

University of Wollongong

Research Online

---

Faculty of Engineering and Information  
Sciences - Papers: Part B

Faculty of Engineering and Information  
Sciences

---

2020

## Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles

Hailong Huang

Andrey V. Savkin

Chao Huang

University of Wollongong, ch449@uowmail.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers1>



Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

---

### Recommended Citation

Huang, Hailong; Savkin, Andrey V.; and Huang, Chao, "Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles" (2020). *Faculty of Engineering and Information Sciences - Papers: Part B*. 3927.

<https://ro.uow.edu.au/eispapers1/3927>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

---

# Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles

## Abstract

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This paper proposes a novel parcel delivery system which consists of a drone and public transportation vehicles such as trains, trams, etc. This system involves two delivery schemes: drone-direct scheme referring to delivering to a customer by a drone directly and drone-vehicle collaborating scheme referring to delivering a customer based on the collaboration of a drone and public transportation vehicles. The fundamental characteristics including the delivery time, energy consumption and battery recharging are modelled, based on which a time-dependent scheduling problem for a single drone is formulated. It is shown to be NP-complete and a dynamic programming-based exact algorithm is presented. Since its computational complexity is exponential with respect to the number of customers, a sub-optimal algorithm is further developed. This algorithm accounts the time for delivery and recharging, and it first schedules the customer which leads to the earliest return. Its computational complexity is also discussed. Moreover, extensive computer simulations are conducted to demonstrate the scheduling performance of the proposed algorithms and the impacts of several key system parameters are investigated.

## Disciplines




Engineering | Science and Technology Studies

## Publication Details

H. Huang, A. Savkin & C. Huang, "Scheduling of a parcel delivery system consisting of an aerial drone interacting with public transportation vehicles," *Sensors* (Switzerland), vol. 20, (7) 2020.

Article

# Scheduling of a Parcel Delivery System Consisting of an Aerial Drone Interacting with Public Transportation Vehicles

Hailong Huang <sup>1</sup>, Andrey V. Savkin <sup>1,\*</sup> and Chao Huang <sup>2</sup>

<sup>1</sup> School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney 2052, Australia; hailong.huang@unsw.edu.au

<sup>2</sup> Electrical Computer and Telecommunications Engineering, University of Wollongong, Wollongong 2500, Australia; ch449@uowmail.edu.au

\* Correspondence: a.savkin@unsw.edu.au

Received: 3 March 2020; Accepted: 2 April 2020; Published: 5 April 2020



**Abstract:** This paper proposes a novel parcel delivery system which consists of a drone and public transportation vehicles such as trains, trams, etc. This system involves two delivery schemes: drone-direct scheme referring to delivering to a customer by a drone directly and drone-vehicle collaborating scheme referring to delivering a customer based on the collaboration of a drone and public transportation vehicles. The fundamental characteristics including the delivery time, energy consumption and battery recharging are modelled, based on which a time-dependent scheduling problem for a single drone is formulated. It is shown to be NP-complete and a dynamic programming-based exact algorithm is presented. Since its computational complexity is exponential with respect to the number of customers, a sub-optimal algorithm is further developed. This algorithm accounts the time for delivery and recharging, and it first schedules the customer which leads to the earliest return. Its computational complexity is also discussed. Moreover, extensive computer simulations are conducted to demonstrate the scheduling performance of the proposed algorithms and the impacts of several key system parameters are investigated.

**Keywords:** UAVs; drones; parcel delivery; scheduling problem; public transportation vehicles

## 1. Introduction

With the fast development of drones (also called unmanned aerial vehicles (UAVs)) technology, drones have found various applications in civilian domains [1], such as wireless communication support [2], structural health inspection [3], farming [4], surveillance and monitoring [5–7] and parcel delivery [8,9]. Thanks to their mobility and flexibility, many logistics companies and many advanced control approaches such as autonomous landing [10], such as Amazon, Alibaba, DHL, SF Express, etc., have started to pay more attention to the application of drones in parcel delivery. Researchers and logistics companies have conducted much research on designing cost-and-time efficient systems such that the delivery can be done in a fast and low-cost way. One model, which has attracted much attention in the research community, is called drone-truck collaboration [11]. Specifically, a truck is equipped with one or several drones. The drones can launch from the truck, deliver to some customers, and then fly to somewhere to dock with the truck. Meanwhile, the truck can service some other customers. Some variants of this model have also been published, such as clustering TSP [12], TSP with drone (TSP-D) [13], minimum cost TSP [14], TSP with Drone Station (TSP-DS) [15], Heterogeneous Delivery Problem (HDP) [16,17], Drone Scheduling Problem (DSP) [18], and Vehicle Routing Problem with Drones (VRPD) [19].

Another model is to equip a depot with a fleet of drones, and the drones fly between customers and the depot [20]. An important problem is the delivery scheduling problem. A number of publications present some theoretical results for this model [21,22]. Comparing to the first model, this model can further reduce the human labour participation. However, the currently available commercial drones are often constrained in flight distance due to the limited on-board battery capacity. For example, Amazon released that their commercial delivery drone can fly about half an hour and travel up to 32 km (<https://www.cbsnews.com/news/amazon-unveils-futuristic-plan-delivery-by-drone/>). If a customer is located slightly outside the delivery area, they cannot be served by the drones. Therefore, the delivery area of this model is also limited.

Following the second model, some publications have considered the facility location problem under the context of drone delivery to extend the delivery area [23,24]. Clearly, one disadvantage of this method is the investment in constructing the new facilities. The current paper also focuses on addressing this shortcoming. Unlike [23,24] to place more facilities, we introduce the idea of drone–vehicle collaboration [25,26]. Here, the vehicle refers to the public transportation vehicles. Unlike the trucks in the first model, the public transportation vehicles have their own routes and timetables, and they cannot be controlled like the trucks by the logistics companies. The public transportation vehicles can take drones to some places which are unreachable by flying themselves. More importantly, as public transportation vehicles are natural mobile platforms which have already existed in the urban area, this model does not introduce much extra cost.

We propose a novel parcel delivery system involving two delivery schemes. The first one, hereafter called drone-direct scheme, uses a drone to directly fly to the customer and then return to the depot. The second one, hereafter called drone–vehicle scheme, makes use of public transportation vehicles. A drone can travel with a vehicle to some position near the customer like a passenger. Differently, it rests on the roof of the vehicle. Then, the drone leaves the vehicle and flies to the customer. After dropping the parcel, it then flies to another vehicle that can transport it back to the depot. In the two schemes, the drone operates autonomously. Comparing these two schemes, the drone-direct scheme generally achieves fast delivery than the drone–vehicle scheme. However, the former may also consume more energy than the latter, since all the trip is made by a drone itself. In contrast, the drone–vehicle scheme explores the vehicle mobility, which saves the energy for the drone. Thus, after delivery, a drone that follows the drone-direct scheme may need longer time for recharging the battery than the drone–vehicle scheme.

Although the delivery time may be longer as the drone needs to wait for the transportation of the public vehicles, the second scheme creates the possibility of delivering to a customer in a low-cost manner. It is quite useful for a supplier that wishes to cost-effectively deliver to a non-urgent customer. One challenge brought by the drone–vehicle scheme is the collaboration between the drone and the vehicles. With the development of vehicle-to-vehicle communication technology, we believe that in the future a drone can also talk to a normal vehicle to collect the trip information of the vehicle. Then, there will be more options for the drone to travel. In this paper, we limit ourselves to a situation where the vehicle-to-vehicle communication technology is not used, and the drone only travels with public vehicles, whose trip information can be well managed.

We study the fundamental characteristics of the two schemes and also investigate how to select the delivery scheme. Based on these models, we furthermore study the scheduling task for a single drone given a set of customers such that the total delivery time is minimized. This problem has practical applications since many suppliers wish to service their customers in queue in the fastest manner. The time needed to recharge the battery is accounted in our model. In particular, to deliver to a certain customer, the residual energy of the battery has to be greater than that to be consumed. Otherwise, the drone has to recharge the battery before departure. In our formulation, the charging rate can be adjusted, and it covers the case where battery charging is replaced by battery swapping by simply setting the charging rate sufficiently large. With the energy constraint, a scheduling problem is formulated. The solution of the scheduling problem involves the permutation of the to-be-served

customers, the time for recharging, and the delivery scheme for each customer. Since the proposed delivery system involves two delivery schemes and if the drone–vehicle scheme is adopted for a certain customer, the trip time further depends on the departure instant. We mathematically formulate this time-dependent scheduling problem and show that it is NP-complete.

To find the optimal solution, we develop a dynamic programming-based algorithm. It gives the optimal schedule, but its computational complexity is exponential with respect to the number of customers, which is infeasible for a slightly higher number of customers. Considering this, an algorithm which finds a sub-optimal solution is developed. This algorithm accounts the time for delivery and recharging and it first schedules the customer which leads to the earliest return. It is shown that the sub-optimal algorithm is much more computationally efficient than the exact algorithm. The performance gap between them in terms of the total time to complete all the deliveries is demonstrated via extensive computer simulations. The simulation results display that the sub-optimal algorithm achieves close performance to the exact algorithm. Finally, we conduct more simulations to investigate the impacts of some important system parameters including the energy consumption rate, recharging rate, the timetable of the vehicles, and the speed of the drone.

The rest of the paper is organized as follows. Section 2 discusses some related publications. Section 3 presents the system models and states the studied problem. Section 4 presents the proposed solutions. In Section 5, extensive simulation results are provided and Section 6 concludes the paper with some discussions about future research directions.

## 2. Related Work

The current paper considers the drone delivery application and the problem under investigation is, in essence, a scheduling problem. The scheduling problem is a conventional problem, which in general consists of the allocation of resources over time to perform a set of jobs. The aim of scheduling is to figure out the optimal schedule, among all feasible schedules, which optimizes a given criterion function. In this paper, the drone is the resource, delivering a customer is a job, and the aim is to minimize the total time for delivery.

In terms of drone delivery, the first category of relevant publications is more or less related to the conventional travelling salesman problem (TSP). The papers [11,13,15–17,19] consider the scenario when a truck carries one or several drones and investigate how to schedule the deliveries for both the truck and the drones. A very basic idea is presented in [11] which firstly constructs a TSP tour for the truck with all the customers. Then, it considers whether assigning some customer to a drone reduces the total delivery time. The reference [12] presents a method which clusters the customers first and then constructs a TSP tour. The authors of [13] discuss a variant scenario of [11], where the drone also needs to follow the road network like trucks. This helps the drone to avoid collisions with buildings in urban environments, however, it loses the benefits of faster delivery by the drone. Instead of the total delivery time, the reference [14] aims at minimizing the cost associated with the deliveries, where the transportation cost of truck and drone and the part relating to the time a vehicle has to wait for the other are accounted. The paper [15] presents a model involving a drone station, which stores a fleet of drones and is activated by a truck when the truck loads parcels to the station. This paper investigates the problem of customer assignment with the objective to minimize the total delivery time. Different from the above models, in [16] the truck is used only to transport a drone and it does not deliver to customers. The authors model it as a generalized TSP and the solution includes not only the route of the truck but also where the drone launches and returns to the truck. The paper [17] considers a similar scenario to [16], while the authors further account the delivery order constraint (i.e., some customers should be served before some others). They show the NP-hardness of the problem and derive the lower bound of the optimal time to deliver to all the customers. Heuristic algorithms are also provided to address the problem. The paper [18] also considers the same scenario, while it assumes the route of the truck is fixed and discusses how to schedule multiple drones. The authors of [19] formulate the Vehicle Routing Problem with Drones problem (VRPD) as a mixed integer programming model and

propose a branch-and-price algorithm to address it. In particular, a drone is allowed to deliver more than one customers in one trip. Considering this, many routing methods can be applied if the drone can deliver multiple parcels in one trip with the presence of charging static or mobile stations [27,28].

The second category of related work is about the drone-direct scheme, where only drones are used for delivery. The paper [20] considers the scenario where a drone delivers more than one customers in a trip, although this is not the favoured mode by logistics companies. Clearly, if there is only one drone, the problem is a variant of TSP; and if there are multiple drones, the problem turns to be a variant of vehicle routing problem (VRP), which is also a variant of TSP. In [20], the energy constraint of drones is accounted, and the authors investigate the influence of the payload on the energy consumption. In [21], the authors analyze the system stability based on the queuing theory. The paper [22] investigates the impact of the unreliability of drones on the delivery performance due to inaccurate estimation on the battery consumption rate. Since the capacity of the on-board battery is often constrained, the delivery range by the drone-direct scheme is limited. The paper [23] proposes a model to deploy a certain number of charging stations such that the coverage of customers is maximized, which is a coverage problem. In this paper, a drone only delivers to one customer and the impact of the payload is also accounted. The reference [24] addresses the problem of positioning warehouses as well as refuel stations.

The last category is about the utilization of public transportation vehicles [25,26]. Unlike trucks in the first category, the public transportation vehicles are not controlled by logistics companies. The authors of [25] carried out interviews with experts from relevant industries and concluded that the utilization of autonomous mobility is a feasible alternative delivery method. The reference [26] presents an idea of using public transportation vehicles to move drones to some positions of interest and even recharge the drones.

There are some similarities and differences between the current paper and the aforementioned ones. The model proposed herein combines the drone-direct scheme and the drone-vehicle scheme. It involves the delivery scheme selection which depends on the location of a specific customer, the residual energy of the drone, and the timetable of the public transportation vehicles. This paper targets the scheduling problem and the time for recharging is accounted. The aforementioned references, especially those in the first category, generally do not consider this, whereas they assume that a sufficient number of batteries is available at the truck and the battery can be replaced with a fresh one after each delivery.

### 3. System Model and Problem Statement

Consider a parcel delivery system consisting of a depot, a drone and a fleet of public transportation vehicles. The vehicles keep running on a predefined route  $\mathcal{R}$  following a timetable  $\mathcal{T}$  (Notice that this paper only considers the public transportation vehicles, such as trains and trams, whose timetables are accurate. The impacts of the stochastics of unreliable (in terms of timetable) public transportation vehicles such as buses and environmental factors such as wind are left as our future work.). Along the route  $\mathcal{R}$ , there is a set of stops  $\mathcal{S}$ . There are two delivery schemes: drone-direct delivery and drone-vehicle delivery. In the drone-direct delivery scheme, a drone departs the depot and flies to the customer directly. After dropping the parcel, it returns to the depot. In the drone-vehicle delivery scheme, after departing the depot, a drone catches a vehicle and travels with the vehicle on the roof. The drone leaves the vehicle at some position near the customer. After finishing the delivery, the drone flies to the nearest vehicle stop  $s \in \mathcal{S}$  and waits for a vehicle that can transport it back to the depot. The current paper considers a scenario where the vehicle route is closed, see Figure 1. Notice that the discussed route can be extended to the complex public transportation network and as will be shown later, a delivery time calculator is needed to compute the delivery time between the depot and a customer in the complex public transportation network. The main notations and their meanings are listed in Table 1.

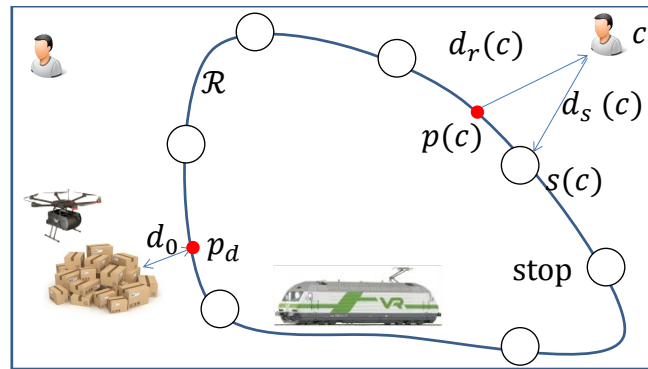


Figure 1. The considered parcel delivery system.

Table 1. Notations and descriptions.

Notation	Description
$\mathcal{R}$	Vehicle route
$\mathcal{T}$	Vehicle timetable
$\mathcal{S}$	A set of stops on $\mathcal{R}$
$v$	Drone speed
$p$	Energy consumption rate
$q$	Battery recharging rate
$E_0$	Battery capacity
$\epsilon$	Safety residual energy
$\epsilon_1$	Energy consumption for take-off
$\epsilon_2$	Energy consumption for landing
$\delta_1$	Time needed for take-off
$\delta_2$	Time needed for landing
$\tau_r(c)$	Charging time for delivering customer $c$
$\tau_t(c)$	Trip time for delivering customer $c$
$\tau_f(c)$	Flight time for delivering customer $c$
$t_d(c)$	Departure instant for delivering customer $c$
$t_r(c)$	Returning instant after delivering customer $c$
$d_d(c)$	Distance between customer $c$ and the depot
$p(c)$	The closest point on $\mathcal{R}$ to customer $c$
$d_r(c)$	Distance between $p(c)$ and customer $c$
$s(c)$	The closest stop among $\mathcal{S}$ to customer $c$
$d_s(c)$	Distance between $s(c)$ and customer $c$

The drone-direct scheme is easy to implement and fast to complete a delivery. However, a shortcoming is the limitation in delivery range due to the constrained on-board battery capacity. The drone-vehicle scheme overcomes this drawback thanks to the transportation provided by vehicles. A vehicle can take a drone to the position near the customer and such a position can be outside the direct flight range of a drone. Clearly, a negative point of the drone-vehicle scheme is that it may take a long time, since the drone may wait to catch vehicles.

We assume that the drone flies at a constant speed  $v$  and the energy consumption rate  $p$  is also a constant when it flies. When the drone travels with vehicles, it turns off the motors. Thus, we assume that the energy consumption during this time is zero, since it is negligible compared to that consumed by flying. We also assume that the battery recharging rate  $q$  is a constant. The value of  $q$  depends on the type of drones. For example, charging DJI Spark directly with USB takes 80 min. Notice that the used energy consumption and recharging models can be extended to some more accurate models. When a drone is at the depot, it recharges the battery up to  $E_0$ , which represents the battery capacity. A drone can be sent to deliver only if the residual energy is no less than the energy to be consumed for the delivery plus a safety residual energy  $\epsilon$ .



No matter which delivery scheme is used, after returning from the last customer  $c_0$  at time instant  $t_r(c_0)$ , the drone may start to recharge the battery. The recharging time is denoted by  $\tau_r(c)$  if the next customer to be delivered is  $c$ . Once the energy is sufficient, the drone may depart to deliver to customer  $c$  at instant  $t_d(c)$ . The return instant for customer  $c$  is denoted by  $t_r(c)$  and the trip time is  $\tau_t(c) = t_r(c) - t_d(c)$ . Throughout this paper the notation  $t$  with some subscript or superscript is used to represent a time instant, while  $\tau$  is used to represent a time interval. The relations between some important notations are demonstrated in Figure 2.

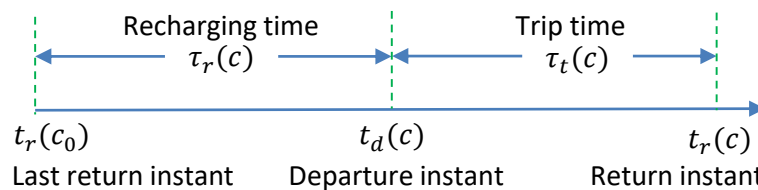


Figure 2. Illustration of key notations.

The two delivery schemes both have some limitations in terms of delivery area. For the drone-direct scheme, its delivery area is simply a circular area, denoted by  $\mathcal{A}_1$ . Let  $d_d(c)$  be the distance between customer  $c$  and the depot and let  $\tau_f^{dd}(c)$  denote the flight time to deliver to customer  $c$ . The drone takes off from the depot, flies to the customer, drops the parcel, returns to the depot and lands. We assume that the drone flies at a pre-defined altitude. The take-off time from the ground to this given altitude is denoted by  $\delta_1$  and that for landing is denoted by  $\delta_2$ . Notice that the flight time equals the trip time mentioned above plus the time for take-off and landing:

$$\tau_t^{dd}(c) = \tau_f^{dd}(c) = \frac{2d_d(c)}{v} + \delta_1 + \delta_2. \quad (1)$$

Here, the superscript  $dd$  is used to indicate the **drone-direct** scheme. Below, we will use the superscript  $dv$  to indicate the **drone-vehicle** scheme. We have the following constraint:

$$\frac{2d_d(c)}{v} p \leq E_0 - \epsilon - \epsilon_1 - \epsilon_2, \quad (2)$$

where  $\epsilon_1$  and  $\epsilon_2$  are the energy consumption for take-off and landing, respectively. From (2) we obtain:

$$d_d(c) \leq \frac{E_0 - \epsilon - \epsilon_1 - \epsilon_2 v}{2p}, \quad (3)$$

i.e., if the constraint (3) holds for customer  $c$ , we say customer  $c$  is within the delivery area  $\mathcal{A}_1$ .

To deliver to a customer  $c$ , there is a requirement of the residual energy of the drone, i.e., it should be no less than the to be consumed energy for  $c$  plus the safety residual energy  $\epsilon$ . When the drone returns from customer  $c_0$ , if the residual energy  $E$  is not enough to deliver to customer  $c$ , it needs to recharge the battery. Let  $\tau_r^{dd}(c) \geq 0$  denote the recharging time for  $c$ , and it should satisfy the following constraint:

$$E + \tau_r^{dd}(c)q - \tau_f^{dd}(c)p \geq \epsilon + \epsilon_1 + \epsilon_2, \quad (4)$$

from which we obtain:

$$\tau_r^{dd}(c) = \max\left\{0, \frac{\epsilon + \epsilon_1 + \epsilon_2 - E + \tau_f^{dd}(c)p}{q}\right\}. \quad (5)$$

In words, if the residual energy is sufficient to delivery customer  $c$ , the drone can depart immediately without recharging; otherwise, it has to recharge for a minimum time  $\tau_r^{dd}(c)$ . With



the recharging time and the above trip time, the return instant after delivering customer  $c$ , i.e.,  $t_r^{dd}(c)$ , is given by:

$$t_r^{dd}(c) = t_r(c_0) + \tau_r^{dd}(c) + \tau_t^{dd}(c). \quad (6)$$

For the drone–vehicle scheme, the delivery area  $\mathcal{A}_2$  further depends on the route  $\mathcal{R}$ . The trip time is no smaller than the flight time for customer  $c$ , i.e.,  $\tau_t^{dv}(c) \geq \tau_f^{dv}(c)$ , since the drone travels with vehicles in this scheme. For this scheme, we introduce some new notations. Let  $p_d$  be the point on  $\mathcal{R}$  which is the closest to the depot. Let  $d_0$  be the distance between the depot and  $p_d$ , see Figure 1. For customer  $c$ , let  $p(c)$  denote the point on  $\mathcal{R}$ , which is the closest to the customer  $c$ . Let  $d_r(c)$  denote the distance between the customer  $c$  and the point  $p(c)$ . Let  $s(c) \in \mathcal{S}$  denote the stop which is the closest to customer  $c$  and let  $d_s(c)$  denote the distance between customer  $c$  and stop  $s(c)$ . The flight time  $\tau_f^{dv}(c)$  for customer  $c$  is defined as follows and if it satisfies the right inequality, customer  $c$  can be delivered by the drone–vehicle scheme:

$$\tau_f^{dv}(c) = \frac{d_0 + d_r(c) + d_s(c) + d_0}{v} + \delta_1 + \delta_2. \quad (7)$$

We have the constraint on the energy expenditure:

$$\frac{d_0 + d_r(c) + d_s(c) + d_0}{v} p \leq E_0 - \epsilon - \epsilon_1 - \epsilon_2, \quad (8)$$

from which we can obtain:

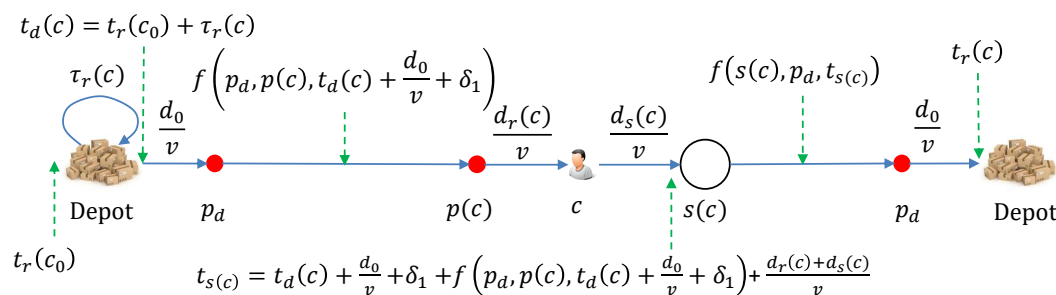
$$d_r(c) + d_s(c) \leq \frac{E_0 - \epsilon - \epsilon_1 - \epsilon_2}{p} v - 2d_0. \quad (9)$$

If the customer  $c$  satisfies the constraint (9), we say this customer locates within  $\mathcal{A}_2$ . There are two assumptions regarding the energy consumption. One is that the time needed to drop off the parcel is negligible. The other is that we assume that the stops and the point  $p_d$  are with some platforms where the drone can rest. Thus, the corresponding energy consumption when the drone is resting is ignored. More information about the second assumption will be given later in this section. Similar to (5), the time of recharging to deliver to customer  $c$  is given as follows:

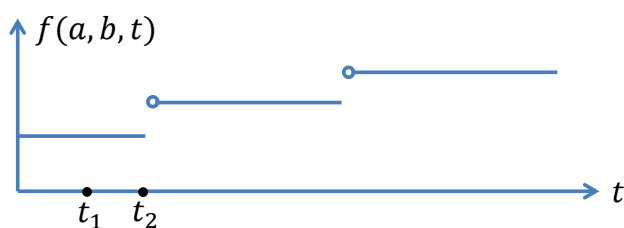
$$\tau_r^{dv}(c) = \max\left\{0, \frac{\epsilon + \epsilon_1 + \epsilon_2 - E + \tau_f^{dv}(c)p}{q}\right\}. \quad (10)$$

Now, we consider the model of return instant for the drone–vehicle scheme, see Figure 3. Let  $c_0$  denote the last delivered customer. After recharging the battery for  $\tau_r^{dv}(c)$ , the drone departs at instant  $t_d^{dv}(c) = t_r(c_0) + \tau_r^{dv}(c)$ . The drone then arrives at the point  $p_d$ , where it lands on the platform to rest and waits for the coming vehicle. The time needed to reach  $p_d$  is given by  $\frac{d_0}{v} + \delta_1$  and the arrival instant is  $t_d^{dv}(c) + \frac{d_0}{v} + \delta_1$ . Introduce a function  $f(a, b, t)$  which computes the minimum time needed for a considered vehicle to travel from point  $a$  to point  $b$  on route  $\mathcal{R}$  following the timetable  $\mathcal{T}$ , when the drone arrives at  $a$  at time instant  $t$ . Considering the basics of the timetable  $\mathcal{T}$ , the function  $f(a, b, t)$  is a multi-step function and the duration for each step and the gap between two consecutive steps may vary since the timetable  $\mathcal{T}$  may be time-dependent. An illustration is shown in Figure 4. Clearly, arriving  $p_d$  at time instants  $t_1$  and  $t_2$  result in the same time to reach  $p(c)$ . This is because that the vehicle arrives at  $p_d$  at time instant  $t_2$ . If the drone arrives at  $p_d$  later than  $t_2$ , it just misses the vehicle and has to wait for the next. When the drone arrives at  $t_1$ , it has to wait for a duration  $t_2 - t_1$ . Obviously, such a waiting may be unnecessary for the drone–vehicle scheme, as the drone can further recharge the battery for some time. Thus, we assume that the timetable  $\mathcal{T}$  is available to

the drone and the drone can depart the depot at an instant such that it arrives at  $p_d$  at the same with a vehicle. To achieve this, we introduce another function  $g(t)$  which computes the time for vehicles to arrive at  $p_d$  from their current positions, and this computation is done at instant  $t$ . Here, we need to do the computation once the drone is ready to depart, i.e., at instant  $t_d^{dv}(c)$ . As there are several vehicles running on the route at the same time, the function  $g(t_d^{dv}(c))$  returns a positive vector whose size equals the number of vehicles. We only consider the one which is the most closest to  $\frac{d_0}{v}$  and no smaller than  $\frac{d_0}{v}$ . For those which are smaller than  $\frac{d_0}{v}$ , the drone will never catch the corresponding vehicles. We denote such a value by  $\bar{g}(t_d^{dv}(c))$ . Then, the drone can recharge the battery at the depot for an extra period of  $\bar{g}(t_d^{dv}(c)) - \frac{d_0}{v} - \delta_1$ . Of course, arrival time at  $p(c)$  will remain the same, since the waiting time at  $p_d$  is simply shifted to the depot. The further recharged energy may reduce the time in the following recharging process.



**Figure 3.** An illustration to compute the several key time instants and intervals for the drone–vehicle scheme. Here the superscript  $dv$  is not displayed as this only suits the drone–vehicle scheme.



**Figure 4.** An example of function  $f(a, b, t)$ .

The arrival time at  $p_d$  can be still regarded as  $t_d^{dv}(c) + \frac{d_0}{v}$ . Then, the travel time from  $p_d$  to  $p(c)$  is given by  $f(p_d, p(c), t_d^{dv}(c) + \frac{d_0}{v} + \delta_1)$ . After that, the drone leaves the vehicle at point  $p(c)$ , flies to the customer  $c$ , drops off the parcel, and then flies to the stop  $s(c)$ . When the drone arrives at  $s(c)$ , the time instant, denoted by  $t_{s(c)}$ , is computed as follows:

$$t_{s(c)} = t_d^{dv}(c) + \frac{d_0}{v} + \delta_1 + f(p_d, p(c), t_d^{dv}(c) + \frac{d_0}{v} + \delta_1) + \frac{d_r(c) + d_s(c)}{v}. \tag{11}$$

Then, from  $s(c)$  to  $p_d$ , the travel time is  $f(s(c), p_d, t_{s(c)})$ , which is followed by the final flight from  $p_d$  to the depot. Therefore, the return instant  $t_r^{dv}(c)$  is as follows:

$$t_r^{dv}(c) = t_{s(c)} + f(s(c), p_d, t_{s(c)}) + \frac{d_0}{v} + \delta_2. \tag{12}$$

The trip time is:

$$\tau_t^{dv}(c) = t_r^{dv}(c) - t_d^{dv}(c). \tag{13}$$

**Remark 1.** Notice that for a given timetable of vehicles running on a simple route like the currently considered closed route, the functions  $f()$ ,  $g()$ , and  $\bar{g}()$  can be obtained easily. The whole picture of drone parcel delivery would be that one depot is associated with a fleet of drones and the drones can reach customers and return to the depot via multiple public transportation vehicles. Specifically, a drone can transfer between vehicles like passengers to reach its destination. In this case, these may be multiple positions like  $p_d$  in Figure 1, where the drone can join and leave the transportation network. We will meet the problem of how to estimate the time and energy cost for delivering a customer. In [29], we have developed a one-way trip planning algorithm to plan the path from the depot to a customer for a drone within a given deadline. We will improve it and develop a round-trip planning method. With this technique, we will further consider the scheduling problem in the whole picture: scheduling deliveries with multiple drones and multiple customers either directly or via the public transportation network.

**Remark 2.** We also note that the above models ignore the impact of package weight. In practice, the heavier the package, the larger the energy consumption rate. To accommodate this feature, we can extend the current models by introducing another notation, i.e.,  $w(c)$ , representing the energy consumption rate for delivering customer  $c$ . Note that the function  $w(c)$  depends on the package weight of customer  $c$ , and this function can be known by carrying out real experiments. With the function  $w()$ , the energy consumption models for both delivery schemes are modified as follows. Specifically, constraint (2) becomes:

$$\frac{d_d(c)}{v}(w(c) + p) \leq E_0 - \epsilon - \epsilon_1 - \epsilon_2. \quad (14)$$

Constraint (8) becomes:

$$\frac{d_0 + d_r(c)}{v}w(c) + \frac{d_s(c) + d_0}{v}p \leq E_0 - \epsilon - \epsilon_1 - \epsilon_2, \quad (15)$$

Other models for these two schemes can be modified accordingly.

So far, the recharging time, flight time, and trip time have been discussed for both schemes, respectively. If  $c \in \mathcal{A}_1 \cap \mathcal{A}_2$ , we need to make a decision on the selection of the delivery schemes. Introduce a binary variable  $x(c)$  signifying the selected scheme:

$$x(c) = \begin{cases} 1, & \text{if } c \text{ is served by drone-direct scheme;} \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

Then, the return instant for delivering customer  $c$ , the flight time and the trip time can be represented as follows:

$$t_r(c) = x(c)t_r^{dd}(c) + (1 - x(c))t_r^{dv}(c), \quad (17)$$

$$\tau_f(c) = x(c)\tau_f^{dd}(c) + (1 - x(c))\tau_f^{dv}(c), \quad (18)$$

$$\tau_t(c) = x(c)\tau_t^{dd}(c) + (1 - x(c))\tau_t^{dv}(c). \quad (19)$$

Consider a set of customers  $\mathcal{C}$  in the area  $\mathcal{A}_1 \cup \mathcal{A}_2$  ( $|\mathcal{C}| = n$ ). Given the timetable  $\mathcal{T}$ , the functions of  $f()$  and  $g()$  are known. Let  $\lambda$  denote a schedule containing the permutation  $\{c_1, c_2, \dots, c_n\}$ , the recharging times  $\{\tau_r(c_1), \tau_r(c_2), \dots, \tau_r(c_n)\}$ , and the selection of the delivery scheme  $\{x(c_1), x(c_2), \dots, x(c_n)\}$ . Clearly, for any customer  $c$ , both the return instant  $t_r(c)$  and energy consumption (which is proportional to  $\tau_f(c)$ ) depend only on  $\tau_r(c)$  and  $x(c)$ . In other words, with the basic information in  $\lambda$ , the return instant and the corresponding energy consumption can be further computed. Specifically, the initial energy is  $E_0$  (the initial recharging is not necessary, i.e.,  $\tau_r(c_1) = 0$ , since the battery is full) and let  $E$  represent the residual energy after delivering a customer. The energy consumed for serving customer  $c_1$  is given by  $p\tau_f(c_1)$ . The recharging time for customer  $c_2$  is given by  $\tau_r(c_2)$ . Thus, the residual energy of the drone just before the

departure for customer  $c_2$  is given by  $\min\{E_0, E + q\tau_r(c_2) - p\tau_f(c_1)\}$ . It is easy to compute the residual energy of the drone before the departure for any customer  $c_j$  ( $2 \leq j \leq n$ ) in the similar way, i.e.,  $\min\{E_0, E + q\sum_{i=1}^j \tau_r(c_i) - p\sum_{i=1}^{j-1} \tau_f(c_i)\}$ .

The problem under consideration is to figure out the optimal schedule  $\lambda$  such that the return instant after delivering the final customer is minimized. Let  $c_0$  be a virtual customer and  $t_r(c_0) = 0$ , or we say the starting instant is zero. Then, the return instant from  $c_1$  is given by  $t_r(c_1) = t_r(c_0) + \tau_r(c_1) + \tau_t(c_1)$ . The return instant from  $c_2$  is given by  $t_r(c_2) = t_r(c_1) + \tau_r(c_2) + \tau_t(c_2)$ . From them, we obtain  $t_r(c_2) = t_r(c_0) + \sum_{i=1}^2 \tau_r(c_i) + \sum_{i=1}^2 \tau_t(c_i)$ . Then, the return instant from  $c_n$  is  $t_r(c_n) = t_r(c_0) + \sum_{i=1}^n \tau_r(c_i) + \sum_{i=1}^n \tau_t(c_i)$ .

The problem of minimizing the final return instant is formulated as follows:

$$\min \sum_{i=1}^n (\tau_r(c_i) + \tau_t(c_i)) = \min \sum_{i=1}^n (\tau_r(c_i) + x(c_i)\tau_f^{dd}(c_i) + (1-x(c_i))\tau_f^{dv}(c_i)) \quad (20)$$

subject to

$$\begin{aligned} x(c_j)t_r^{dd}(c_j) + (1-x(c_j))t_r^{dv}(c_j) - \tau_r(c_j) > \\ x(c_{j-1})t_r^{dd}(c_{j-1}) + (1-x(c_{j-1}))t_r^{dv}(c_{j-1}), \forall 1 \leq j \leq n, \end{aligned} \quad (21)$$

$$\begin{aligned} \min\{E_0, E + q\sum_{i=1}^j \tau_r(c_i) - p\sum_{i=1}^{j-1} (x(c_i)\tau_f^{dd}(c_i) + (1-x(c_i))\tau_f^{dv}(c_i))\} - \\ p(x(c_j)\tau_f^{dd}(c_j) + (1-x(c_j))\tau_f^{dv}(c_j)) \geq \epsilon, \forall 2 \leq j \leq n. \end{aligned} \quad (22)$$

The variables of this problem include the delivery sequence  $c_1, \dots, c_n$ , the charging times  $\tau_r(c_1), \dots, \tau_r(c_n)$  and the delivery scheme selections  $x(c_1), \dots, x(c_n)$ . The constraints (21) require that the return instant for delivering a customer should always be greater than that for the last delivered customer by the recharging time. The constraints (22) require that for any customer  $c_j$  ( $2 \leq j \leq n$ ), the residual energy after delivering customer  $c_j$  no less than  $\epsilon$ . Notice that for the customer  $c_1$ ,  $E_0 - p\tau_f(c_1) \geq \epsilon$  holds as  $c_1 \in \mathcal{C}$  is inside the area  $\mathcal{A}_1 \cup \mathcal{A}_2$ .

The analysis of the problem complexity is based on the relaxation of the original problem to a problem with known complexity through the following assumptions:

- Only the drone-vehicle scheme is adopted;
- The energy constraints (22) are removed;
- The timetable  $\mathcal{T}$  does not vary with time, i.e., the vehicles keep running on the route periodically.

With the first assumption, the issue of scheme selection is not involved any more and the problem turns to be a conventional scheduling problem, where delivering a customer can be regarded as a job and the recharging time plus the flight time can be regarded as the processing time. With the second assumption, two jobs can be processed without idle time in between. With the third assumption, the functions  $f()$  and  $g()$  are not time varying. With these assumptions, the original problem (20)–(22) is reduced to the scheduling problem to minimize the makespan with multi-step deteriorating time on a single machine, which is NP-complete [30]. Obviously, without these assumptions, the original problem is much more complex. It is costly to address the considered problem optimally due to its NP-hardness.

#### 4. Proposed Solution

In this section, we discuss how to address the original problem (20). We first present a dynamic programming-based exact algorithm that can figure out the optimal schedule, and then we develop a sub-optimal algorithm.

#### 4.1. Dynamic Programming-Based Exact Algorithm

Consider a subset of customers  $\mathcal{M} \subseteq \mathcal{C}$  and a customer  $c \in \mathcal{M}$ . Denote by  $T(\mathcal{M}, c)$  the earliest return instant to deliver to all the customers in  $\mathcal{M}$  with  $c$  as the final delivered customer. Clearly, if the size of  $\mathcal{M}$  is 1, i.e.,  $|\mathcal{M}| = 1$ , then  $\mathcal{M} = \{c\}$  and  $T(\mathcal{M}, c) = 0 + \tau_r(c) + \tau_t(c)$ , where 0 represents the initial instant. Notice that for the determination of the delivery scheme for each customer, the scheme which makes the drone to return to the depot at the earliest instant is selected. If  $|\mathcal{M}| > 1$ , consider that the drone delivers the customer  $c$  right after delivering the customer  $c_0$ . The return instant  $T(\mathcal{M}, c)$  after delivering  $c$  should equal to the return instant after delivering  $c_0 \in \mathcal{M} - \{c\}$  plus the recharging time and the trip time for  $c$ , i.e.,  $\tau_r(c)$  and  $\tau_t(c)$ . With this idea,  $T(\mathcal{M}, c)$  can be computed in a standard recursive manner as follows:

$$T(\mathcal{M}, c) = \min_{c_0 \in \mathcal{M} - \{c\}} \{T(\mathcal{M} - \{c\}, c_0) + \min_{dd, dv} \{\tau_r^{dd}(c) + \tau_t^{dd}(c), \tau_r^{dv}(c) + \tau_t^{dv}(c)\}\}. \quad (23)$$

Based on (23), the exact algorithm is presented in Algorithm 1. If  $|\mathcal{M}| = 1$ ,  $T(\mathcal{M} = \{c\}, c)$  can be computed straightforwardly, i.e.,  $T(\mathcal{M}, c) = \min_{dd, dv} \{\tau_r^{dd}(c), \tau_t^{dv}(c)\}$ . Here, as this is the first delivery, the drone does not need to recharge. So, both  $\tau_r^{dd}(c)$  and  $\tau_r^{dv}(c)$  are 0. With all of  $T(\mathcal{M} = \{c\}, c)$  pre-computed, the exact algorithm checks  $\mathcal{M}$  of size 2, 3, and up to  $n$ . For each size, the algorithm figures out the best last second customer  $c_0$  such that  $T(\mathcal{M}, c)$  is minimized. Obviously, to achieve this, the already computed  $T(\mathcal{M} - \{c\}, c_0)$  ( $\forall c_0 \in \mathcal{M} - \{c\}$ ) is used. Moreover, in Line 4, the algorithm selects the delivery scheme by comparing  $\tau_r^{dd}(c) + \tau_t^{dd}(c)$  and  $\tau_r^{dv}(c) + \tau_t^{dv}(c)$ , and the smaller one is selected. Notice that, the calculations of  $\tau_r^{dd}(c)$ ,  $\tau_t^{dd}(c)$ ,  $\tau_r^{dv}(c)$ , and  $\tau_t^{dv}(c)$  all depend on the return instant of the last visited customer, i.e.,  $c_0$  in Line 4. The residual energy  $E$  should be updated in the similar way as  $T(\mathcal{M}, c)$  to assist these computations.

---

**Algorithm 1:** The exact algorithm.

---

```

for  $c \in \mathcal{C}$  do
  for  $k = 2, \dots, n$  do
    for any  $\mathcal{M} \in \mathcal{C}$  of size  $k$  and containing  $c$  do
       $T(\mathcal{M}, c) \leftarrow$ 
       $\min_{c_0 \in \mathcal{M} - \{c\}} \{T(\mathcal{M} - \{c\}, c_0) + \min_{dd, dv} \{\tau_r^{dd}(c) + \tau_t^{dd}(c), \tau_r^{dv}(c) + \tau_t^{dv}(c)\}\}$ 
    end
  end
end

```

---

**Optimality:** We first state that there may exist multiple optimal solutions to the considered problem. It is easy to understand that in some extreme cases where all the customers are inside the delivery range of the drone and using drone-direct scheme leads to shorter delivery time for all these customers, the drone-vehicle scheme will not be used for any customers. Then, the drone should depart with the energy of  $E_0$  and when it returns the residual energy should be  $\epsilon$ , since this is the best way to save time on recharging. The final return instant is independent on the customer permutation, if the drone recharges the battery for a just sufficient time to deliver the next customer. The existence of multiple optimal solutions also holds when the drone-vehicle scheme is used. For example, in the case where two customers locate on two sides of the vehicle route and they have the same point on the route for the drone to leave the vehicle and the same drone returning stop. Then, delivering these two customers in any order leads to the same trip time and recharging time.

We now state that the solution obtained by Algorithm 1, i.e., including the permutation  $\{c_1, c_2, \dots, c_n\}$ , the recharging times  $\{\tau_r(c_1), \tau_r(c_2), \dots, \tau_r(c_n)\}$ , and the selection of the delivery scheme  $\{x(c_1), x(c_2), \dots, x(c_n)\}$ , is optimal in terms of the overall delivery time. In Algorithm 1, we want to find out the last customer to be delivered among the set of  $\mathcal{C}$ , such that the overall delivery

time is the shortest. This time is stored in  $T(\mathcal{C}, c)$ , if  $c$  is the last customer. To achieve this, we check all the customers in the set of  $\mathcal{C}$ , see Line 1. Since  $c$  is the last customer to be delivered, the overall delivery time should be the sum of the return instant from the  $n - 1$ th customer ( $T(\mathcal{C} - \{c\}, c_0)$ ) and the recharging and trip time for the  $n$ th customer  $c$ , see Line 4. Then, we need to know what is the return instant to deliver  $n - 1$  customers. Thus, for a customer  $c$  as the last customer, we need to know the return instant to deliver  $n - 1, n - 2, \dots, 2$  customers, and we can call them in a recursive manner following (23). For implementation, we compute them in a forward way (see Lines 2–6), since it is impossible to compute the return instant to deliver to a larger customer set before the completion of its subsets. After these computations, we can have the smallest return instant  $T(\mathcal{C}, c)$ , where  $c$  is the last customer. As there are  $n$  selections for the last customer, we pick the smallest one among  $n$  values of  $T(\mathcal{C}, c)$ , and this gives the shortest delivery time. Also, the last customer is known, and by a standard backtracking algorithm, the permutation, the recharging time, and the delivery scheme selection can be computed easily.

**Complexity:** we also analyze the complexity of this algorithm. For  $n$  customers, there are  $O(n)$  choices for the customer  $c$ . For a certain customer  $c$  and given that  $\forall c_0 \in \mathcal{M} - \{c\}$ ,  $T(\mathcal{M} - \{c\}, c_0)$  has been computed already, for each subset  $\mathcal{M}$ , the complexity of (23) is  $O(|\mathcal{M}|)$  and the worst case is  $O(n)$ . In the dynamic programming term, solving (23) is solving a subproblem. Thus, we need to know how many subproblems to be solved, which is consistent with the number of subsets we need to generate. Clearly, there are  $C_{n-1}^{k-1}$  subsets having customer  $c$  and with the size of  $k$  (here the notation  $C_{n-1}^{k-1}$  represents the combination number). Then, for  $n$  customers, the total number of subproblems is  $O(\sum_{k=1}^{n-1} C_{n-1}^{k-1}) = O(2^n)$ . Therefore, the overall complexity of the exact algorithm is  $O(n^2 2^n)$ . This is much less than that of the naive algorithm which evaluates all the permutations of customers, i.e.,  $O(n!)$ . However, it is still infeasible for a slightly higher number of customers, because both the time complexity as well as the space required (to store  $T(\mathcal{M}, c)$  and the corresponding residual energy) are exponential with respect to  $n$ . As a consequence, it is reasonable to design a heuristic algorithm to find sub-optimal solutions, which should be computationally efficient and applicable to large scale cases.

#### 4.2. Sub-Optimal Algorithm

Considering the weakness of the exact algorithm in scaling, this subsection presents a sub-optimal algorithm which can quickly find a feasible solution to the scheduling problem (20). The basic idea of the sub-optimal algorithm is to first schedule the customer which enables the earliest return and we order the customers iteratively. Specifically, when we consider whether to put a customer  $c$  in the queue, we compute the time needed to recharge the battery upon the remaining energy. As the recharging time relates to the flight time which does not depend on the departure instant (see (2) and (8)), the recharging times for all the remaining customers can be computed. Then, the corresponding departure instant is known for each customer, and the return instants can be calculated further. The customer with the earliest return instant (departure instant plus the recharging time and the trip time) is scheduled.

As shown in Algorithm 2, in Line 1, the initial return instant  $t_r$  is set as 0 and the initial energy is  $E_0$ . When there are some customers unscheduled in the set  $\mathcal{C}$ , for a customer  $c$ , the corresponding recharging time and trip time for the two delivery schemes can be computed according to (2), (5), (10) and (13). With these values, the return instants for the two delivery schemes are obtained (Line 4 and 5). Furthermore, it makes a decision between the two delivery schemes by selecting the one that leads to the earliest return instant for customer  $c$  (Line 6). After checking all the remaining customers in  $\mathcal{C}$ , the customer which has the earliest return instant is found out, see Line 8. By backtracking, the delivery scheme for this customer can be known. Then, in Line 9, the residual energy is updated based on the selected customer to be delivered  $c$  and it is also removed from the set  $\mathcal{C}$ . The scheduling process terminates when all the customers are scheduled.

**Complexity:** the computational complexity of this sub-optimal algorithm is only  $O(n^2)$ , where  $O(n)$  is for the while-loop and  $O(n)$  is for the for-loop, which is much less than that of the exact



algorithm. However, this algorithm generally cannot give the optimal solution. The approximation factor of this algorithm to the exact algorithm depends on specific cases and in the next section, we will demonstrate the performance gap between them via computer simulations.

---

**Algorithm 2:** The sub-optimal algorithm.

---

```

 $t_r \leftarrow 0, E \leftarrow E_0.$  while  $\mathcal{C}$  is not empty do
  for any  $c \in \mathcal{C}$  do
     $t_r^{dd}(c) \leftarrow t_r + \tau_r^{dd}(c) + \tau_t^{dd}(c).$ 
     $t_r^{dv}(c) \leftarrow t_r + \tau_r^{dv}(c) + \tau_t^{dv}(c).$ 
     $t_r(c) \leftarrow \min_{dd,dv} \{t_r^{dd}(c), t_r^{dv}(c)\}.$ 
  end
   $t_r \leftarrow \min_c \{t_r(c)\}.$ 
   $E \leftarrow \min \{E_0, E + q\tau_r^{dd|dv}(c)\} - p\tau_f^{dd|dv}(c).$ 
   $\mathcal{C} \leftarrow \mathcal{C} - \{c\}.$ 
end

```

---

## 5. Simulation Results

In this section, the results of computer simulations, conducted by Matlab, are presented to demonstrate the performance of the sub-optimal algorithm against the exact algorithm. The parameter turnings are investigated as well.

In an area as shown in Figure 5a, some vehicles keep running on the given route. There are 10 stops along this route where the drone can rest (represented by triangles with labels). There is a timetable  $\mathcal{T}$  associated with these stops, see Figure 5b where the unit is minute. The vehicles are running in two directions: clockwise and counterclockwise. We generate the basis of the timetable which is the middle two rows in Figure 5b. Specifically, at instant 0, there is a vehicle departs stop 1, it moves counterclockwise and arrives at stop 2 at instant 4, stop 3 at instant 7, etc. This vehicle finally returns to stop 1 via stop 10 and the arrival time at stop 1 is at instant 40. At instant 0, another vehicle departs stop 1 and it moves clockwise. It arrives at stop 10 at instant 6 and at stop 9 at instant 10. With this basis, we generate the first two rows' and the last two rows' timetable, where the interval between two vehicle departures is 5 min. More parts of the timetable can be generated in the same way, but they are not shown. As the timetable  $\mathcal{T}$  only provides the information about the time instants at stops rather than the positions between two stops, the time instants at which the vehicles pass the nearest points to customers need to be estimated. For this, we assume the vehicles run at constant speeds between any two neighbour stops. Other system parameters are:  $E_0 = 32$  unit,  $\epsilon = 2$  unit,  $p = q = 1$  unit per minute, and  $v = 1$  km per minute (The values for the parameters  $E_0$ ,  $p$  and  $v$  are based on the data of the Amazon Prime Air project in which a drone can fly about 32 km in 30 min.).

Firstly, consider a case with 5 randomly generated customers, see Figure 6, where the schedules by the two algorithms are shown. The labels near the customers give their positions in the permutations. The delivery schemes are marked differently: red squares are by drone-vehicle scheme and blue circles are by drone-direct scheme. The optimal schedule delivers to one customer by the drone-direct scheme while the sub-optimal schedule delivers two customers by the drone-direct scheme. Their delivery permutations are different. The schedule by the exact algorithm uses 210 min to deliver to all the customers and the algorithm running time is about 0.333 s, while that of the sub-optimal algorithm uses 215 min, which is very close to the optimal solution, and it takes only 0.001 s to run the algorithm (The algorithm running time is measured in Matlab on a PC with: Intel(R) i5-3570S CPU and 8 GB RAM). For the comparison with the drone-direct only scheme, we also show the delivery area by this scheme (which is the black dash circle). For this example, only 2 out of the 5 customers can be delivered by the drone-direct scheme, while they all can be delivered by the proposed delivery system. This feature exists in all the other simulated cases. Of course, due to the randomness, there are some



customers that cannot even be delivered by the drone–vehicle scheme, but the deliverable ratio of the drone–vehicle scheme is generally larger than that of the drone-direct scheme.

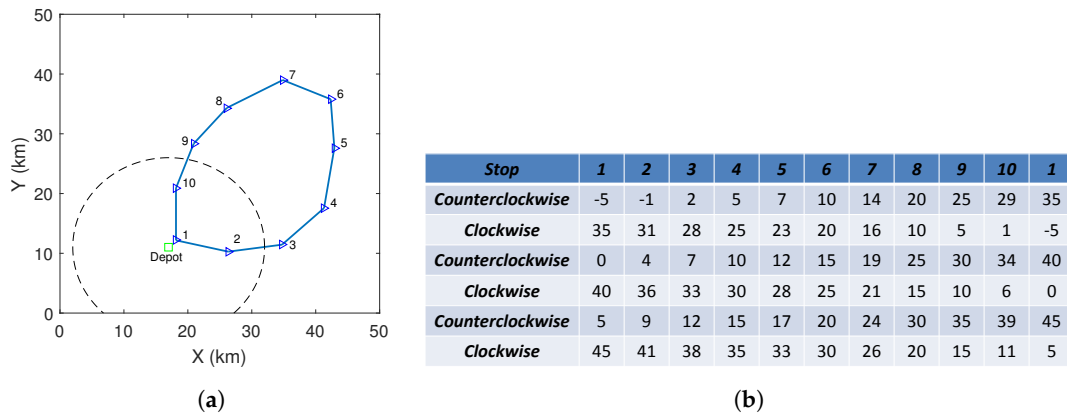


Figure 5. (a) The considered area with a circular trajectory. (b) Timetable  $\mathcal{T}$  (in minutes).

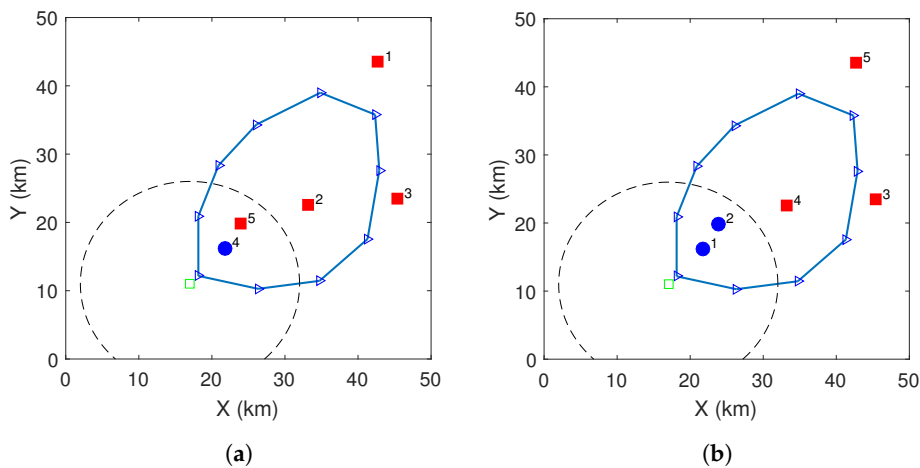
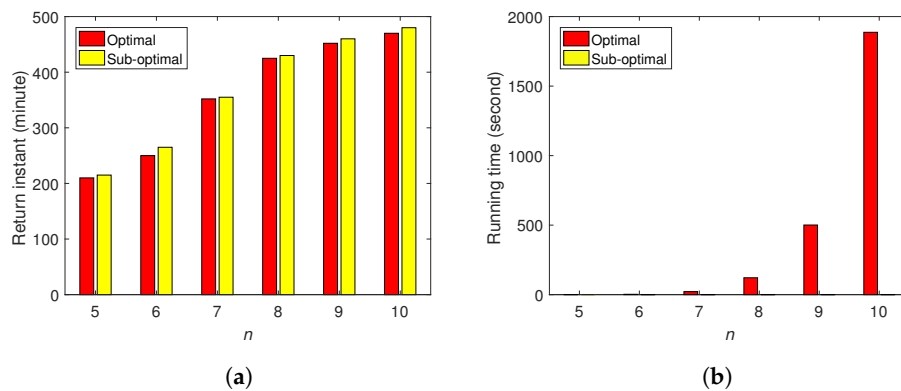


Figure 6. (a) Schedule by the exact algorithm. (b) Schedule by the sub-optimal algorithm.

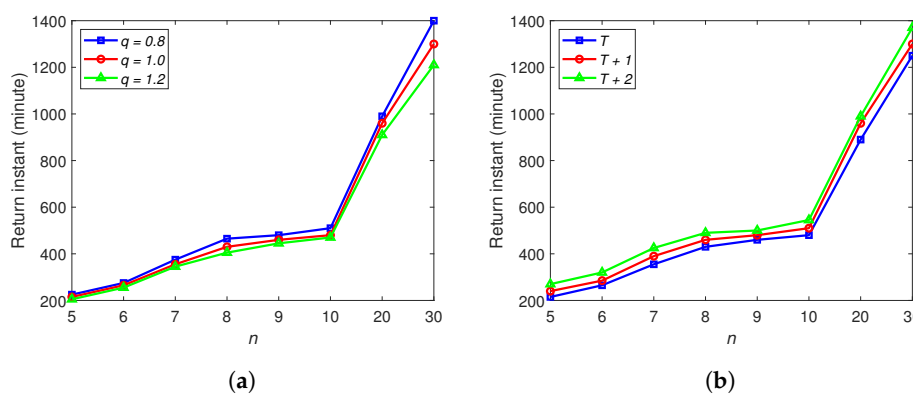
We conduct more simulations for different numbers of customers  $n$  to demonstrate the performance of the sub-optimal algorithm against the exact algorithm. As the complexity of the exact algorithm is exponential to  $n$ , we only conduct simulations with up to 10 customers using the exact algorithm. As shown in Figure 7a, in all these simulations, the return instants by the sub-optimal algorithm are only 5 to 15 min more than the schedules obtained by the exact algorithm. The advantage of the sub-optimal algorithm lies in the computational efficiency. As shown in Figure 7b, the algorithm running time of the exact algorithm increases dramatically with  $n$ . However, that of the sub-optimal algorithm is lower than 0.1 s for these cases. From these results we can see that the proposed sub-optimal algorithm achieves close performance to the optimal solution and it is computationally efficient.



**Figure 7.** (a) Return instants for different  $n$ . (b) Algorithm running time. The results of the sub-optimal algorithm are all lower than 0.1 s.

Moreover, we investigate the influence of some key parameters on the sub-optimal algorithm. Here, we consider large scale cases where the number of customers can be 20, 30, and even more. The customer locations are again randomly generated and the sub-optimal can address the case with 30 customers in seconds.

We first consider the parameters of  $p$  and  $q$ . To study their impacts, we fix  $p$  as 1 and consider some values of  $q$ . When  $q < 1$ , the recharging time becomes longer to recharge a certain amount of energy than the consumption of the same amount; while when  $q > 1$ , the recharging process becomes faster. It can be seen from Figure 8a compared to the cases with  $q = 1$ , when  $q = 1.2$ , the return instants for the above cases become smaller; while when  $q$  is decreased to 0.8, the return instants all become larger. Clearly, the larger the  $q$ , the more efficient in terms of recharging, and then the drone can be back to deliver the next customer sooner.

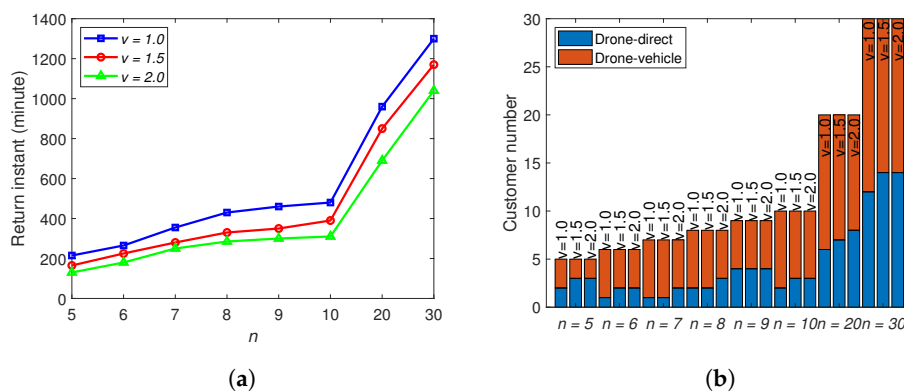


**Figure 8.** (a) The influence of  $q$ . (b) The influence of  $T$ .

The timetable  $T$  also has a significant impact on the scheduling performance. We create two more timetables:  $T + 1$  and  $T + 2$ . Based on  $T$ , we increase the time interval between two neighbour stops by 1 min and 2 min, respectively, to get  $T + 1$  and  $T + 2$ . As seen in Figure 8b, with  $T + 2$ , the drone uses the longest time to complete the deliveries, which is followed by the timetable  $T + 1$ . In other words, a longer interval between stops worsens the time for delivery, because it takes a longer time to travel with the vehicles if the drone-vehicle scheme is selected.

Finally, the impact of the drone speed  $v$  is considered. In all the above simulations, the value of  $v$  together with those of  $E_0$  and  $p$  are consistent with Amazon Prime Air project. Here, we keep the values for  $E_0$  and  $p$ , and consider some higher speed. The timetable  $T$  is used here. According to our model, increasing  $v$  enlarges the delivery area of both the drone-direct scheme and drone-vehicle scheme, see (3) and (9). The final return instants for all the cases have a decreasing trend, see Figure 9a,

with the increase of  $v$ . However, when  $v$  doubles, the new return instants are not half of the old values. The reason lies in the utility of the vehicles, whose travel times are not impacted by  $v$ . Moreover, when  $v$  increases, the delivery areas of these two schemes are enlarged. Then, the delivery scheme selection will be different. As shown in Figure 9b, the ratio of using drone-direct scheme has an increasing trend with the increase of  $v$ , because higher speed not only increases the number of deliverable customers but also decreases the trip time, which is much more significant for the drone-direct scheme than the drone-vehicle scheme.



**Figure 9.** (a) The influence of  $v$ . (b) The number of customers served by different delivery schemes for different  $v$ .

For summary, in this section, we have shown that the sup-optimal achieves close performance to the exact algorithm and the former is much more computationally efficient than the latter. The study of the parameter turnings presents how these parameters impact the system performance.

## 6. Conclusions and Future Work

In this paper, a novel parcel delivery system consisting of a drone and public transportation vehicles was proposed. It involves the drone-direct scheme and the drone-vehicle scheme. The fundamental characteristics such as the trip time, energy consumption and battery recharging were modelled. A time-dependent scheduling problem for a single drone was formulated, which is NP-complete. An exact algorithm was first presented but it is computationally inefficient. Then, a sub-optimal solution was developed, which achieves close performance to the exact algorithm and is much more computationally efficient than the exact algorithm. The study of the parameter turnings presents how these parameters impact on the system performance. One future research direction is to replace the vehicles with a simple route by a more complex public transportation network. This definitely enlarges the delivery area but also brings some challenges in the estimations of trip time, since the transportation network is with more uncertainty. Another direction is to develop algorithms for multiple drones. The third is to extend the static scenario where all the orders are ready before the scheduling to the stochastic scenario where customers place orders in a dynamic manner. The latter case will be useful in real applications where suppliers wish to service the dynamic orders in the shortest time.

**Author Contributions:** Conceptualization, H.H. and C.H.; Methodology, H.H.; Software, C.H. and H.H.; Validation, H.H. and C.H.; Formal Analysis, H.H.; Investigation, H.H.; Resources, H.H.; Data Curation, H.H.; Writing—Original Draft Preparation, H.H.; Writing—Review & Editing, A.V.S.; Visualization, C.H.; Supervision, A.V.S.; Project Administration, H.H.; Funding Acquisition, A.V.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Australian Research Council.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Huang, H.; Savkin, A.V. Towards the Internet of Flying Robots: A Survey. *Sensors* **2018**, *18*, 4038. [[CrossRef](#)] [[PubMed](#)]
2. Huang, H.; Savkin, A.V. A Method for Optimized Deployment of Unmanned Aerial Vehicles for Maximum Coverage and Minimum Interference in Cellular Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2638–2647. [[CrossRef](#)]
3. Marchewka, A.; Ziółkowski, P.; Aguilar-Vidal, V. Framework for Structural Health Monitoring of Steel Bridges by Computer Vision. *Sensors* **2020**, *20*, 700. [[CrossRef](#)] [[PubMed](#)]
4. Barbedo, J.G.A.; Koenigkan, L.V.; Santos, T.T.; Santos, P.M. A Study on the Detection of Cattle in UAV Images Using Deep Learning. *Sensors* **2019**, *19*, 5436. [[CrossRef](#)] [[PubMed](#)]
5. Savkin, A.V.; Huang, H. A Method for Optimized Deployment of a Network of Surveillance Aerial Drones. *IEEE Syst. J.* **2019**, *13*, 4474–4477. [[CrossRef](#)]
6. Savkin, A.V.; Huang, H. Proactive Deployment of Aerial Drones for Coverage over Very Uneven Terrains: A Version of the 3D Art Gallery Problem. *Sensors* **2019**, *19*, 1438. [[CrossRef](#)]
7. Savkin, A.V.; Huang, H. Asymptotically optimal deployment of drones for surveillance and monitoring. *Sensors* **2019**, *19*, 2068. [[CrossRef](#)]
8. Ham, A.M. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transp. Res. Part C Emerg. Technol.* **2018**, *91*, 1–14. [[CrossRef](#)]
9. Macias, J.E.; Angeloudis, P.; Ochieng, W. Optimal hub selection for rapid medical deliveries using unmanned aerial vehicles. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 56–80. [[CrossRef](#)]
10. Liu, X.; Zhang, S.; Tian, J.; Liu, L. An Onboard Vision-Based System for Autonomous Landing of a Low-Cost Quadrotor on a Novel Landing Pad. *Sensors* **2019**, *19*, 4703. [[CrossRef](#)]
11. Murray, C.C.; Chu, A.G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 86–109. [[CrossRef](#)]
12. Chang, Y.S.; Lee, H.J. Optimal delivery routing with wider drone-delivery areas along a shorter truck-route. *Expert Syst. Appl.* **2018**, *104*, 307–317. [[CrossRef](#)]
13. Agatz, N.; Bouman, P.; Schmidt, M. Optimization approaches for the traveling salesman problem with drone. *Transp. Sci.* **2018**, *52*, 965–981. [[CrossRef](#)]
14. Ha, Q.M.; Deville, Y.; Pham, Q.D.; Hà, M.H. On the min-cost traveling salesman problem with drone. *Transp. Res. Part C Emerg. Technol.* **2018**, *86*, 597–621. [[CrossRef](#)]
15. Kim, S.; Moon, I. Traveling salesman problem with a drone station. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, *49*, 42–52. [[CrossRef](#)]
16. Mathew, N.; Smith, S.L.; Waslander, S.L. Planning Paths for Package Delivery in Heterogeneous Multirobot Teams. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1298–1308. [[CrossRef](#)]
17. Bai, X.; Cao, M.; Yan, W.; Ge, S.S. Efficient Routing for Precedence-Constrained Package Delivery for Heterogeneous Vehicles. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 248–260. [[CrossRef](#)]
18. Boysen, N.; Briskorn, D.; Fedtke, S.; Schwerdfeger, S. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks* **2018**, *72*, 506–527. [[CrossRef](#)]
19. Wang, Z.; Sheu, J.B. Vehicle routing problem with drones. *Transp. Res. Part Methodol.* **2019**, *122*, 350–364. [[CrossRef](#)]
20. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle routing problems for drone delivery. *IEEE Trans. Syst. Man, Cybern. Syst.* **2017**, *47*, 70–85. [[CrossRef](#)]
21. Grippa, P.; Behrens, D.A.; Wall, F.; Bettstetter, C. Drone delivery systems: Job assignment and dimensioning. *Auton. Robot.* **2019**, *43*, 261–274. [[CrossRef](#)]
22. Torabbeigi, M.; Lim, G.J.; Kim, S.J. Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates. *J. Intell. Robot. Syst.* **2020**, *97*, 471–487. [[CrossRef](#)]
23. Hong, I.; Kuby, M.; Murray, A.T. A range-restricted recharging station coverage model for drone delivery service planning. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 198–212. [[CrossRef](#)]
24. Shavarani, S.M.; Mosallaeipour, S.; Golabi, M.; İzbirak, G. A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system. *Comput. Oper. Res.* **2019**, *108*, 57–68. [[CrossRef](#)]

25. Yoo, H.D.; Chankov, S.M. Drone-delivery Using Autonomous Mobility: An Innovative Approach to Future Last-mile Delivery Problems. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 16–19 December 2018; pp. 1216–1220.
26. Trotta, A.; Andreagiovanni, F.D.; Di Felice, M.; Natalizio, E.; Chowdhury, K.R. When UAVs Ride A Bus: Towards Energy-efficient City-scale Video Surveillance. In Proceedings of the IEEE Conference on Computer Communications, Chengdu, China, 7–10 December 2018; pp.1043–1051.
27. Sundar, K.; Rathinam, S. Algorithms for Routing an Unmanned Aerial Vehicle in the Presence of Refueling Depots. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 287–294. [[CrossRef](#)]
28. Yu, K.; Budhiraja, A.K.; Tokekar, P. Algorithms for Routing of Unmanned Aerial Vehicles with Mobile Recharging Stations. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–5.
29. Huang, H.; Savkin, A.V.; Huang, C. Reliable Path Planning for Drone Delivery using a Stochastic Time-Dependent Public Transportation Network. *IEEE Trans. Intell. Transp. Syst.* **2020**, in press.
30. Mosheiov, G. Scheduling jobs with step-deterioration; minimizing makespan on a single-and multi-machine. *Comput. Ind. Eng.* **1995**, *28*, 869–879. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).