\bar{X}_{i,t} = 1$ if truck t will serve customer i , and $\bar{X}_{i,t} = 0$ otherwise.
- $Z_{i,d}^{(b)}(\omega)$ is a penalty variable. $Z_{i,d}^{(b)}(\omega) = 1$ if the penalty associated to customer i , which will be served by drone d ,

but the drone cannot take off from the depot, is paid in scenario ω , and $Z_{i,d}^{(b)}(\omega) = 0$ otherwise.

- $Z_{i,d}^{(a)}(\omega, \lambda)$ is another penalty variable. $Z_{i,d}^{(a)}(\omega, \lambda) = 1$ if the penalty associated to customer i , which will be served by drone d , but the drone breaks down, is paid in scenarios ω and λ , $Z_{i,d}^{(a)}(\omega, \lambda) = 0$ otherwise.
- $Z_d^{(m)}(\omega, \lambda)$ is a repair variable. $Z_d^{(m)}(\omega, \lambda) = 1$ if drone d in scenario ω and λ breaks down during delivery, and $Z_d^{(m)}(\omega, \lambda) = 0$ otherwise.
- $S_{i,t}$ is an auxiliary variable for sub-tour elimination in truck delivery.
- $U_{i,d}$ is a serving order of the drones. $U_{i,d} < U_{j,d}$ means that drone d will serve customer i before customer j .
- $M_{i,j,d}$ is an auxiliary binary variable for ensuring the nonequivalent of $U_{i,d}$.

The types and bounds of all decision variables are indicated in constraints (34)–(37).

The constraints in (5) and (6) ensure that the initial cost of trucks and the initial cost of drones will be paid if they are used, where Δ is any large number $\Delta \geq c'$. The constraints in (7) and (8) ensure that the total weight of packages does not exceed the capacity limit of a truck and a drone, respectively. The constraint in (9) ensures that the flying distance between the depot to a customer and vice versa does not exceed the flying limit. The daily traveling distance limits are controlled by the constraints in (10) and (11) for drone delivery and truck delivery, respectively. The total time of the truck delivery, i.e., traveling time and serving time, must not exceed the working hours as imposed in the constraint in (12).

$$\sum_{i \in C} \bar{X}_{i,t} \leq \Delta \bar{W}_t, \quad \forall t \in T \quad (5)$$

$$\sum_{i \in C} \hat{X}_{i,d} \leq \Delta \hat{W}_d, \quad \forall d \in D \quad (6)$$

$$\sum_{i \in C} g_i \bar{X}_{i,t} \leq \bar{f}_t, \quad \forall t \in T \quad (7)$$

$$g_i \hat{X}_{i,d} \leq \hat{f}_d, \quad \forall i \in C, \forall d \in D \quad (8)$$

$$(k_{0,i} + k_{i,0}) \hat{X}_{i,d} \leq \hat{e}_d, \quad \forall i \in C, \forall d \in D \quad (9)$$

$$\sum_{i \in C} (k_{0,i} + k_{i,0}) \hat{X}_{i,d} \leq \hat{l}_d, \quad \forall d \in D \quad (10)$$

$$\sum_{\substack{i',j' \in \\ C \cup \{0\}}} k_{i',j'} V_{i',j',t} \leq \bar{l}_t, \quad \forall t \in T \quad (11)$$

The allocation of trucks and drones is imposed by the constraint in (13), where one customer can be served by either a truck or a drone. The constraints in (14)–(19) are to find the route for truck delivery. Only the allocated truck can take the arrival and departing route to serve the customer. The constraint in (19) ensures that no subtour exists in the solution.

If drones cannot take off from the depot, the penalty must be paid for all associated customers, as indicated in the constraint in (20). In the case that the drones can take off, but they are broken after departing, the constraint in (21) ensures that the penalty must be paid. When breakdown occurs after departing from the depot, the drone will be no longer able to serve all the next customers. Consequently, the penalty associated to these

customers must be paid as indicated in the constraint in (22). Note that the next customer j after customer i has a higher serving order, i.e., $U_{j,d} > U_{i,d}$. Moreover, the constraint in (23) ensures that the repair cost must be paid if the drone is broken after departing from the depot.

The constraints in (24)–(27) decide serving orders of drones. The constraint in (24) ensures that $U_{i,d}$ must not be zero when drone d is allocated to serve customer i , i.e., $\bar{X}_{i,d} = 1$. $U_{i,d} = 0$ which represents the drone serving order is thus not needed. The constraints in (26)–(27) ensure that the serving orders must not be the same when $\bar{X}_{i,d} = 1$. Again, $M_{i,j,d}$ is an auxiliary binary variable for ensuring that $U_{i,d} \neq U_{j,d}$ when $\bar{X}_{i,d} = 1$.

$$\sum_{\substack{i',j' \in \\ C \cup \{0\}}} \left(\frac{k_{i',j'}}{q_{i',j'}} + \bar{r} \right) V_{i',j',t} \leq \bar{h}_t, \quad \forall t \in T \quad (12)$$

$$\sum_{t \in T} \bar{X}_{i,t} + \sum_{d \in D} \hat{X}_{i,d} = 1, \quad \forall i \in C \quad (13)$$

$$\sum_{i \in C} V_{0,i,t} \leq 1, \quad \forall t \in T \quad (14)$$

$$\sum_{i \in C} V_{i,0,t} \leq 1, \quad \forall t \in T \quad (15)$$

$$\sum_{i' \in C \cup \{0\}} V_{i',i,t} = \bar{X}_{i,t}, \quad \forall t \in T, \forall i \in C \quad (16)$$

$$\sum_{i' \in C \cup \{0\}} V_{i,i',t} = \bar{X}_{i,t}, \quad \forall t \in T, \forall i \in C \quad (17)$$

$$V_{i',i',t} = 0, \quad \forall t \in T, \forall i' \in C \cup \{0\} \quad (18)$$

$$S_{i,t} - S_{j,t} + c' V_{i,j,t} \leq c' - 1, \quad \forall i \in C, \forall j \in C, \forall t \in T \quad (19)$$

$$\hat{X}_{i,d} \mathbb{R}_d(\omega) = Z_{i,d}^{(b)}(\omega), \quad \forall i \in C, \forall d \in D, \forall \omega \in \Omega \quad (20)$$

$$\hat{X}_{i,d} (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda) \leq Z_{i,d}^{(a)}(\omega, \lambda), \quad \forall i \in C, \forall d \in D, \forall \omega \in \Omega, \lambda \in \Lambda \quad (21)$$

$$(U_{i,d} - U_{j,d}) \leq \Delta \left(1 - Z_{j,d}^{(a)}(\omega, \lambda) + Z_{i,d}^{(a)}(\omega, \lambda) \right), \quad \forall i, j \in C, \quad i \neq j, \forall d \in D, \forall \omega \in \Omega, \lambda \in \Lambda \quad (22)$$

$$Z_d^{(m)}(\omega, \lambda) \geq Z_{i,d}^{(a)}(\omega, \lambda), \quad \forall i \in C, \forall d \in D, \forall \omega \in \Omega, \lambda \in \Lambda \quad (23)$$

$$\hat{X}_{i,d} \leq U_{i,d}, \quad \forall d \in D, \forall i \in C \quad (24)$$

$$0 \leq U_{j,d} \leq \sum_{i \in C} \hat{X}_{i,d}, \quad \forall d \in D, \forall j \in C \quad (25)$$

$$U_{i,d} - U_{j,d} \leq \Delta M_{i,j,d} - \hat{X}_{i,d}, \quad \forall i, j \in C, \quad i \neq j, \forall d \in D \quad (26)$$

$$U_{i,d} - U_{j,d} \geq \hat{X}_{i,d} - \Delta (1 - M_{i,j,d}), \quad \forall i, j \in C, \quad i \neq j, \forall d \in D \quad (27)$$

In order to enhance customers' satisfaction, suppliers allow their customers to specific a delivery time slot, which can be referred to as time windows. Time windows constraint can be

hard time windows or soft time windows. In the hard time window, all customers must be served within time windows. On the other hand, the soft time window is more relaxed, and thus the supplier can choose to pay for the penalty of not delivery on time. We consider the hard time window in this paper. The constraints in (28)-(29) ensures that the delivery is done within the assigned time windows. $\mathcal{L}^{(m)}$ and $\mathcal{L}^{(a)}$ denote a set of customers who require to be served in the morning and afternoon, respectively, and we have $\mathcal{L}^{(m)} \subseteq \mathcal{C}$, $\mathcal{L}^{(a)} \subseteq \mathcal{C}$ and $\mathcal{L}^{(m)} \cap \mathcal{L}^{(a)} = \emptyset$. The constraints in (28) and (29) ensure that all morning customers must be served before the afternoon customers. The constraints in (30)-(33) ensure that traveling time of trucks and drones does not exceed the morning and afternoon traveling time limits, i.e., time windows. The morning and afternoon traveling time limits are defined as $l^{(m)}$ and $l^{(a)}$, respectively. More specific time slots and explanation of hard time window can be found in [17].

$$U_{i,d} \leq U_{j,d} + \Delta(1 - \hat{X}_{i,t}), \quad \forall i \in \mathcal{L}^{(m)}, \quad \forall j \in \mathcal{L}^{(a)}, \quad \forall d \in \mathcal{D}, \quad \forall \omega \in \Omega \quad (28)$$

$$S_{i,t} \leq S_{j,t} + \Delta(1 - \bar{X}_{i,t}), \quad \forall i \in \mathcal{L}^{(m)}, \quad \forall j \in \mathcal{L}^{(a)}, \quad \forall t \in \mathcal{T}, \quad \forall \omega \in \Omega \quad (29)$$

$$\sum_{m \in \mathcal{L}^{(m)}} \left(\frac{k_{0,m}}{\hat{q}_{0,m}} + \frac{k_{m,0}}{\hat{q}_{m,0}} \right) \hat{X}_{m,d} \leq l^{(m)}, \quad \forall d \in \mathcal{D} \quad (30)$$

$$\sum_{a \in \mathcal{L}^{(a)}} \left(\frac{k_{0,a}}{\hat{q}_{0,a}} + \frac{k_{a,0}}{\hat{q}_{a,0}} \right) \hat{X}_{a,d} \leq l^{(a)}, \quad \forall d \in \mathcal{D} \quad (31)$$

$$\sum_{i' \in \mathcal{C} \cup \{0\}} \sum_{m \in \mathcal{L}^{(m)}} \left(\frac{k_{i',m}}{\hat{q}_{i',m}} + \bar{r} \right) V_{i',m,t} \leq l^{(m)}, \quad \forall t \in \mathcal{T} \quad (32)$$

$$\sum_{i \in \mathcal{C}} \sum_{a' \in \mathcal{L}^{(a)} \cup \{0\}} \left(\frac{k_{i,a'}}{\hat{q}_{i,a'}} + \bar{r} \right) V_{i,a',t} \leq l^{(a)}, \quad \forall t \in \mathcal{T} \quad (33)$$

$$\hat{X}_{i,d}, \bar{X}_{i,t}, \hat{W}_d, \bar{W}_t \in \{0, 1\}, \quad \forall d \in \mathcal{D}, \quad \forall t \in \mathcal{T}, \quad \forall i \in \mathcal{C} \quad (34)$$

$$V_{i',j',t} \in \{0, 1\}, \quad \forall i', j' \in \mathcal{C} \cup \{0\}, \quad \forall t \in \mathcal{T} \quad (35)$$

$$U_{i,d}, S_{i,t} \in \{0, 1, \dots, c'\}, \quad \forall d \in \mathcal{D}, \quad \forall t \in \mathcal{T}, \quad \forall i \in \mathcal{C} \quad (36)$$

$$Z_{i,d}^{(b)}(\omega), Z_{i,d}^{(a)}(\omega, \lambda), Z_d^{(m)}(\omega, \lambda), M_{i,j,d} \in \{0, 1\}, \quad \forall d \in \mathcal{D}, \quad \forall i, j \in \mathcal{C}, \quad \omega \in \Omega, \quad \lambda \in \Lambda \quad (37)$$

The above optimization problem can be complex to solve when the numbers of customers, trucks, and drones grow. Therefore, in the next section, we present the decomposition method to obtain an optimal solution of the problem more efficiently.

V. DECOMPOSITION

The GADOP optimization, which is presented in the last section, is an NP-hard problem. The computational time for solving the problem depends on the numbers of variables, parameters, and constraints. To address the computational issue, we decompose the GADOP optimization into multiple sub-problems according to the decision stages, i.e., one first-stage problem, ω' second-stage sub-problems,

Algorithm 1: Decomposition Algorithms, Where k Is the Iteration Index

Result: The optimal solution is taken from the latest iteration

while *Convergence condition is not met* **do**

if $k = 0$ **then**

 //first iteration;

 - θ_1^{*k} and θ_2^{*k} are set at very small values;

 - Solving master problem without (39) and (40);

else

 - Generating the feedback constraints as in (39) and (40) by using $E_{i,d}^{(p),k-1}$, $E_{i,d}^{(m),k-1}$, and $E_{i,d}^{(r),k-1}$;

 - Solving master problem with the feedback constraints;

end

 - Obtaining $X_{i,d}^{*k}$, θ_1^{*k} , and θ_2^{*k} ;

foreach *scenario* $\omega \in \Omega$ **do**

 - Solving the second-stage sub-problem by using

$X_{i,d}^{*k}$ as parameters;

 - Solving the third-stage sub-problem by using

$X_{i,d}^{*k}$ as parameters;

end

 - Calculating $E_{i,d}^{(p),k}$, $E_{i,d}^{(m),k}$, and $E_{i,d}^{(r),k}$, $\forall d \in \mathcal{D}, \forall i \in \mathcal{C}$, as in (52), (54), and (51), respectively;

 - Calculating B as in (56);

if $\theta_1^{*k} + \theta_2^{*k} \geq B$ **then**

 - Convergence condition is met;

else

 - $k \leftarrow k + 1$;

end

end

and ω' third-stage sub-problems. Each second-stage sub-problem and each third-stage sub-problem are associated with a certain takeoff scenario, i.e., $\omega \in \Omega$. The decomposition is executed iteratively as described in the following steps:

- (i) Solving the first-stage problem, which is the master problem,
- (ii) Solving all the second-stage sub-problems and the third-stage sub-problems using the results from the master problem as the parameters,
- (iii) Calculating feedback parameters, i.e., $E_{i,d}^{(p)}$, $E_{i,d}^{(r)}$, and $E_{i,d}^{(m)}$ representing the penalty, the traveling cost of drones, and drone repair cost from the second-stage sub-problems and third-stage sub-problem, and
- (iv) Checking the convergence condition, if the convergence condition is met, then the solution is found, and otherwise the feedback parameters are sent to the master problem, and Step (i) repeats with the additional constraints applying the feedback parameters.

This decomposition is based on the L-Shape method [15]. The algorithm to execute the decomposition is shown in Algorithm 1.

Figure 4 presents an example of the GADOP optimization with five takeoff scenarios and ten breakdown scenarios.

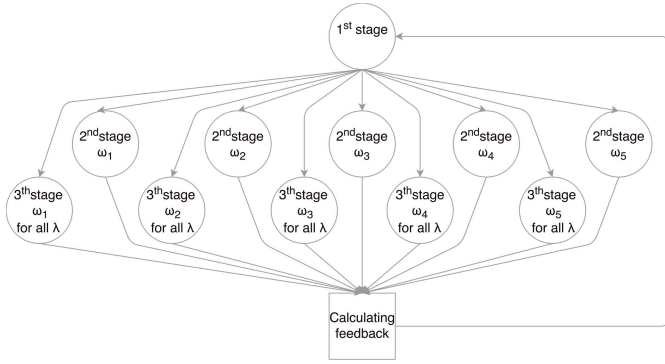


Fig. 4. Decomposition of scenarios and stages where a circle represents one optimization master problem or sub-problem.

After decomposed, the example has one master problem, five second-stage sub-problems, and five third-stage sub-problems to be solved. Note that the numbers of the second-stage sub-problems and the third-stage sub-problems depend on the number of the takeoff scenarios. In fact, after applying the decomposition, we have found that the solution of the second-stage problem can be obtained by a simple calculation due to special structure of the GADOP problem. As a result, for the same takeoff scenario, the second-stage sub-problem and the third-stage sub-problem do not have any interdependency with each other, and thus we can solve the second-stage sub-problems and the third-stage sub-problems simultaneously. Additionally, a takeoff scenario does not have any interdependency with other takeoff scenarios. However, dependencies of breakdown scenarios exist in serving orders ($U_{i,d}$). Since the serving orders need to be decided before the delivery begins, the same serving order ($U_{i,d}$) is used for all breakdown scenarios. As a result, the third-stage optimization cannot be further decomposed according to breakdown scenarios.

A. Master Problem or First-Stage Problem

Minimizing:

$$\begin{aligned} & \sum_{t \in \mathcal{T}} \bar{c}_t^{(i)} \bar{W}_t + \sum_{d \in \mathcal{D}} \bar{c}_d^{(i)} \bar{W}_d \\ & + \sum_{\substack{i', j' \in \mathcal{T} \\ C \cup \{0\}}} \bar{c}_{i', j'}^{(r)} V_{i', j', t} + \theta_1 + \theta_2, \end{aligned} \quad (38)$$

subject to: (5) to (19), (29) to (33), (39), and (40).

$$\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} (E_{i,d}^{(p)} + E_{i,d}^{(r)}) \hat{X}_{i,d} + \theta_1 \geq 0 \quad (39)$$

$$\sum_{d \in \mathcal{D}} E_{i,d}^{(m)} \bar{W}_d + \theta_2 \geq 0, \quad \forall i \in \mathcal{C} \quad (40)$$

The objective function of the master problem is presented in (38). The constraints in (39) and (40) are the feedback cutting constraints [15]. Again, $E_{i,d}^{(p)}$, $E_{i,d}^{(r)}$, and $E_{i,d}^{(m)}$ are the feedback parameters, which are dedicated to the penalty, the drone traveling cost, and the drone repair cost, respectively. θ_1 and θ_2 are initially set at small values for the first iteration. After the first iteration, θ_1 and θ_2 become positive

decision variables. The results of θ_1 and θ_2 are denoted by θ_1^* and θ_2^* , respectively, which are used in checking the convergence condition. The result of $\hat{X}_{i,d}$ is denoted by $\hat{X}_{i,d}^*$, which is used as a parameter in the second-stage sub-problems and the third-stage sub-problems. After decomposition, the first-stage problem is still an NP-hard problem. However, the first-stage problem becomes less complex than the original GADOP because the number of decision variables in the first-stage problem is fewer than that in the original problem.

B. Second-Stage Sub-Problem

Minimize:

$$\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d}^* (1 - \bar{\mathbb{R}}_d) + \sum_{d \in \mathcal{D}} p \tilde{Z}_{i,d}^{(b)}, d \quad (41)$$

subject to: (42).

$$\hat{X}_{i,d}^* \bar{\mathbb{R}}_d = \tilde{Z}_{i,d}^{(b)}, \quad \forall i \in \mathcal{C}, \quad \forall d \in \mathcal{D} \quad (42)$$

We decompose the second-stage sub-problem given a certain takeoff scenario, and thus the variable ω is removed from the objective function in (41) and the constraint in (42). To minimize the expected cost in the second stage, i.e., $\mathbb{E}(\mathcal{L}(\hat{X}_{i,u}))$, we need to solve all the second-stage sub-problems associated to a takeoff scenario (ω). Note that random parameters $\bar{\mathbb{R}}_d$ and $\bar{\mathbb{B}}_{i,d}$, i.e., for scenario variables \mathbb{R}_d and $\mathbb{B}_{i,d}$ defined in Section III-A, respectively, as well as the decision variable $\tilde{Z}_{i,d}^{(b)}$ is associated with the takeoff scenario, and therefore the ω is omitted.

In general, the second-stage problem is an optimization, and thus we present the second-stage problem in the optimization form. Nevertheless, in our exact problem, there is a special structure that makes the second-stage problem is able to be solved directly through the assignment of $\hat{X}_{i,d}$ and $\bar{\mathbb{R}}_d$.

After applying the decomposition, there is only one decision variable in the second-stage problem that is $\tilde{Z}_{i,d}^{(b)}$. Furthermore, the decision variable $\hat{X}_{i,d}$ is solved in the first-stage problem, and thus it becomes a parameter, i.e., $\hat{X}_{i,d}^*$, in the second-stage problem. According to the constraint in (42), $\tilde{Z}_{i,d}^{(b)}$ depends on parameters only, and consequently the second-stage problem is no longer an optimization problem.

C. Third-Stage Sub-Problem

Minimize:

$$\sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) \left(\sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p \tilde{Z}_{i,d}^{(a)}(\lambda) + \sum_{d \in \mathcal{D}} m \tilde{Z}_d^{(m)}(\lambda) \right), \quad (43)$$

subject to: (24), (44) to (48).

$$\hat{X}_{i,d}^* (1 - \bar{\mathbb{R}}_d) \bar{\mathbb{B}}_{i,d} \leq \tilde{Z}_{i,d}^{(a)}(\lambda), \quad \forall i \in \mathcal{C}, \quad \forall d \in \mathcal{D} \quad (44)$$

$$U_{i,d} - U_{j,d} \leq \Delta \left(1 - \tilde{Z}_{j,d}^{(a)}(\lambda) + \tilde{Z}_{i,d}^{(a)}(\lambda) \right), \quad \forall i, j \in \mathcal{C}, \quad i \neq j, \quad \forall d \in \mathcal{D} \quad (45)$$

$$\tilde{Z}_d^{(m)}(\lambda) \geq \tilde{Z}_{i,d}^{(a)}(\lambda), \quad \forall i \in \mathcal{C}, \quad \forall d \in \mathcal{D} \quad (46)$$

$$\hat{X}_{i,d}^* \leq U_{i,d}, \quad \forall d \in \mathcal{D}, \quad \forall i \in \mathcal{C} \quad (47)$$

$$0 \leq U_{j,d} \leq \sum_{i \in \mathcal{C}} \hat{X}_{i,d}^*, \quad \forall d \in \mathcal{D}, \quad \forall j \in \mathcal{C} \quad (48)$$

$$U_{i,d} - U_{j,d} \leq \Delta M_{i,j,d} - \widehat{X}_{i,d}^*, \quad \forall i, j \in \mathcal{C}, \quad i \neq j, \quad \forall d \in \mathcal{D} \quad (49)$$

$$U_{i,d} - U_{j,d} \geq \widehat{X}_{i,d}^* - \Delta(1 - M_{i,j,d}), \quad \forall i, j \in \mathcal{C}, \quad i \neq j, \quad \forall d \in \mathcal{D} \quad (50)$$

Similar to the second-stage sub-problem, each third-stage sub-problem is formulated given a certain takeoff scenario. The objective function of the third-stage sub-problem is expressed in (43) which is the expected cost of the penalty and repair cost of drones. The constraints in (44), (45), and (46) contain the third-stage decision variables, which are $\tilde{Z}_{i,d}^{(a)}(\lambda)$ and $\tilde{Z}_d^{(m)}(\lambda)$. The decision variables $\tilde{Z}_{i,d}^{(a)}(\lambda)$ and $\tilde{Z}_d^{(m)}(\lambda)$ are associated with the takeoff scenario, and therefore ω is omitted. Since there is the interdependency among $U_{i,d}$, i.e., drone serving order, the constraints in (47)-(50) are directly adopted in this sub-problem. The reason is that $U_{i,d}$ of each takeoff scenario, which depends on $\widehat{X}_{i,d}^*$, has already obtained in the master problem. Again, $\widehat{X}_{i,d}^*$ is the assignment of a drone to a customer.

D. Calculating Feedback Parameters and Convergence

This section explains the feedback parameter calculation and the convergence condition checking. $e_{i,d}^{(p)}(\omega)$ and $e_{i,d}^{(m)}(\omega)$ are the auxiliary parameters in the third-stage sub-problems, which are associated with the drone penalty and the drone repair cost, respectively. The feedback parameters, i.e., $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$, as expressed in (51), (52), and (54), respectively, are returned to the master problem. $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$ are the expected values associated with the drone traveling cost, the drone penalty, and the drone repair cost, respectively. The constraints in (39) and (40) ensure that the supplier pays the routing cost, the penalty, and the repair cost from the second-stage and third-stage when the associated $\widehat{X}_{i,d}$ is selected. Therefore, $E_{i,d}^{(r)}$, $E_{i,d}^{(p)}$, and $E_{i,d}^{(m)}$ must be negative in order to match with θ_1 and θ_2 . The formulations of the feedback parameters are given as follow.

$$E_{i,d}^{(r)} = -\widehat{c}_{i,d}^{(r)} \sum_{\omega \in \Omega} \mathbb{P}(\omega) (1 - \mathbb{R}_d(\omega)), \quad (51)$$

$$E_{i,d}^{(p)} = -\sum_{\omega \in \Omega} \mathbb{P}(\omega) (p \mathbb{R}_d(\omega) + e_{i,d}^{(p)}(\omega)), \quad (52)$$

where

$$e_{i,d}^{(p)}(\omega) = p \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda), \quad (53)$$

$$E_{i,d}^{(m)} = -\sum_{\omega \in \Omega} \mathbb{P}(\omega) e_{i,d}^{(m)}(\omega), \quad (54)$$

where

$$e_{i,d}^{(m)}(\omega) = m \sum_{\lambda \in \Lambda} \mathbb{P}(\lambda) (1 - \mathbb{R}_d(\omega)) \mathbb{B}_{i,d}(\lambda). \quad (55)$$

We next define a convergence parameter denoted by B which is used for checking the convergence condition. If B is less than or equal to $\theta_1^* + \theta_2^*$, the convergence condition is met, and then Algorithm 1 is terminated and the solution is

taken from the latest iteration. The convergence parameter is calculated from (56) and (57).

The proof of convergence of the L-shape method can be found in [15]. In the GADOP optimization problem, we observe that the feedback constraints are based on the probabilities of scenarios, the penalty, the routing cost, and the repair cost. As we observe from the GADOP problem, these feedback parameters can be calculated directly from parameters, i.e., the probabilities of scenarios ($\mathbb{P}(\omega), \mathbb{P}(\lambda)$), the penalty (p), the routing cost ($\widehat{c}_{i,d}^{(r)}$), and the repair cost (m). Since the feedback parameters do not depend on the solution at an iteration k , the feedback terms are constants. According to Algorithm 1, we observe that the first-stage solutions at $k = 1$ and $k = 2$ are identical because the feedback terms are the same. Therefore, the convergence of the decomposed GADOP is guaranteed at $k = 2$.

$$B = \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \widehat{X}_{i,d}^* (E_{i,d}^{(p)} + E_{i,d}^{(r)}) + M_d \widehat{W}_d^*, \quad (56)$$

$$M_d = \max(E_{i,d}^{(m)}), \quad \forall i \in \mathcal{D} \quad (57)$$

VI. EVALUATION

A. Parameter Setting

We implement the GADOP framework as a three-stage integer programming by using GAMS [19]. The experiments presented in this section are solved by the CPLEX solver. We use CPLEX default setting which comes with the heuristic option. Unless otherwise stated, we consider three drones and one truck in the GADOP framework. All the three drones are of the same type, the initial costs of which are $\widehat{c}_1^{(i)}, \widehat{c}_2^{(i)}, \widehat{c}_3^{(i)} = \$\$100$ and the capacity limit of which are $\widehat{f}_1, \widehat{f}_2, \widehat{f}_3 = 2$ kilograms. The daily flying distance limit and the flying distance limit per trip are $\widehat{l}_d = 150$ kilometers and $\widehat{e}_d = 15$ kilometers for each drone, respectively. The truck is a panel van, the initial cost of which is $\widehat{c}_1^{(i)} = \$\280 and the capacity limit of which is $\widehat{f}_1 = 1,060$ kilograms. The daily distance traveling limit and daily traveling time limit are $\widehat{l}_t = 200$ kilometers and $\widehat{h}_t = 8$ hours, respectively. The driving speed is set at $q_{i',j'} = 50$ kilometers per hour. The initial costs of the drones and the truck are synthesized based on Singapore shipping services [20]. The traveling cost of the truck is approximated to $\widehat{c}_{i',j'}^{(r)} = k_{i',j'} \times 1.05 \times 0.1$ based on Singapore government statistics [21], where 1.05 is the approximate fuel price factor and 0.1 is the average fuel consumption factor. The cost of a drone traveling round-trip between location i and the depot is $\widehat{c}_i^{(r)} = 0.005 \times (k_{i,0} + k_{0,i})$, where 0 is the index of the depot.

Furthermore, all customers have the same package weight $g_i = 1$ kilogram. The serving time of the truck is $\widehat{r} = 15$ minutes while the serving time of the drone is negligible because the drone just only drops a package off. The penalty and repair cost are $p = \$\20 and $m = \$\50 , respectively. The penalty is set based on the cost of outsourcing package delivery to the Speedpost service [22] by Singpost carrier plus additional cost for collecting the package or giving the compensation to a customer. Note that all customers have no specific time windows due to the space limit.

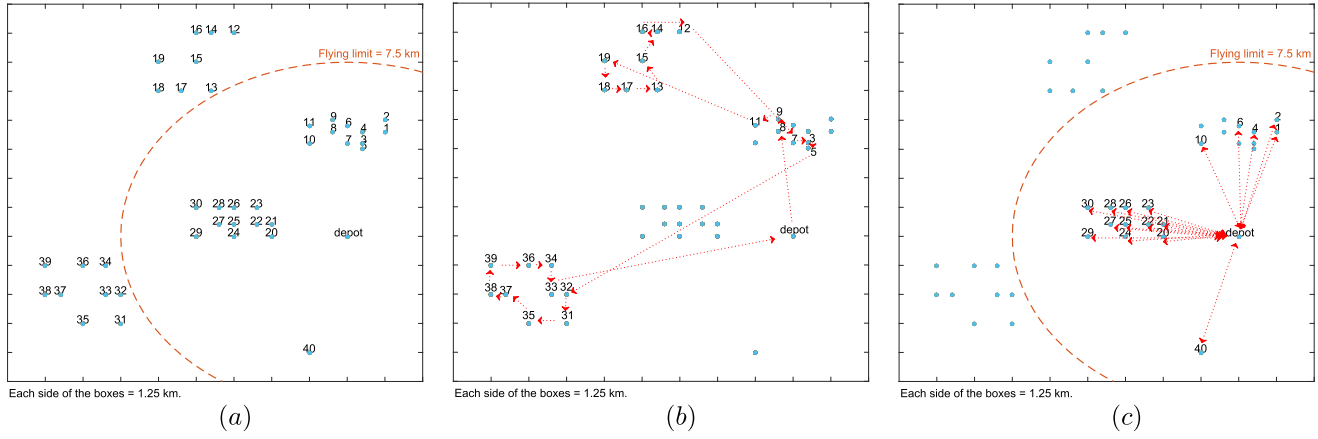


Fig. 5. Solomon benchmark example with the scenario that the drones are always able to takeoff and no breakdown occurs, where (a) presents customers' locations, (b) presents routing path of the truck, and (c) presents the customers served by the drone.

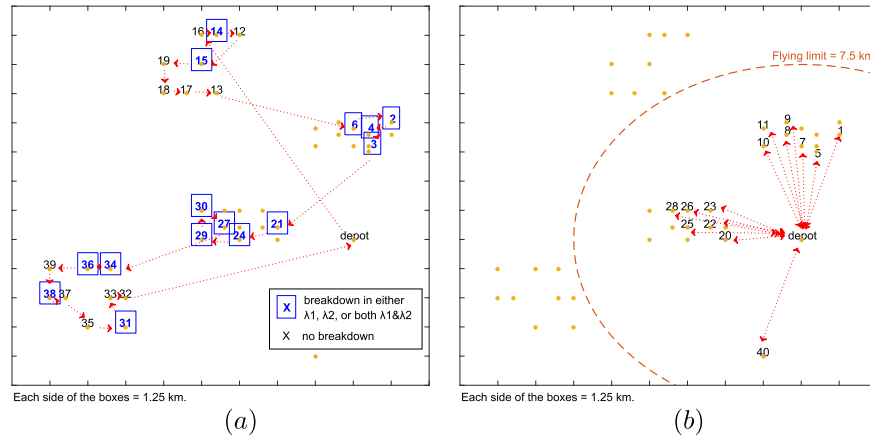


Fig. 6. Solomon benchmark example with two takeoff and breakdown scenarios, where (a) presents the routing path of the truck, and (b) presents the customers served by the drone.

Nevertheless, the experiment associated with time windows can be found in [17].

We evaluate the GADOP framework using two geographically data sets, i.e., Solomon Benchmark suite C101 [23] and a real trace from one of the Singapore logistics companies. Since Singapore has many high buildings, i.e., residential buildings, we use Singapore road network as the drone flying paths. The distances in kilometers between one location to another are obtained from the Google Map [24]. Solomon Benchmark suite is used in all experiments except the experiment in Section VI-C.

B. Tradeoff Between Taking and Not-Taking Uncertainty Into Account

We evaluate the GADOP framework with 40 customers. The Solomon benchmark suite is tested under (i) the deterministic case where the drones are always able to takeoff and no breakdown occurs and (ii) the random case where the drones may not be able to take off and breakdown may occur.

In the deterministic case, the scenarios are fixed as $\Omega = \{\omega_1\}$, $\Lambda = \{\lambda_1\}$, and all $\mathbb{R}_d(\omega_1) = 0$, $\mathbb{B}_{i,d}(\lambda_1) = 0$.

The customer locations and the drone flying distance limit are presented in Figure 5(a). The solution of the optimization in the deterministic case is to serve 17 and 23 customers by one drone and one truck, respectively. The routing path of the truck is presented in Figure 5(b), and the customers that are served by the drone are presented in Figure 5(c). The total payment is S\$386.29, which includes the initial cost of the truck S\$280, the initial cost of the drone S\$100, the routing cost of the truck S\$5.59, and the routing cost of the drone S\$0.70. The total traveling distances of the drone and truck are 140.27 and 53.26 kilometers, respectively.

Furthermore, in the random case, there are two scenarios for each of takeoff condition and breakdown condition ($\Omega = \{\omega_1, \omega_2\}$, $\Lambda = \{\lambda_1, \lambda_2\}$). The probabilities of two takeoff scenarios are $\mathbb{P}(\omega_1) = 0.9$ and $\mathbb{P}(\omega_2) = 0.1$, where all drones can takeoff in scenario ω_1 and cannot takeoff in scenario ω_2 . The probabilities of two random breakdown scenarios are also set to $\mathbb{P}(\lambda_1) = 0.9$ and $\mathbb{P}(\lambda_2) = 0.1$. The solution of the optimization in the random case is to serve 14 and 26 customers by one drone and one truck, respectively. The solution is presented in Figure 6. The total payment is S\$414.25 including the initial cost of the truck S\$280, the initial cost

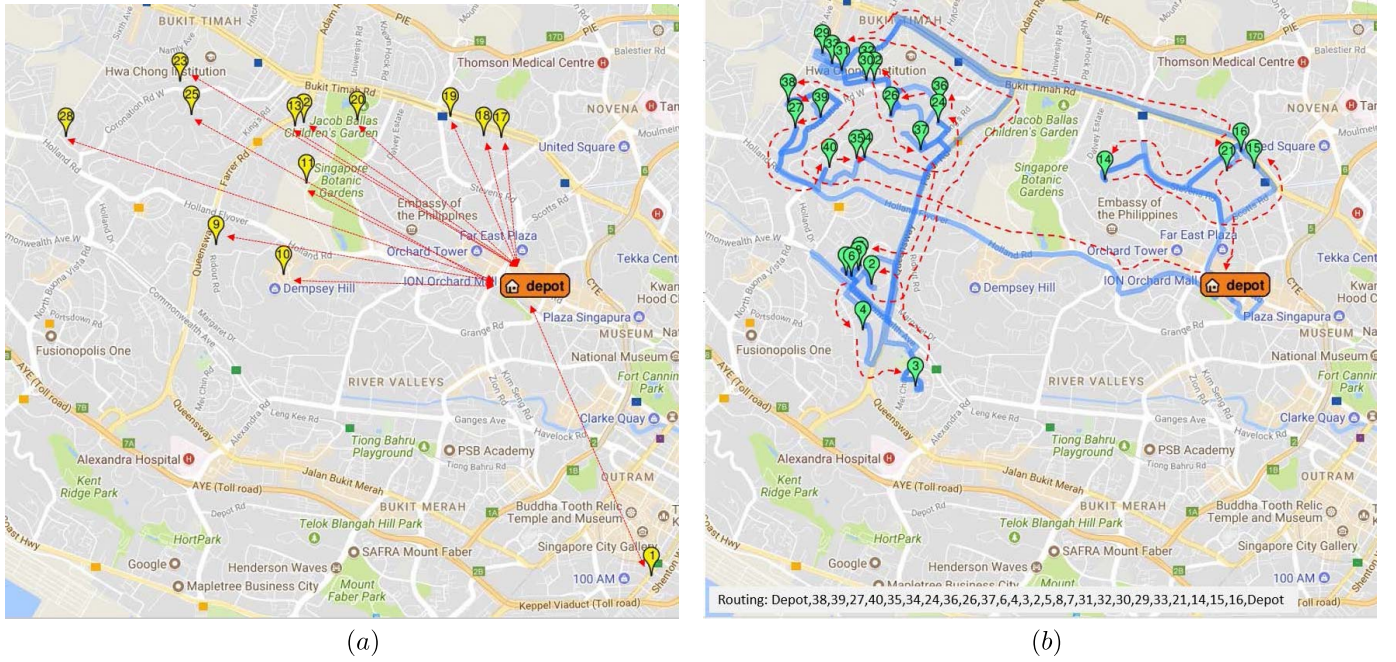


Fig. 7. The real package delivery problem in Singapore. (a) and (b) present the customers who are served by the drone and the truck, respectively.

of the drone S\$100, the routing cost of the truck S\$5.736, the routing cost of the drone S\$0.51, and the penalty incurred from the takeoff scenarios S\$28. The total traveling distances of the drone and the truck are 114.44 and 54.63 kilometers, respectively. Obviously, only drone delivery cannot serve all customers because some customers are outside the flying area. We observe that the solution assigns the drone to serve as many customers as possible in the deterministic case because the traveling cost of the drone is cheaper. Note that daily traveling distance limit of the drone is 150 kilometer. On the contrary, in the random case, since the drone may not be able to take off or the breakdown can occur, the solution avoids the penalty and repair cost by assigning more number of customers to the truck.

C. Routing, Payment, and Explanation of the Real Dataset

We next test the GADOP framework with the real data set from one of the Singapore logistics companies. In this experiment, we use the same takeoff and breakdown scenarios as the random case in Section VI-B, i.e., the two takeoff scenarios and two breakdown scenarios.

The solution of the optimization is presented in Figure 7, which is to use one drone to serve 13 customers and one truck to serve 27 customers. The total payment is S\$411.76 including the initial cost of the truck S\$280, the initial of the drone S\$100, the routing cost of the truck S\$5.17, the routing cost of the drone S\$0.58, and the penalty incurred from the takeoff scenarios S\$26. The total traveling distances of the drone and the truck are 129.20 and 49.31 kilometers, respectively. We observe that the truck is used for the customers which are clustered in the same area or located along the route. Furthermore, the drone tends to be used for the customer far

away from each other such as customer 1 at the bottom right corner of the map (Figure 7).

D. Impacts of the Number of Customers and Drone Flying Limit

We vary the number of customers and the drone flying limit per trip. In this experiment, we test the GADOP framework with one scenario which all drones can take off and no breakdown occurs, i.e., the deterministic case. The results and the locations of the customers are presented in Figure 8. If the drone flying limit per trip is large, i.e., 12.5 kilometers, all the customers are assigned to a single or multiple drones without trucks. However, when the number of the customers is large, i.e., 40, the truck and drone are used together. The reason is that three drones cannot serve all 40 customers in one day. If the drone flying limit per trip is small, i.e., 5 kilometers, the truck is selected to serve all the customers when the number of the customers is less than 25. The reason is that the drone alone cannot serve the customers farther than its flying distance limit while the truck is able to serve all the customers. When the number of the customers is larger than or equal to 30, the truck alone cannot serve all the customers because of its daily traveling time limit, and thus both the truck and drone are used together when the drone flying limit per trip is less than or equal to 10 kilometers.

We conduct the performance comparison between (i) stand-alone ground-based delivery in which only ground-based trucks are used to serve customers, and (ii) joint truck and drone delivery services. As presented in Figure 9, the stand-alone ground-based delivery is always the most expensive delivery mode. This result clearly shows the effectiveness of the proposed joint truck and drone delivery.

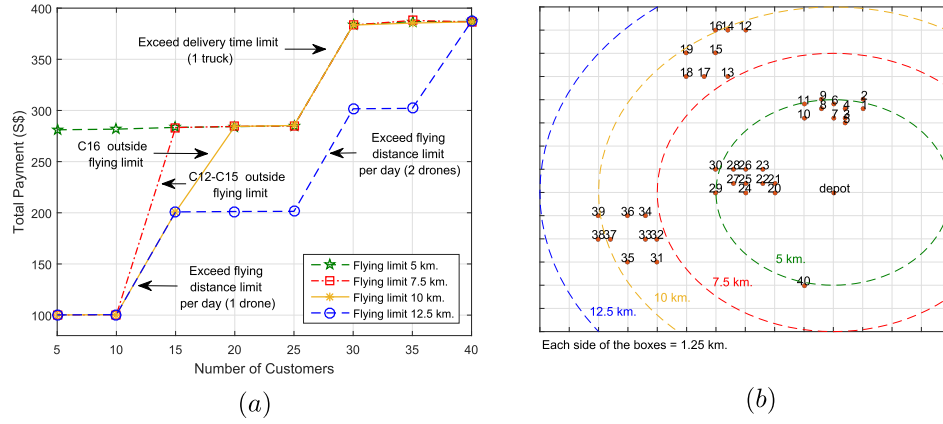


Fig. 8. Increasing number of customers, where (a) presents the total payment and (b) presents customer locations and drone flying limits per trip.

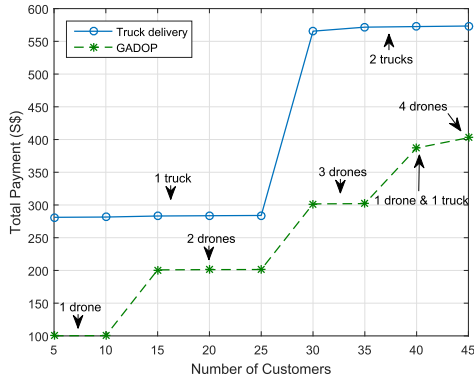


Fig. 9. Performance comparison between truck delivery and the proposed GADOP framework.

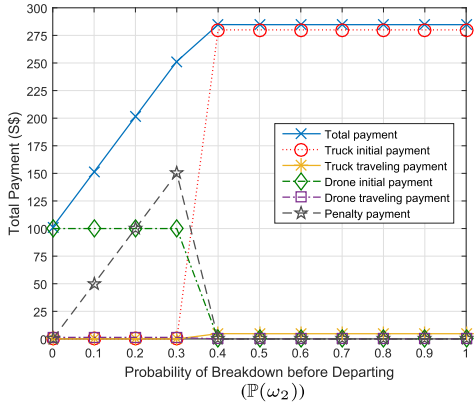


Fig. 10. Increasing the probability of that all drones cannot take off.

E. Impact of Random Takeoff Scenarios

We evaluate the GADOP framework with 25 customers, two takeoff scenarios, and one breakdown scenario, i.e., $\Lambda = \{\lambda_1\}$, and all the drones do not break down, i.e., $\mathbb{B}_{i,d}(\lambda_1) = 0$. The two takeoff scenarios are (i) all the drones can take off (ω_1) and (ii) all the drones cannot take off (ω_2). We vary the probability of that all the drones cannot take off, i.e., $\mathbb{P}(\omega_2)$, and $\mathbb{P}(\omega_1) = 1 - \mathbb{P}(\omega_2)$. We ignore the flying distance limit in this case for the ease of presenting the impact of the takeoff scenarios. The results are presented in Figure 10. The drone

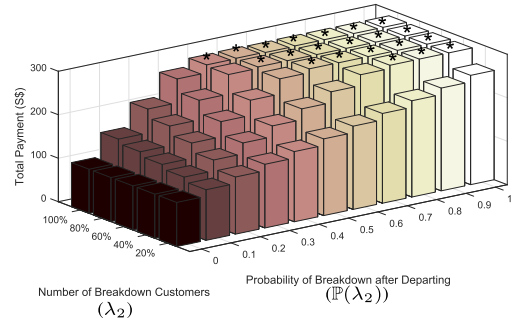


Fig. 11. The results of varying the number of breakdown customers and the probability of the breakdown scenarios, where the bars with an asterisk symbol on the top represents the results that all customers are served by the truck.

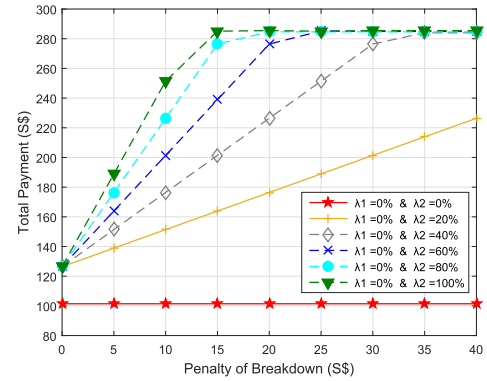


Fig. 12. Effects of increasing the penalty.

is used to serve all the customers when the probability is less than or equal to 0.3. Otherwise, the truck is used to serve all the customers.

F. Impact of Random Breakdown Scenarios

Next we consider 25 customers, one takeoff scenario $\Omega = \{\omega_1\}$, i.e., all the drones can take off $\mathbb{R}_d(\omega_1) = 0$, and two breakdown scenarios, i.e., $\Lambda = \{\lambda_1, \lambda_2\}$. For the two breakdown scenarios, (i) there is no breakdown, i.e., all $\mathbb{B}_{i,d}(\lambda_1) = 0$, and there is breakdown for some customers. For example, breakdown percentage equal to 20% means

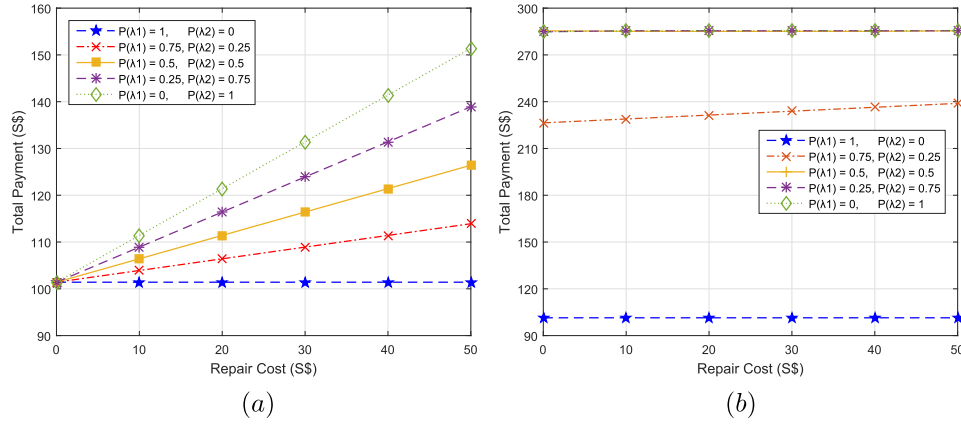


Fig. 13. The drone repair cost is varied for (a) the penalty is $p = 0$ and (b) the penalty is $p = 20$.

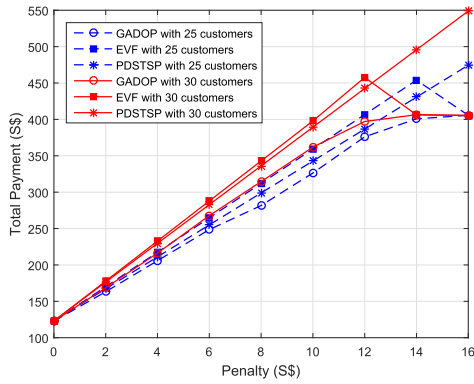


Fig. 14. The comparison between the solutions from the GADOP, the EVF, and the PDSTSP.

5 customers have breakdown and 20 customers have no breakdown, which can be referred to $\mathbb{B}_{i,d}(\lambda_2) = 1$ for $i = 1, \dots, 5$, and $\mathbb{B}_{i,d}(\lambda_2) = 0$ for $i = 6, \dots, 25$. We vary the probability of the breakdown scenario $\mathbb{P}(\lambda_2)$ and the percentage of the breakdown customers. Note that we again ignore the flying distance limit. The results are presented in Figure 11. From the results, the GADOP framework always selects the drone to serve the customers when the percentage of the breakdown customers is less than or equal to 20% or the probability of the breakdown scenario $\mathbb{P}(\lambda_2) \leq 0.2$.

G. Impact of Penalty

We use the same setting and scenarios as in Section VI-F, but the probabilities of the breakdown scenarios are equal, i.e., $\mathbb{P}(\lambda_1) = \mathbb{P}(\lambda_2) = 0.5$. The results are presented in Figure 12. When the breakdown penalty cost is high, i.e., $p \geq 15$, the truck is used to serve all the customers. On the contrary, the drone is used when the penalty cost is $p < 15$. The same result is also observed when the percentage of breakdown customers is 80%, 60%, and 40% when $p \geq 20, 25$, and 35, respectively. Apparently, the number of breakdown customers has a significant impact on the solution.

H. Impact of Repair Cost

Next, we consider two specific scenarios that (i) there is no breakdown and (ii) there is breakdown to all customers.

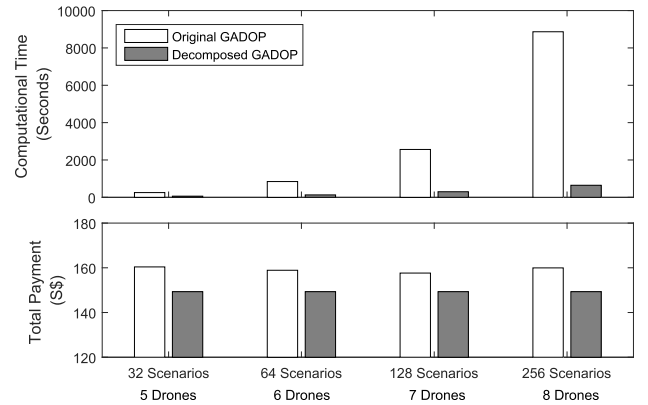


Fig. 15. The comparison between the original GADOP and the decomposition. The number of takeoff scenarios and the number of drones are varied.

We vary the drone repair cost and the probability of these two scenarios. Figure 13 shows the total payments, where (a) the penalty is ignored $p = 0$ and (b) the penalty is $p = 20$. When the probability that the breakdown occurs to all customers, i.e., $\mathbb{P}(\lambda_2) = 0$, the repair cost is not taken into account. As shown in Figure 13(a), the drone is used to serve all the customers. On the other hand, Figure 13(b) shows that the drone is selected when the probability that the breakdown occurs to all customers is $\mathbb{P}(\lambda_2) \leq 0.25$. Otherwise, the truck is used. We observe that the impact of the drone repair cost is significantly less than the penalty p and the probability of that the breakdown occurs to all customers.

I. Simulation

We conduct a simulation and compare the performance in terms of the total payments obtained from the solutions of the GADOP optimization, expected value formulation (EVF), the parallel drone scheduling traveling salesman problem (PDSTSP) [1]. We adapt the PDSTSP from minimizing the delivery time to minimizing total payments. The EVF and adapted PSDTSP formulations can be found in the Appendix.

Here, we set the initial cost of a truck to \$400 and the daily traveling time limit is 10 hours. For the EVF, all the decisions are optimized only for the first stage. For the PDSTSP, the uncertainties are not considered. Clearly, the total

payment obtained from solving the GADOP optimization is significantly lower than that of the EVF and the PSDTSP. The reason is that the solution of the GADOP optimization takes into account the takeoff and breakdown conditions explicitly which lower the chance of incurring penalty, i.e., by using trucks for more customers. On the contrary, the solution of the EVF considers only the probability of unsuccessful delivery and implicitly ignores the effect of the penalty, even though the penalty is considered. Note that when the penalty is large, e.g., \$14 and \$16 for the cases with 30 and 25 customers, respectively, the trucks are used for all customers which is the same for both GADOP optimization and EVF. However, the PSDTSP decisions are not influenced by the penalty. Therefore, the total payments of GADOP and the EVF are identical, and the total payment of PSDTSP is the highest.

J. Decomposition

We study the decomposition of the GADOP optimization. The penalty, repair cost, and the drone initial cost are $p = 5$, $m = 5$, and $\hat{c}_d^{(i)} = 10$ respectively. In this experiment, we consider one truck and 10 breakdown scenarios ($\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{10}\}$). The number of drones and the number of takeoff scenarios are varied. The takeoff scenarios include all possibilities of takeoff events that can occur to all the drones. To illustrate, five drones are experimented with 32 takeoff scenarios ($\Omega = \{\omega_1, \omega_2, \dots, \omega_{32}\}$), six drones are experimented with 64 takeoff scenarios, and so on.

The comparison is shown in Figure 15. When the number of takeoff scenarios grows, the difference between the computational time of the original and decomposed problems becomes larger. As shown in Figure 15, the decomposed GADOP achieves 407%, 648%, 854%, and 1372% speedup with 32, 64, 128, and 256 scenarios, respectively. Meanwhile, the decomposed GADOP always achieves lower total payment. Note that, in some cases, the CPLEX solver cannot give an optimal solution for the original problem when the problem is too large. Thus, the CPLEX solver may return a suboptimal solution instead of the optimal solution. In this regard, the suboptimal solution achieves much higher payment than that of the decomposition while the computational time of the decomposition may not significantly improve [25]. In summary, the decomposition enables the GADOP to handle larger number of scenarios.

VII. CONCLUSION AND FUTURE WORKS

We have proposed the joint ground and aerial delivery service optimization and planning (GADOP) framework. The **uncertainty** of drone delivery has been taken into account by considering takeoff and breakdown conditions. The aim of the GADOP framework is to help a supplier to make the decision either to use ground-based trucks, drones, or the combination to serve customers. We have formulated the GADOP optimization as a three-stage stochastic integer programming model. The performance evaluation of both Solomon benchmark suite and the Singapore real road network has been conducted. The results clearly show that using drones in the package delivery can reduce the total payment of the supplier significantly. Moreover, simulation results have shown that the GADOP

framework always achieves the lower total payment than that of the expected value formulation (EVF) and the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP). We have adopted the **L-shape decomposition method** to enhance the solvability of the optimization problem.

For future work, the **uncertainty in customers' demand** and **traveling time** will be considered and, we will also incorporate multiple-stage scenarios of the breakdown condition, which depend on each customer. Moreover, different types of subtour elimination, which aim to **speedup computational time**, will be studied.

APPENDIX

A. Expected Value Formulation (EVF)

The EVF objective function is similar to the original objective function in (2), but only the formulation of the $\mathbb{E}(\mathcal{L}(\hat{X}_{i,d}))$ is different. The equation is indicated in (58), where $\mathbb{P}(d)$, $\mathbb{P}(i)$, and $\mathbb{M}(d)$ denote the probability that drone d cannot take off, the probability that breakdown occurs while serving customer i , and the probability that drone d needs to be repaired, respectively. The EVF subjects to the constraints in (5)-(19) and (24)-(37).

$$\begin{aligned} \mathbb{E}(\mathcal{L}(\hat{X}_{i,d})) = & \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d} \mathbb{P}(d) + \sum_{d \in \mathcal{D}} p \hat{X}_{i,d} (1 - \mathbb{P}(d)) \\ & + \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} p \hat{X}_{i,d} (1 - \mathbb{P}(d)) \mathbb{P}(i) \\ & + \sum_{d \in \mathcal{D}} m (1 - \mathbb{P}(d)) \mathbb{M}(d) \end{aligned} \quad (58)$$

B. Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP)

The PDSTSP can handle multiple drones with only one truck. It does not take the serving time constraint and the constraints of traveling limits, i.e., time and distance, into account. Furthermore, the the uncertainties i.e., takeoff and breakdown conditions, are ignored in the PDSTSP. We adapt the PDSTSP objective function, which is proposed in [1], from minimizing the delivery time to minimizing the total payment. The new objective function is presented in (59). The two delivery time constraints in [1] are replaced by the constraints in (60) and (61). The constraint in (60) ensures that the initial cost of the truck is paid when the truck is used. Similarly, the constraint in (61) ensures that the initial cost of drones are paid when the drones are used. The rest of the constraints in [1] are kept as the original.

$$\begin{aligned} \text{Minimize: } & \bar{c}^{(i)} \bar{W} + \sum_{d \in \mathcal{D}} \hat{c}_d^{(i)} \hat{W}_d + \sum_{\substack{i', j' \in \\ \mathcal{C} \cup \{0\}}} \bar{c}_{i', j'}^{(r)} V_{i', j'} \\ & + \sum_{i \in \mathcal{C}} \sum_{d \in \mathcal{D}} \hat{c}_i^{(r)} \hat{X}_{i,d} \end{aligned} \quad (59)$$

$$\sum_{i \in \mathcal{C}} V_i \leq \Delta \bar{W}, \quad \forall i \in \mathcal{C} \quad (60)$$

$$\sum_{i \in \mathcal{C}} \hat{X}_{i,d} \leq \Delta \hat{W}_d, \quad \forall d \in \mathcal{D} \quad (61)$$

REFERENCES

- [1] 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.
- [2] Y. Marinakis and A. Migdalas, "Annotated bibliography in vehicle routing," *Oper. Res.*, vol. 7, pp. 27–46, Jan. 2007.
- [3] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, "Survey of green vehicle routing problem: Past and future trends," *Expert Syst. Appl.*, vol. 41, pp. 1118–1138, Mar. 2014.
- [4] M. M. Solomon, "Time window constrained routing and scheduling problems," *Transp. Sci.*, vol. 22, no. 1, pp. 1–13, 1988.
- [5] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: A classification scheme and survey," *Top.*, vol. 15, pp. 1–31, Jul. 2007.
- [6] V. Pillac, M. Gendreau, C. Gu  ret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, Feb. 2013.
- [7] M. Gendreau, G. Laporte, and R. S  guin, "Stochastic vehicle routing," *Eur. J. Oper. Res.*, vol. 88, no. 1, pp. 3–12, 1996.
- [8] G. Laporte, "The vehicle routing problem: An overview of exact and approximate algorithms," *Eur. J. Oper. Res.*, vol. 59, no. 3, pp. 345–358, 1992.
- [9] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *Int. Trans. Oper. Res.*, vol. 7, nos. 4–5, pp. 285–300, 2000.
- [10] C. Sabo, D. Kingston, and K. Cohen, "A formulation and heuristic approach to task allocation and routing of UAVs under limited communication," *Unmanned Syst.*, vol. 2, pp. 1–17, Dec. 2013.
- [11] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 287–294, Jan. 2014.
- [12] L. Evers, T. Dollevoet, and A. I. Barros, "Robust UAV mission planning," *Ann. Oper. Res.*, vol. 222, pp. 293–315, Nov. 2014.
- [13] L. Evers, A. I. Barros, H. Monsuur, and A. Wagelmans, "Online stochastic UAV mission planning with time windows and time-sensitive targets," *Eur. J. Oper. Res.*, vol. 238, pp. 348–362, Oct. 2014.
- [14] 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.
- [15] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. New York, NY, USA: Springer, 2011.
- [16] L. A. Haidari *et al.*, "The economic and operational value of using drones to transport vaccines," *Vaccine*, vol. 34, pp. 4062–4067, Jul. 2016.
- [17] S. Sawadsitang, R. Kaewpuang, S. Jiang, D. Niyato, and P. Wang, "Optimal stochastic delivery planning in full-truckload and less-than-truckload delivery," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Sydney, NSW, Australia, Jun. 2017, pp. 1–5.
- [18] F. Schenkelberg, "How reliable does a delivery drone have to be," in *Proc. Annu. Rel. Maintainab. Symp. (RAMS)*, Tucson, AZ, USA, Jan. 2016, pp. 1–5.
- [19] D. Chattopadhyay, "Application of general algebraic modeling system to power system optimization," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 15–22, Feb. 1999.
- [20] *Lorry Dimensions, Lorry Capacity and Sizes*. Accessed: Sep. 5, 2018. [Online]. Available: <https://rentallorrysingapore.com/lorry-dimensions/>
- [21] Land Transportation Authority of Singapore. (2009). *Singapore Land Transportation Statistic in Brief*. [Online]. Available: https://www.lta.gov.sg/content/dam/ltaweb/corp/PublicationsResearch/files/FactsandFigures/Statistics_in_Brief_2009.pdf
- [22] *Home, Singapore Post*. Accessed: Sep. 5, 2018. [Online]. Available: <http://www.singpost.com>
- [23] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.
- [24] *Get Started, Distance Matrix API, Google Developers*. Accessed: Sep. 5, 2018. [Online]. Available: <https://developers.google.com/maps/documentation/distance-matrix/>
- [25] S. Sawadsitang, S. Jiang, D. Niyato, and P. Wang, "Optimal stochastic package delivery planning with deadline: A cardinality minimization in routing," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC Fall)*, Toronto, ON, Canada, Sep. 2017, pp. 1–7.



Sutinee Sawadsitang received the B.Eng. degree in computer engineering from King Mongkut's University of Technology Thonburi, Thailand, in 2012, and the M.Eng. degree from Shanghai Jiao Tong University, China, in 2015. She is currently pursuing the Ph.D. degree with SIMTech-NTU Joint Laboratory on Complex Systems, Nanyang Technological University, Singapore. Her research interests are in the areas of transportation systems, operations research, optimization, and high-performance computing.



Dusit Niyato (M'09–SM'15–F'17) received the B.Eng. degree from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from University of Manitoba, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the areas of energy harvesting for wireless communication, Internet of Things, and sensor networks.



Puay-Siew Tan received the Ph.D. degree in computer science from the School of Computer Engineering, Nanyang Technological University, Singapore. She is currently an Adjunct Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, and also the Co-Director of the Singapore Institute of Manufacturing Technology (SIMTech), Nanyang Technological University Joint Laboratory on Complex Systems. She is also currently Full-Time Staff with SIMTech, where she leads the Manufacturing Control Tower program. She is also the Deputy Director of the Manufacturing Productivity Technology Centre. Her research interests are in the cross-field disciplines of computer science and operations research for virtual enterprise collaboration, in particular sustainable complex manufacturing and supply chain operations in the era of Industry 4.0.



Ping Wang (M'08–SM'15) received the Ph.D. degree in electrical engineering from University of Waterloo, Canada, in 2008. She was with Nanyang Technological University, Singapore. She is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, York University, Canada. Her current research interests include resource allocation in multimedia wireless networks, cloud computing, and smart grid. She was a co-recipient of the Best Paper Award from the IEEE Wireless Communications and Networking Conference 2012 and the IEEE International Conference on Communications 2007. She served as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, EURASIP Journal on Wireless Communications and Networking, and International Journal of Ultra Wideband Communications and Systems.