UAV-Assisted Routing Algorithm for Truck Parcel Delivery

Chung-Yi Tsai, Jang-Ping Sheu, and Pi-Yu Yi
Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
jid050887@gapp.nthu.edu.tw, sheujp@cs.nthu.edu.tw, go111062685@gapp.nthu.edu.tw

***Abstract*—This paper studies how UAVs can efficiently deliver parcels with trucks in rural areas. Due to the limited payload of the UAVs, they can only serve the parcels under the payload limitation, and the other parcels are delivered by truck. Furthermore, the battery capacities of the UAVs are limited, and they cannot deliver all parcels at once. The UAV will take off from the truck to deliver parcels and then meet with the truck to load new parcels and change the new battery for the next trip. For UAV safety considerations, the truck must wait at the rendezvous node for the UAV to land. Our paper aims to deliver all the parcels as quickly as possible. Since the problem is NP-hard, we proposed a three-stage heuristic algorithm to solve this problem. The simulation results show that our proposed algorithm outperforms the candidate algorithms in minimizing task completion time.**

***Index Terms*—Unmanned aerial vehicle, trajectory planning, truck-UAV cooperation.**

I. Introduction

Unmanned aerial vehicles (UAVs, also known as drones) in various applications have gained more attention recently. Unlike traditional vehicles, UAVs are not restricted by terrain and do not require a human pilot. UAVs are fast, lightweight, and easy to deploy. As a result, UAVs have been widely used to support various civilian, commercial, and military applications, including edge computing [1], data sensing and relay [2], and UAV-assisted parcel delivery [3]. With the prosperity of e-commerce, the logistics industry has become increasingly important. The parcel delivery requirements have significantly increased. Some parcels may be located at remote locations, where the truck needs to pass obstacles, such as lakes and hills, to reach destinations. The truck would need additional delivery time, and fuel costs with long-distance driving will bring more atmospheric pollution. The labor costs will also increase due to the increasing delivery time. Thus, many researchers seek more efficient ways to solve the delivery problem.

Some works provide joint car-sharing to deliver parcels [4]. However, the car may suffer traffic congestion and extend the required delivery time. Since UAVs have high mobility and low cost, UAVs have the potential to significantly reduce the cost and time of making last-mile deliveries [5]. Many enterprises, including Google, Amazon, and Federal Express, have recently tried to add UAVs to their commercial package delivery services. Some studies have used only UAVs to deliver parcels [5], [6] . It can be seen that due to the power limitation of the UAV, the area it can serve is limited. Furthermore, the UAV cannot deliver heavy parcels. Insufficient payload and limited battery capacity make using only UAVs to deliver parcels unsuitably. The truck has a large load capacity and better endurance than UAVs. Trucks can serve as a mobile depot to provide UAVs with a way to change batteries and reload parcels, increasing UAVs' flight range [7]. On the other hand, UAVs can ignore most terrain restrictions and deliver parcels, decreasing the delivery cost. Thus, the truck-UAV cooperative delivery system seems more efficient than the truck-only or UAV-only delivery system.

Some studies investigated leveraging trucks and UAVs to collaborate in delivering parcels to customers [8], [9]. These works assume UAVs can only deliver one parcel on each flight. However, we do not need the restriction since current UAV technology has allowed us to pick up small items, such as vaccines and medicines, and deliver them one at a time. So, without exceeding their load capability and battery constraint, UAVs can carry several parcels and deliver them in one flight [10], [11]. In this situation, UAVs save more time between trips to and from the truck. When the UAV can take multiple parcels, we must consider which parcels will be delivered together. The latter problem becomes more complicated compared to the former one. Besides using only one UAV to collaborate with the truck, several works used multiple UAVs to assist one truck in delivering parcels [12], [13]. More UAVs can make deliveries faster, but scheduling multiple UAVs makes the problem more complicated and costly. So, it is often assumed that the truck can only dispatch or recycle one UAV at a time.

This work considers a truck carrying a UAV and delivering parcels in rural areas. The UAV can deliver multiple parcels in one flight under payload and battery constraints. There are two types of parcels. Some parcels are located where the truck is inaccessible with terrain obstacles such as mountains, lakes, and lanes. These parcels can only be delivered by the UAV, which we name this type of parcel as UAV parcel (UP). The second type of parcel can only be delivered by truck, which we call a truck parcel (TP). Our main objective is to minimize total parcel delivery time. The proposed scheme is composed of three stages. First, we find a truck route to visit all the TPs and model the route as a traveling salesperson problem (TSP). Several rendezvous candidate nodes (RCNs) are on the truck route for UAVs selected as take-off or landing nodes. Second, we partition all the UPs into clusters according to their distances and weights. Third, we choose take-off and landing nodes from the truck route for each cluster to reduce

This work was supported in part by the National Science and Technology Council, Taiwan, under grant 112-2221-E-007-046-MY3 and Qualcomm Technologies, Inc. under grant SOW NAT-487844.

2024 IEEE Wireless Communications and Networking Conference (WCNC) | 979-8-3503-0358-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/WCNC57260.2024.10571229

the parcel delivery time. The simulation results show that the proposed algorithm outperforms the baselines in minimizing parcel delivery time.

The rest of this paper is organized as follows. Section II introduces the system model and problem formulation. Section III presents a three-stage heuristic algorithm to solve the problem. Section IV shows the simulation results, and Section V concludes.

II. System Model and Problem Formulation

A. System Model

We consider a delivery network consisting of one truck with one UAV to deliver n parcels. Let $\mathcal{P} = \{p_1, ..., p_n\}$ denote the set of n parcels. As mentioned, there are two types of parcels: the UAV parcel (UP) and the truck parcel (TP). Let the first h parcels in $\mathcal{P}$ be UPs, and we name the set of UPs as $\mathcal{P}^u = \{p_1^u, ..., p_h^u\}$. The rest of $n-h$ parcels are TPs, denoted as $\mathcal{P}^t = \{p_1^t, ..., p_{n-h}^t\}$. Let w_i be the weight of parcel p_i. The truck will depart from the depot and deliver all the TPs sequentially, and the truck can only travel on the roads.

On the other hand, the UAV can fly in any direction. In this scenario, the truck can carry all parcels. However, the UAV's payload and battery capacity are limited. Therefore, the UAV must take off or land on the truck to reload the parcels and change its battery to serve all the UPs. The maximum payload the UAV can carry is w_{max}, and the maximum distance the UAV can fly is d_{max}. We set several RCNs along the truck route for the UAV to take off or land from the truck. The truck must reach the landing node before the UAV for safety considerations. The speed of the truck and UAV are denoted as v_t and v_u, respectively. In our system model, the UAV can deliver several parcels as long as the weight does not exceed its payload limit and its travel distance does not exceed the distance constraint. The objective is to find the truck and UAV routes that minimize total parcel delivery time. Fig. 1 shows an example of the truck and UAV routes to deliver all parcels.

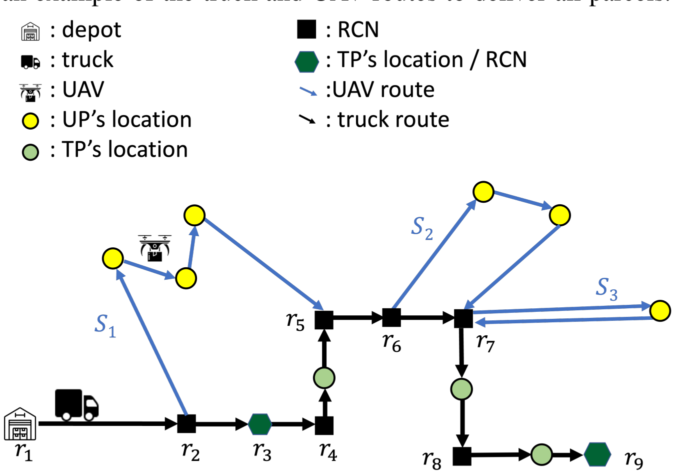

Fig. 1. An example of the truck and UAV routes to deliver parcels.

To solve the UAV-truck cooperation delivery problem, we must find the UAV and truck routes to minimize the delivery time. First, we focus on the truck route. Assume the truck will go through some RCNs when it delivers the TPs. The RCNs can be found when the truck route is determined. We denote the RCNs with $\mathcal{R} = \{r_1, r_2, ..., r_e\}$, where r_1 represents the depot, and the last position truck arrives at is r_e. Each r_i in $\mathcal{R}$ will correspond to a timestamp t_i representing the truck departure time from the RCN r_i. And we denote these timestamps as a set $\mathcal{T} = \{t_1, t_2, ..., t_e\}$, where t_e is the time all UPs and TPs have been delivered and UAV has returned to the truck.

Second, we consider the UAV route. Because the UAV has capacity and power limitations, it cannot deliver all UPs in one flight. To serve all the UPs, the UAV is dispatched several times, and each dispatched corresponds to a UAV subroute. We denote all the subroutes as a set $\mathcal{S} = \{S_1, S_2, ..., S_q\}$ if the number of subroutes is q. A subroute is a route from the take-off node to deliver a set of UPs and return to the landing node. Notice that the take-off and landing nodes could be different. For subroute $\mathcal{S}_k$, the take-off node's index on $\mathcal{R}$ is denoted by o_k, and the landing node's index on $\mathcal{R}$ is denoted by l_k. Furthermore, the UAV flying distance of each subroute $\mathcal{S}_k$ is denoted by d_k^u.

Since we assume the UAV arrives at the landing node after the truck, the truck waiting time for subroute $\mathcal{S}_k$ to be served is denoted by a_k, which can be expressed as:

$$a_k = \left(\frac{d_k^u}{v_u}\right) - \left(\frac{\sum_{i=o_k}^{l_k-1} d(r_i, r_{i+1})}{v_t}\right), \tag{1}$$

where $d(r_i, r_{i+1})$ is the truck travels distance between r_i and r_{i+1}. For example, considering $\mathcal{S}_1$ in Fig. 1. The time for the UAV to fly the subroute is $\frac{d_1^u}{v_u}$. At the same time, the truck departs from the UAV take-off node r_2 and goes to the landing node r_5, and the time it takes is $\frac{\sum_{i=2}^{4} d(r_i, r_{i+1})}{v_t}$. So the truck waiting time is $\frac{d_1^u}{v_u}$ minus $\frac{\sum_{i=2}^{4} d(r_i, r_{i+1})}{v_t}$.

When we calculate t_i, we need to consider all the waiting time at this landing node. Let $z_{i,k} \in \{0, 1\}$ denote whether r_i is the landing node of UAV for subroute $\mathcal{S}_k$. Since an RCN may be the landing node of multiple subroutes, we treat them as different RCNs to represent our algorithm easily. For example, in Fig. 1, the landing nodes of $\mathcal{S}_2$ and $\mathcal{S}_3$ are denoted as r_7 and r_8, respectively. And their corresponding timestamps are t_7 and t_8, respectively. We can express timestamp t_i as:

$$t_i = \begin{cases} 0 & i = 1 \\ t_{i-1} + \frac{d(r_{i-1}, r_i)}{v_t} + \sum_{k=1}^{|S|} z_{i,k} a_k & \text{otherwise,} \end{cases} \tag{2}$$

where $i = 1$ is the depot, and $\frac{d(r_{i-1}, r_i)}{v_t}$ is the time the truck needs to travel from the previous RCN to the current RCN. Note that each RCN can be the landing node of at most one subroute.

B. Problem Formulation

Here, we aim to minimize our system's total parcel delivery time. In other words, we want to minimize t_e. The routing problem can be formulated as follows:

$$\begin{aligned} \min \quad & t_e && \text{(3a)} \\ \text{s.t.} \quad & d_k^u \leq d_{max}, \forall \mathcal{S}_k \in \mathcal{S} && \text{(3b)} \\ & a_k \geq 0, \forall \mathcal{S}_k \in \mathcal{S} && \text{(3c)} \end{aligned}$$

Equation (3b) shows that UAV flying distance in each subroute should not exceed d_{max}. Constraint (3c) ensures the truck arrives at rendezvous nodes before the UAV.

To confirm our problem is an NP-hard problem, we assume that the UAV can always perfectly rendezvous with the truck, which means the truck will not need to wait for the UAV. The TP locations in our problem can be mapped to the cities in TSP. The objective of TSP is to find the minimum cost. Our problem (3a) aims to minimize the total parcel delivery time, equivalent to the minimum cost of TSP. Thus, TSP is a special case of our problem. Since TSP is NP-hard, our problem is also an NP-hard problem.

III. Minimize Parcels Delivery Time (MPDT) Algorithm

This section describes the proposed algorithm to minimize the total delivery time of TPs and UPs. Several RCNs are distributed on the roads for the UAV to take off or land on the truck. The problem involves finding the truck route to serve all TPs and UAV subroutes to serve all UPs. Here, we propose a three-stage heuristic algorithm called minimize parcel delivery time (MPDT).

In the first stage, we solve the truck routing problem. We first determine the all-pairs shortest paths of all TPs and use a heuristic algorithm to find the minimum traveling distance for all the TPs. In the second stage, we cluster all UPs by considering the distance between UPs first and then the UAV payload limits. We construct every UAV subroute with the clustering result in the final stage. The subroute consists of a route of parcels' locations, a take-off node, and a landing node. We first determine the order of choosing each cluster's take-off and landing nodes. Then, we assign each cluster a range of RCNs and select the take-off and landing nodes from this range.

A. Truck Route Construction

In this stage, we need to find the TPs traveling order, which is similar to TSP, except our truck does not need to return to its starting node. Since the TSP is NP-hard, we propose the following heuristic algorithm. First, we determine the distance between all pairs of shortest paths of TPs. We can find the all-pairs shortest paths by using the Floyd-Warshall algorithm [14, Chapter 25]. The time complexity of the Floyd-Warshall algorithm is $O(|\mathcal{P}^t|^3)$, where $|\mathcal{P}^t|$ is the number of TPs. After this algorithm, we have the shortest path and the corresponding distance between every pair of TPs. Then, we can use a heuristic method described in [14, Chapter 35] to solve the TSP problem in polynomial time. The time complexity of the heuristic algorithm is $O(|\mathcal{P}^t|^2)$. When the truck delivers the TPs order is determined, it will pass through several RCNs, which is the set $\mathcal{R}$.

B. UPs Clustering

Since the UAV cannot serve all UPs simultaneously, we need to cluster UPs. The UAV will serve exactly one cluster in one flight. We propose a two-step clustering algorithm. In the first step, we cluster the UPs by distance. In the second step, we divide these clusters into sub-clusters by the UP's weight to satisfy the UAV payload constraint. We try to separate as few clusters as possible after the two steps. When delivering in rural environments, some UPs may be far from the road, which makes it take a long time for the UAV to take off from the truck and back to the truck after delivery. Fewer clusters mean the UAV can spend less time on round trips.

To avoid the UAV spending too much time traveling between UPs, in the first step, we cluster the UPs by considering the distance between each other. We use k-means++ [15] to cluster the UPs. Besides, we estimate the number of clusters k based on the experiment result, which will show in Section IV. The k-means++ is a variant based on k-means, which has improved speed and accuracy in clustering. This algorithm provides an $O(log\ k)$-competitive with the optimal clustering. Unlike k-means chooses initial centroids randomly, k-means++ tries to find k initial centroids, which keeps the distance between each other as far as possible.

After the first step, we have k clusters clustered by the distance. Due to UAV payload limitations, in the second step, we divide every cluster into sub-clusters, such that the sum of the UPs' weight in each sub-cluster will not exceed the UAV payload limit. Dividing a cluster into sub-clusters is very similar to the bin-packing problem. The problem becomes minimizing the number of clusters we divide. To solve it, we apply the First-fit-decreasing (FFD) algorithm [16]. After applying FFD for all clusters, assume we have q sub-clusters, and $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_q\}$. Furthermore, each cluster in $\mathcal{C}$ satisfies the weight constraint.

After determining which UPs are inside which clusters, we look for the UPs' delivery order for each cluster. Since finding a delivery order is analogous to TSP, we use the nearest neighbor algorithm (NNA) [17] to solve the TSP problem, which is efficient for small-size instances. After applying NNA, the visit order of the UPs' location in cluster $\mathcal{C}_i$ is denoted as set $\{c_i^1, c_i^2, ..., c_i^f\}$, where c_i^f is the last UP's location of $\mathcal{C}_i$. The UAV's flying direction inside the cluster has not been decided. K-means++ has an $O(h^2)$ time complexity, and FFD's time complexity is $O(h\ log\ h)$, where h is the number of UPs. The time complexity for NNA is $O(h^2)$. Therefore, the time complexity of the two-step clustering algorithm is $O(h^2)$.

C. UAV Subroutes Construction

In stage 3, for each cluster $\mathcal{C}_i$, we form the UAV subroute $\mathcal{S}_i$ with a take-off node o_i and a landing node l_i. We will first decide the order of the clusters to choose their take-off and landing nodes. Let y_i denote the centroid of cluster $\mathcal{C}_i$ in $\mathcal{C}$. Let g_i be the index of RCN in $\mathcal{R}$ closest to y_i and $\mathcal{G} = \{g_1, ..., g_q\}$. If the truck visits the RCN g_i before RCN g_j, the parcels of cluster $\mathcal{C}_i$ are sent by the UAV before $\mathcal{C}_j$. Therefore, $\mathcal{C}_i$ will find its subroute before $\mathcal{C}_j$. For ease of description, we relabel

Algorithm 1 : Choosing Take-off And Landing Nodes

Input: Set of clusters $\mathcal{C}$, Set of RCNs $\mathcal{G}$ that closest to the centroid of each cluster
Output: Timestamp t_e
1: Initialize lb_1 to 1, and rb_q to e.
2: $\mathcal{C}' \leftarrow \emptyset$
3: **for** $\mathcal{C}_i$ in $\mathcal{C}$ **do** $\triangleright$ Construct subroutes from i = 1 to $|\mathcal{C}|$.
4: Set lb_i as l_{i-1} and rb_i as $\lfloor \frac{g_i+g_{i+1}}{2} \rfloor$'s index.
5: $a_i \leftarrow \infty$ $\triangleright$ a_i is shortest waiting time.
6: **for** j = lb_i to rb_i **do**
7: **for** k = j to rb_i **do**
8: **if** $a_i > WT^1$ **then**
9: Update a_i to WT^1.
10: Record j as take-off node's index o_i.
11: Record k as landing node's index l_i.
12: **if** $a_i > WT^2$ **then**
13: Update a_i to WT^2.
14: Record j as take-off node's index o_i.
15: Record k as landing node's index l_i.
16: **if** $a_i \neq \infty$ **then**
17: Construct $\mathcal{S}_i$ with $\mathcal{C}_i$ and the recorded take-off and landing nodes.
18: $\mathcal{C}'$ = $\mathcal{C}' \cup \mathcal{C}_i$
19: **else**
20: Divide $\mathcal{C}_i$ evenly into two sub-clusters.
21: Divide the choice range evenly into two parts.
22: For each sub-clusters, repeat lines 6 - 22.
23: Update the timestamp t_α sequentially according to equation (2), where $1 \leq \alpha \leq e$.
24: $\mathcal{C}$ = $\mathcal{C}'$
25: **while** t_e is shorten **do**
26: **for** $\mathcal{C}_i$ in $\mathcal{C}$ **do** $\triangleright$ Construct subroutes from i = 1 to $|\mathcal{C}|$.
27: **if** $lb_i^{'} < lb_i$ or $rb_j^{'} > rb_i$ **then**
28: Set lb_i as $lb_i^{'}$, rb_i as $rb_i^{'}$.
29: Repeat lines 6 - 15.
30: Update the timestamp t_α sequentially according to equation (2), where $1 \leq \alpha \leq e$.
31: **return** t_e

$\mathcal{C}_i$ and its corresponding g_i with the truck visiting order, for $i = 1, \ldots, q$. After that, $\mathcal{C}_1$ is the first cluster that can choose the take-off and landing nodes to form a subroute, $\mathcal{C}_2$ is the second, and so on. To guarantee all clusters have more than one RCN to choose from as their take-off and landing nodes, we assign a range of RCNs for each cluster to select its take-off and landing nodes. For any cluster $\mathcal{C}_i$, we set a left bound (lb_i) and a right bound (rb_i) on $\mathcal{R}$ as the selection range. For cluster $\mathcal{C}_i$, lb_i is the index of the landing node l_{i-1} and rb_i = $\lfloor \frac{g_i+g_{i+1}}{2} \rfloor$. We set lb_1 of $\mathcal{C}_1$ to 1 and rb_1 = $\lfloor \frac{g_1+g_2}{2} \rfloor$. Assume $\mathcal{S}_1$'s take-off and landing nodes are r_1 and r_3, respectively. So $\mathcal{C}_2$'s left bound lb_2 is 3, and rb_2 = $\lfloor \frac{g_2+g_3}{2} \rfloor$, which is 7. Thus, $\mathcal{C}_2$ can choose take-off and landing nodes from the set $\{r_3, r_4, r_5, r_6, r_7\}$. For the last cluster $\mathcal{C}_q$, the rb_q is set to e, which is the last index of RCNs in $\mathcal{R}$.

In Algorithm 1, we construct subroute $\mathcal{S}_i$ with the shortest truck waiting time for each cluster $\mathcal{C}_i$. We can find the shortest truck waiting time a_i in lines 6-15 for an exhaustive search o_i and l_i between the left bound lb_i and right bound rb_i. Here, o_i is set to j in line 6, and l_i is set to k in line 7, respectively. Note that we can deliver UPs from c_i^1 to c_i^f, the corresponding waiting time is WT^1 (lines 8-11), or from c_i^f to c_i^1, and the corresponding waiting time is WT^2 (lines 12-15). The waiting time can be computed using equation (1). After finding the shortest waiting time of a subroute, the corresponding indices of take-off and landing nodes are assigned to o_i and l_i, respectively.

The cluster will split into two sub-clusters if a subroute distance exceeds the UAV's maximum flying distance. Since clusters may increase during iterations, we use a new set $\mathcal{C}'$ to store the clusters in line 18. In line 16, if there is a feasible subroute that meets both constraints (3b) and (3c), we can form the subroute S_i with o_i and l_i and insert this cluster into $\mathcal{C}'$. In line 19, if there is no feasible subroute for $\mathcal{C}_i$, the cluster and its corresponding range from lb_i to rb_i will be evenly divided into two sub-clusters and two sub-ranges, respectively. Each sub-cluster will keep splitting until we can find a feasible solution for the sub-cluster. Once the solution is found, this sub-cluster will add to set $\mathcal{C}'$ in line 18. After completing the subroute S_i, we continue to build subroute S_{i+1} for next cluster $\mathcal{C}_{i+1}$. When we find the UAV subroutes for all UPs, we can compute the total delivery time t_e by equation (2) in line 23 and update set $\mathcal{C}$ to $\mathcal{C}'$ in line 24.

Furthermore, we can shorten the waiting time a_i of each subroute $\mathcal{S}_i$ after all the clusters choose their take-off and landing nodes. First, we reset the lb_i and rb_i of cluster $\mathcal{C}_i$ in $\mathcal{C}$ to the landing node's index of subroute $\mathcal{S}_{i-1}$ and the take-off node's index of $\mathcal{S}_{i+1}$, respectively. Let $lb_i^{'}$ = l_{i-1} be the new left bound and $rb_i^{'}$ = o_{i+1} be the new right bound. If $lb_i^{'} < lb_i$ or $rb_i^{'} > rb_i$, it means we have more possible combinations of take-off and landing nodes for $\mathcal{C}_i$ than original ranges between lb_i and rb_i. Thus, we can repeat lines 6-15 to search for a shorter waiting time a_i and update $\mathcal{S}_i$. Once $\mathcal{C}_i$ chooses new take-off or landing nodes, $\mathcal{C}_{i-1}$'s right bound or $\mathcal{C}_{i+1}$'s left bound may change again, respectively. For example, if we update $\mathcal{S}_i$'s take-off node, $\mathcal{C}_{1-1}$'s right bound may be changed again. In line 25, we will repeat the process until the last RCN's timestamp t_e is not changed. We only need to calculate those new subroutes generated by the new bounds to save the calculation time.

After we constructed the truck route and the UAV subroutes, all UPs and TPs can be served. The delivery completion time is the output of Algorithm 1. The time complexity in Lines 3-22 is $O(e^2)$, where e is the number of RCNs in $\mathcal{R}$. Since the number of executions of line 25 is usually less than 10 in our experiments, and the time complexity in Lines 26-29 is also $O(e^2)$. The time complexity of Algorithm 1 is $O(e^2)$. Therefore, the time complexity of our proposed algorithm MPDT is $O(|\mathcal{P}^t|^3)$ + $O(h^2)$ + $O(e^2)$ = $O(|\mathcal{P}^t|^3 + h^2 + e^2)$.

IV. Performance Evaluation

A. Simulation Settings

According to Amazon Prime Air UAV [18], the UAV in our experiments can deliver a maximum payload of 2.3 kg, and its endurance distance is 16 km. As for the truck, it can travel up to 400 km with full fuel. Thus, the truck can be assumed to finish daily parcel delivery without refueling. Our experimental environment is an 8 km by 8 km area. The road design references the rural terrain of the United States, and RCNs are randomly deployed on the road. The distance between two neighboring RCNs is a multiple of 200 m, and the number of RCNs is set to 225. The total number of parcels is 90. The ratio of TP sets to 30%, and UP sets to 70%. TPs are distributed randomly on the road. UPs are randomly deployed on the map, and the weight of each UP also follows the normal distribution, with μ being 0.9 kg and σ being 0.45. In stage 2 of MPDT, the number of clusters k in k-means++ [15] is set to 7 according to the elbow method. This number k is related to the UAV endurance distance and the environment size. The velocities of the UAV and truck are set to 50 km/hr and 30 km/hr, respectively.

Since no articles consider the same problem as ours, we modify some existing algorithms [10], [19] as the baselines. First, we assume all algorithms can obtain the truck route through stage 1 of MPDT. Then, we compare the performance of MPDT with three different algorithms: Randomized Iterative best insertion (RISE) [19], Simulated Annealing (SA) [10], and Greedy method. We first determine the UAV delivery order of all UPs to construct the UAV subroutes for the Greedy algorithm. We denote the RCN's index closest to UP p_i^u as ω_i, and $\Omega = \{\omega_1, ..., \omega_h\}$, where h is the number of UPs. If the truck visits the RCN ω_i before RCN ω_j, the parcel p_i^u is sent by UAV before p_j^u. UAV departs from the closest RCN to the first UP in the UAV delivery order at the beginning. It will try to deliver the most UPs in the delivery order and land on the last delivered UP's closest RCN as long as the subroute satisfies the constraints (3b) and (3c). After completing the first subroute, we will take off from the RCN nearest the next undelivered UP and construct the second subroute similarly. This process will continue until all UPs are served.

For RISE and SA algorithms, we use the UAV subroutes of the Greedy algorithm as the initial solution and then improve the result of the UAV subroutes. We remove and insert multiple UPs from the UAV subroutes for the RISE algorithm. The UP can either insert into an existing subroute or let the UAV build a new subroute to serve it. The route will be updated if the result is better than the current result. The algorithm will terminate until the above updating process has reached the iteration limit. For the SA algorithm, we remove and insert one UP from the UAV subroutes, and there are three modes for insertion. The first two modes are the same as RISE. The third mode is to exchange the delivery order of one UP with another UP. The route will be updated if the result is better than the current result.

B. Simulation Results

In this section, we compare the performance of MPDT with the baselines. The performance is an average of 35 simulations. In Fig. 2, we show the task competition time for different numbers of parcels. MPDT can achieve the shortest task competition time compared to other algorithms. We can see that the task competition time in all algorithms increases when more parcels are in the system. However, since all the algorithms have the same truck route, we use total truck waiting time as our performance metric in the rest of our simulations. The longer the truck waits, the less efficient the algorithm is.

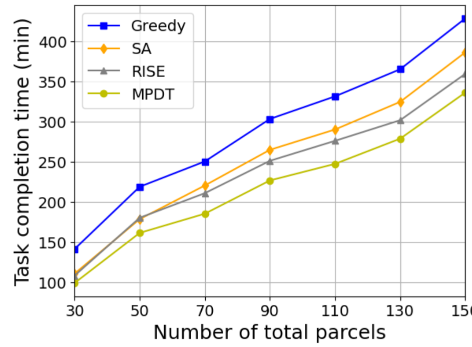

Fig. 2. Task completion time versus the number of parcels.

In Fig. 3, we show the truck waiting time for different numbers of parcels. We can see that the truck waiting time in all algorithms increases when more parcels are in the system. MPDT can achieve the shortest truck waiting time compared to other algorithms. The Greedy algorithm performs worst since the UAV only returns to the RCN closest to the currently delivered UP. When the UAV only delivers one UP, its take-off and landing nodes are the same. And this makes the truck wait for a long time. The SA algorithm keeps searching for a better solution by removing and inserting. However, this algorithm needs many iterations to get a good solution. The result of the RISE algorithm is better than the SA algorithm. Unlike the SA algorithm, RISE can remove at most three UPs simultaneously, so more possibilities can be considered when inserting, and it performs better.

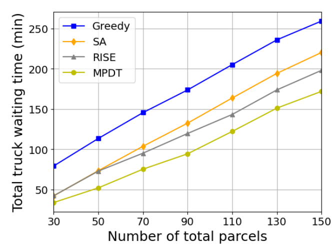

Fig. 3. Total truck waiting time versus the number of parcels.

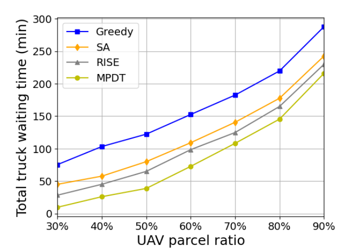

Fig. 4. Total truck waiting time versus UAV parcel ratio.

Fig. 4 shows the truck waiting time concerning different parcel ratios. With a fixed number of parcels, we can see that the truck waiting time becomes longer when the UAV parcel ratio increases because more UPs need to be delivered. When the UAV parcel ratio reaches 90%, the number of TPs decreases, so the average truck path is shorter. Consequently, the number of RCNs that the truck will pass through decreases. With fewer RCNs, each cluster has fewer choices to construct the subroute. So, the total waiting time of all algorithms is close.

In Fig. 5, we show the truck waiting time concerning the

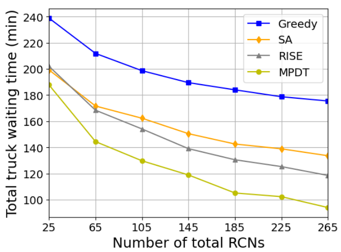

Fig. 5. Total truck waiting time versus number of RCNs.

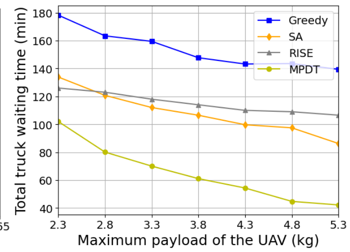

Fig. 6. Total truck waiting time versus maximum payload of the UAV.

number of RCNs. MPDT can achieve the shortest truck waiting time compared to other algorithms. The Greedy method has the worst performance. Greedy restricts the take-off and landing nodes to the closest RCN that the UAV serves, so the increase in the number of RCNs will not make more choices for taking off and landing. The performance of the RISE algorithm is better than SA since it can remove multiple UPs at once. When the number of RCNs is small, each algorithm's total truck waiting time has little difference because the number of RCNs that can be selected as the landing node and the take-off node is limited. As the number of RCNs increases, the gap between MPDT and other algorithms will gradually increase. With more RCNs, MPDT has more choices of take-off and landing nodes for each cluster, so the truck waiting time decreases.

In Fig. 6, we show the truck waiting time for the different maximum payloads of the UAV. The maximum payload varies from 2.3 kg to 5.3 kg. RISE's performance changes slightly better when the UAV's maximum payload increases. The reason is that although the UAV can deliver more UPs per subroute, the flight path inside the cluster is not planned, so the increase in the UAV flying distance makes the truck wait for more time. Unlike RISE, SA can exchange the delivery order of one UP with another UP. By doing so, the flight distance inside the clusters may be shortened, so SA's truck waiting time is shorter than RISE's when the maximum payload of the UAV increases. Compared with other algorithms, the truck waiting time of MPDT decreases faster as the maximum payload of the UAV increases. At the beginning of MPDT, we used k-means++ to cluster UPs, so the close UPs are assigned to the same cluster. This greatly reduces the possibility that the UAV may have to fly far away to deliver the next UP. Therefore, when the maximum payload of the UAV increases, the UAV can deliver more UPs at one time and satisfy the maximum flight distance limit.

V. Conclusion

This work studies the UAV-assisted truck for parcel delivery. The UAV can deliver parcels to and from the truck. We proposed a three-stage algorithm to minimize the delivery time subject to UAV endurance distance and payload limit constraints. First, we treat the truck delivering TPs as a TSP and determine the delivery order by a heuristic algorithm. Second, we cluster UPs by considering their distances and weights. Third, we construct the UAV subroutes. We determine the take-off and landing nodes of each subroute. Finally, the simulation results show that our algorithm can achieve the shortest truck waiting time compared to the baselines.

References

[1] Z. Wu, Z. Yang, C. Yang, J. Lin, Y. Liu, and X. Chen, "Joint deployment and trajectory optimization in UAV-assisted vehicular edge computing networks," *Journal of Communications and Networks*, vol. 24, no. 1, pp. 47–58, 2022.

[2] J. Li, H. Zhao, H. Wang, F. Gu, J. Wei, H. Yin, and B. Ren, "Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1464–1475, 2020.

[3] S. Vásquez, G. Angulo, and M. Klapp, "An exact solution method for the TSP with drone based on decomposition," *Computers & Operations Research*, vol. 127, pp. 1–12, Mar. 2021.

[4] S. Zhang, C. Markos, and J. J. Q. Yu, "Autonomous vehicle intelligent system: Joint ride-sharing and parcel delivery strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18 466–18 477, 2022.

[5] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.

[6] S. Ito, K. Akaiwa, Y. Funabashi, H. Nishikawa, J. Kong, I. Taniguchi, and H. Tomiyama, "Load and wind aware routing of delivery drones," *Drones*, vol. 6, pp. 1–14, Feb. 2022.

[7] Y. Zhu and S. Wang, "Aerial data collection with coordinated UAV and truck route planning in wireless sensor network," in *IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, 2021, pp. 1–6.

[8] M. Ha, Y. Deville, D. Pham, and M. Hà, "A hybrid genetic algorithm for the traveling salesman problem with drone," *Journal of Heuristics*, vol. 26, pp. 219–247, Apr. 2020.

[9] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5772–5782, 2021.

[10] Y. Liu, Z. Liu, J. Shi, G. Wu, and W. Pedrycz, "Two-echelon routing problem for parcel delivery by cooperated truck and drone," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 12, pp. 7450–7465, 2021.

[11] X. Bai, Y. Ye, B. Zhang, and S. S. Ge, "Efficient package delivery task assignment for truck and high capacity drone," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2023.

[12] C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, Jan. 2020.

[13] B. Mahmoudi and K. Eshghi, "Energy-constrained multi-visit TSP with multiple drones considering non-customer rendezvous locations," *Expert Systems with Applications*, vol. 210, pp. 1–20, Aug. 2022.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, Mass: MIT Press, 2009.

[15] D. Arthur and S. Vassilvitskii, "K-Means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, no. 1, 2007, pp. 1027–1035.

[16] G. Dósa, R. Li, X. Han, and Z. Tuza, "Tight absolute bound for first fit decreasing bin-packing," *Theoretical Computer Science*, vol. 510, pp. 13–61, 2013.

[17] G. Kizilates-Evin and F. Nuriyeva, "On the nearest neighbor algorithms for the traveling salesman problem," *Advances in Intelligent Systems and Computing*, vol. 225, pp. 111–118, Jan. 2013.

[18] S. Jung and H. Kim, "Analysis of amazon prime air UAV delivery service," *Journal of Knowledge Information Technology and Systems*, vol. 12, pp. 253–266, Apr. 2017.

[19] Z. Wang, B. Zhang, and C. Li, "Joint path planning of truck and drones for mobile crowdsensing: Model and algorithm," in *IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, 2021, pp. 1–6.