Planning UAV Trajectory for Multi-Commodity Package Pickup and Delivery

Yi-Fu Chen[†], Jang-Ping Sheu[†], and Jagadeesha R Bhat[‡]
[†]Department of Computer Science, National Tsing Hua University, Taiwan
[‡]Indian Institute of Information Technology Dharwad, India
E-mail: s110062568@m110.nthu.edu.tw, sheujp@cs.nthu.edu.tw, and jagadeesha@iiitdwd.ac.in

***Abstract*—The use of UAVs for logistics services has become a highly regarded application in recent years. This paper studies the package pickup and delivery problem with multi-commodity and multi-visits. Due to the limited load, the UAV has to operate within the load limit when performing package delivery services. In addition, we allow the UAV to visit a location multiple times during the mission. Our objective is to minimize the total flying distance of the UAV. Since the problem is NP-hard, we propose a two-phase heuristic algorithm to solve this problem. First, the trajectory of the UAV to pick up or deliver packages is constructed using a greedy algorithm. Second, we optimize the previously built trajectory to obtain a shorter flying distance for the UAV. The simulation results show that the proposed algorithm outperforms the baselines regarding total flight distance and execution time.**

***Index Terms*—Unmanned aerial vehicle, pickup and delivery problem, trajectory planning.**

I. Introduction

In recent years, Unmanned Aerial Vehicles (UAVs) have been widely used in various applications such as environmental monitoring, search and rescue [1], data sensing [2], and package delivery [3]. UAVs are more mobile, agile, and flexible than traditional vehicles in package delivery to remote and difficult-to-reach areas. UAVs do not have terrain restrictions, such as crossing rivers or hills. In addition, the labor costs associated with traditional vehicles in the package delivery service are much higher. One advantage of UAVs is that they can operate autonomously or remotely, allowing efficient and cost-effective operations. Their small size and maneuverability enable them to navigate tight spaces and dense environments, providing valuable insights and services in areas where human access is limited or impractical. With the booming development of e-commerce, UAVs have become increasingly important to the logistics industry and play an important role in modern package delivery services. In recent trends, Google and Amazon have used UAVs for commercial package delivery services.

Recently, many researchers have investigated the package delivery problem, which is also called the Pickup and Delivery Problem (PDP) [4], [5]. The PDP aims to determine the optimal route for the vehicle to transport packages or passengers from pickup to delivery locations. Furthermore, because of the lightweight nature of UAVs, they have limited load capacities. UAVs should keep the load within a safe range. Otherwise, the package delivery mission may fail. In [6], the authors considered a package delivery system that cooperates with trucks and UAVs. This work assumed that the UAV performs only one delivery task on each flight to meet its capacity constraint. However, it results in longer delivery time and energy wastage. We envision that there is no need to impose restrictions on the number of delivery tasks per flight, provided the load capacity of the UAV is not exceeded. This approach is better aligned with the needs of a modern package delivery system.

Several works have investigated the package delivery problem without restricting the number of delivery tasks [7]-[12]. In [7], the authors introduced a Selective Pick-up and Delivery Problem (SPDP), a variant of the PDP. Instead of visiting all pickup locations, the vehicle can select some of them to pick up packages in this scenario. The objective was to minimize the travel cost of the vehicle under load capacity constraints. The authors in [8] considered a UAV-assisted package delivery system with the restriction of a no-fly zone. In [9] and [10], the authors investigated the PDP with one commodity, and the commodity needed at the delivery location could be collected from any pickup location. Moreover, the multi-commodity PDP was considered in [11] and [12]. Except for multi-commodity, the authors also assumed that the pickup and delivery locations in this scenario are paired, meaning each request has a pair of source and destination locations.

Most previous work assumed only one type of commodity in their systems, and each location could only be served precisely once. However, in a realistic job, multiple visits to a location to meet its demands can potentially lead to cost savings in logistics. In addition, the realm of package delivery services typically encompasses a variety of package types. The proposed work differs from the existing research in the following sense: this work considers a single UAV for multi-commodity package delivery services. The UAV will first pick up the packages at the source location and deliver these packages to the corresponding destination locations. During flight, the UAV can collect packages from different locations at the source if the load limit does not exceed its capacity. This is one of the uniqueness of the proposed problem compared to the research in [10]-[16]. In addition,

This work was supported in part by the National Science and Technology Council, Taiwan, under Grants 113-2221-E-007-137 and 112-2221-E-007-046-MY3.

2025 IEEE Wireless Communications and Networking Conference (WCNC) | 979-8-3503-6836-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/WCNC61545.2025.10978696

multiple visits are allowed to the same location, and a location can request package pickup or delivery, which is a distinct feature in the proposed problem scenario. Our main objective is to determine the minimum distance trajectory of the UAV under the load constraint throughout the mission.

This paper proposes a two-phase heuristic algorithm to minimize the trajectory of the UAV. First, we use a greedy-based algorithm to construct a UAV trajectory to pick up or deliver packages, which will satisfy all package-delivery requests. Since the UAV has a load limit, we must consider the distance and weight of the package when finding a UAV trajectory. After constructing the trajectory, we divide it into multiple sub-trajectories. Each sub-trajectory consists of two source locations and multiple destination locations, which are located within the two source locations. By fixing two source locations as the starting and end points, we need to find a shortest trajectory for the destination locations, essentially a Traveling Salesperson Problem (TSP). We can refine it through a TSP approximation algorithm [13], shortening the package delivery time.

The main contribution of our work can be summarized as follows. Firstly, we propose a new and realistic problem in autonomous logistic scenarios, as applicable to PDP with multi-commodity and multi-visits. A destination location might have some package to return/exchange and act as a source. Second, we design a heuristic algorithm to determine the trajectory of the UAV under the load constraint. Here, the trajectory of the UAV to pick up or deliver packages is constructed using a greedy algorithm. In addition, we optimize the previously built trajectory to obtain a shorter flight distance for the UAV. Third, the simulation results show that the proposed algorithm outperforms the baselines regarding total flying distance and execution time.

The rest of this paper is organized as follows. In Section II, we introduce our system model and the objective function of this paper. Section III describes the proposed algorithm. Then, Section IV shows the simulation result. Finally, the conclusion of this article is in Section V.

II. System Model and Problem Formulation

A. System Model

We consider a pickup and delivery network consisting of one UAV to deliver packages, such as mail or official documents, on campus. Our system has l locations and one depot l_0. Let $\mathcal{L} = \{1, 2, ..., l\}$ denote the set of l locations. We assume that some locations are sources that will deliver packages to some destination locations. The UAV must go to the sources to pick up packages and then deliver packages to the corresponding destinations. Note that a destination may receive multiple packages from different sources. Let $\mathcal{S} = \{s_1, s_2, ..., s_m\} \subseteq \mathcal{L}$ denote the set of m source (pickup) locations. For each source $i \in \mathcal{S}$ and $\mathcal{D}_i = \{d_{i,1}, d_{i,2}, ..., d_{i,k_i}\} \subseteq \mathcal{L}$ denotes the set of k_i destination (delivery) locations for source i. Let $\mathcal{D} = \bigcup_{i=1}^{m} \mathcal{D}_i$ represent all destinations. Note that a location can be a source and destination location and can be visited multiple times. In addition, the UAV will depart from the depot and deliver all packages. After completion of all tasks, the UAV will return

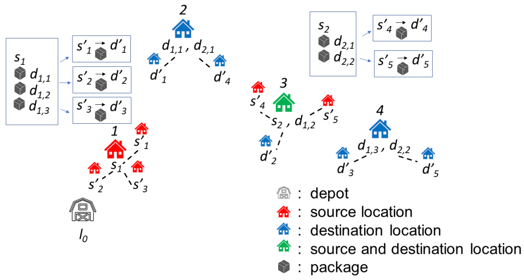

Fig. 1: A scenario of the UAV for multi-commodity package delivery service.

to the depot. Since we assume a UAV can recharge its battery at any location, we do not consider energy constraints in our problem. Thus, our work aims to find the shortest trajectory of the UAV to minimize energy consumption and complete its tasks.

As mentioned above, a source or destination can be visited multiple times to complete its package pickup or delivery. Moreover, if the UAV load does not exceed its limit, it can continue to pick up more packages from different sources until the limit is reached. Since a location may have multiple packages that need to be picked up or dropped off, each package has its source and destination location. By adding dummy locations, the package requests from the same location can be divided into multiple pairs of one-to-one source and destination locations. Therefore, the original problem can be transformed into the problem of a single visit to each dummy location.

An example of a UAV to pick up and deliver packages is shown in Fig. 1. We assume s_1 and s_2 are 1 and 3, respectively. The destinations of s_1 are $\mathcal{D}_1 = \{d_{1,1}, d_{1,2}, d_{1,3}\} = \{2, 3, 4\}$. The destinations of s_2 are $\mathcal{D}_3 = \{d_{2,1}, d_{2,2}\} = \{2, 4\}$. Since the source location 1 needs to deliver three packages to three destination locations, we add three dummy source locations s'_1, s'_2, and s'_3 corresponding to the three dummy destinations d'_1, d'_2, and d'_3, respectively. Similarly, the source location 3 has two destinations, so we add two dummy source locations s'_4 and s'_5. On the other hand, two packages need to be dropped off at each destination, 2 and 4, so we add two dummy locations for each destination. Let d'_1 and d'_4 be dummy locations of destination 2 and d'_3 and d'_5 be dummy locations of destination 4. Note that location 3 serves as both the source and destination locations. Therefore, we use s'_4 and s'_5 as the dummy source locations and d'_2 as the dummy destination location of location 3.

After adding dummy locations, the set of source locations $\mathcal{S}$ can be rewritten as $\mathcal{S}' = \{s'_1, s'_2, ..., s'_{m'}\}$. For each source location, $s'_i \in \mathcal{S}'$, d'_i is its corresponding destination location, i.e., the source and destination locations of the package i, and w'_i is the weight of the package. Let $\mathcal{D}' = \{d'_1, d'_2, ..., d'_{m'}\}$ represent all destination locations and $\mathcal{W}' = \{w'_1, w'_2, ..., w'_{m'}\}$ be the set of package weights. Due to the addition of dummy locations, each location will be visited only once. Therefore, the total trajectory length of the UAV is $2m' + 2$, including the departure of the UAV from the depot and its final return to the depot. Note that each source location corresponds to only one destination location with a package. For the load limit, let y_i be the sum of the weight of the UAV when it is landed at location i, and the maximum load capacity of the UAV is denoted by W. Then, the load constraint of the UAV is $y_i \leq W$.

B. Problem Formulation

In our system, we aim to minimize the total flight distance of the UAV under the constraint of the limited load of the UAV to deliver all packages. Let $\pi = [\pi_1, \pi_2, ..., \pi_{2m'+2}]$ denote the trajectory of the UAV, where π_i represents the i-th visiting location in the trajectory. The distance between location i and location j is denoted as $dis_{i,j}$. Therefore, we can formulate our problem as follows:

$$\min_{\pi} \quad \sum_{i=1}^{2m'+1} dis_{\pi_i,\pi_{i+1}} \tag{1a}$$

$$\text{s.t.} \quad y_{\pi_i} + w'_{\pi_{i+1}} = y_{\pi_{i+1}}, \quad \forall \pi_{i+1} \in \mathcal{S}', \tag{1b}$$

$$y_{\pi_i} - w'_{\pi_{i+1}} = y_{\pi_{i+1}}, \quad \forall \pi_{i+1} \in \mathcal{D}', \tag{1c}$$

$$y_{\pi_i} \leq W, \quad \forall \pi_i \in \mathcal{S}' \cup \mathcal{D}', \tag{1d}$$

$$y_{\pi_1} = 0, \tag{1e}$$

$$\pi_1 = l_0, \tag{1f}$$

$$\pi_{2m'+2} = l_0. \tag{1g}$$

Constraints (1b) and (1c) ensure the current load of the UAV in each location. Constraints (1d) and (1e) ensure the UAV's load limit. Constraints (1f) and (1g) enforce that the UAV must depart from the depot l_0 and return to l_0.

Recall our problem. We consider a pickup and delivery network where the UAV aims to minimize the total flight distance under the load limit constraint. This problem coincides with the well-known TSP when the load of the UAV is large enough, and both aim to find the trajectory with the minimal total distance. Therefore, the TSP is a special case of our problem. Since the TSP has been proven to be an NP-hard problem, our problem is also an NP-hard problem.

III. Minimizing Packages Delivery Distance (MPDD) Algorithm

In this section, we describe the proposed trajectory optimization algorithm to minimize the total flying distance of the UAV in the package pickup and delivery system. This problem involves planning the UAV's trajectory to complete all package delivery tasks. Hence, we propose a two-phase heuristic algorithm to minimize the package delivery distance (MPDD) to solve this problem.

In the first phase, we use a greedy strategy to construct a UAV trajectory. After the first phase, we further reduce the trajectory distance in the next phase. In the second phase, we label all source locations s'_i in the trajectory π, including the return of the UAV to the depot l_0. Every two source locations can form a sub-trajectory. By fixing two source locations as the starting and end points, we aim to find the shortest trajectory that covers all locations, effectively turning the problem into a TSP. Afterward, we refine each sub-trajectory through a TSP approximation algorithm to get a better visiting order for the UAV.

A. UAV Trajectory Construction

We adopt a greedy method since finding an optimal trajectory for our problem is NP-hard. First, the UAV takes off from the depot l_0. After that, we iteratively assign the location with the highest fitness as the next location until all package delivery tasks are completed. Due to the UAV's limited load capacity, the trajectory can consider not only the distances among locations but also the package weights at each location. Therefore, we consider a fitness metric that includes distance and package weight.

Because we first add dummy locations to split the packages for the original locations with multiple packages, each dummy location will only handle one package. An original location may have pickup and delivery requirements. A dummy source location is a better choice for the UAV if it can drop off more packages simultaneously. Let $\delta_{s'_i}$ denote the package weight that can be picked up at the source location s'_i, and $\delta_{d'_i}$ denote the package weight that can be dropped off at the destination location d'_i. Since a dummy source location may have several dummy destinations in the same place, we can deliver packages to these dummy destinations simultaneously. Let $\mathcal{K}_i \subseteq \mathcal{D}'$ be the set of dummy destinations with the same locations as a source s'_i, and the corresponding packages have already been picked up. Therefore, the package weight to be dropped off at the dummy destinations can be added to the source s'_i and can be expressed as follows:

$$\delta_{s'_i} = w'_i + \sum_{j' \in \mathcal{K}_i} \delta_{j'}, \tag{2}$$

$$\delta_{d'_i} = w'_i. \tag{3}$$

In other words, when the UAV arrives at the source location s'_i, it can drop the package from the destination location d'_j at the same time. For example, if the source location s'_i has a destination location d'_j with zero distance, that is, $dis_{s'_i,d'_j} = 0$, and the package should be dropped off at the destination location d'_j after picking up, the package weight $\delta_{d'_j}$ must be added to the package weight of the source location $\delta_{s'_i}$.

To calculate location fitness, it is necessary to normalize $dis_{i',j'}$ and $\delta_{j'}$ to ensure fairness. To do this, we first compute the candidate location closest to the current location i' and the candidate location with the heaviest package weight among the candidates, which can be computed as in equations (4) and (5).

$$d_{min} = \min_{j' \in candidate} dis_{i',j'}, \tag{4}$$

$$w_{max} = \max_{j' \in candidate} \delta_{j'}. \tag{5}$$

$$f_{i',j'} = \alpha \cdot \frac{d_{min}}{dis_{i',j'}} + (1-\alpha) \cdot \frac{\delta_{j'}}{w_{max}}, \tag{6}$$

Then, the location fitness for a dummy location i' to its candidate location j' is calculated as the ratio of the minimum distance d_{min} to $dis_{i',j'}$, denoted as $\frac{d_{min}}{dis_{i',j'}}$. The weighted fitness of the package is calculated as the ratio of $\delta_{j'}$ to the maximum weight w_{max}, denoted $\frac{\delta_{j'}}{w_{max}}$. Note that each candidate location must satisfy the load capacity constraint.

As a result, when the UAV is at dummy location i', the fitness of each candidate location j' can be calculated as in (6), where $\delta_{j'}$ can be calculated with equation (2) or (3) depending on the location j' is the source or destination. In addition, α is a parameter of the two terms of the equation to adjust the balance between distance and package weight in the fitness calculation. Note that if there is some dummy source location j' with the same coordinates as the current location i', that is, $dis_{i',j'} = 0$, we select j' as the next location. If there is more than one dummy source location with the same coordinates as the current location, the source location with the heaviest package weight will be first picked up.

After evaluating the fitness of all possible candidates, we select the candidate with the greatest fitness as the next location, which can be formulated as:

$$j^* \leftarrow \underset{j' \in candidate}{\arg\max} \; f_{i',j'}. \tag{7}$$

When all package delivery requests are fulfilled, the depot l_0 is added to the trajectory as the UAV needs to return to it, thus obtaining the result of the first phase. Since both the source location set $\mathcal{S}'$ and the destination location set $\mathcal{D}'$ have sizes m', the total time complexity of the first phase algorithm is $O(m')$.

B. Trajectory Refinement

Algorithm 1 Trajectory Refinement Algorithm

Input: UAV's trajectory π, the indices of source locations $s[]$
Output: UAV's trajectory π'

$i \leftarrow 1$
while $i < m' + 1$ **do**
  if $s[i+1] - s[i] > 2$ **then**
    for $j = i + 1$ to $m' + 1$ **do**
      $start \leftarrow s[i]$
      $end \leftarrow s[j]$
      Let $tra[]$ be an array of $\{\pi_i\}_{i=start}^{end}$
      Solve the TSP of $tra[]$ and obtain $new\_tra[]$
      Let d and d_{new} be the distances of $tra[]$ and $new\_tra[]$, respectively.
      if $d_{new} \leq d$ and $new\_tra[]$ is feasible **then**
        Replace $tra[]$ with $new\_tra[]$
      else
        $i \leftarrow \max(i + 1, j - 1)$
        break
      end if
    end for
  else
    $i \leftarrow i + 1$
  end if
end while
return UAV's trajectory π'

After the first phase, we will reorder the trajectory π to shorten the total flying distance of the UAV. Because we need to consider the trajectory's feasibility, i.e., the load capacity constraint, we cannot arbitrarily change the visiting order of the source and destination locations. For example, changing the visiting order of a source location can violate the UAV load capacity restriction. However, we can divide the first phase trajectory into multiple sub-trajectories based on the sequences of source locations and refine the sub-trajectories in order. Let $s[i]$ $(1 \leq i \leq m')$ denote the i-th source location on the trajectory π. Furthermore, $s[m' + 1]$ represents the return of the UAV to the depot l_0. Each sub-trajectory consists of two source locations ($s[i]$ and $s[i+1]$) serving as the starting and end points, respectively. The locations between the two source locations are destination locations only. Thus, the order of visit of destination locations between the two source locations can be changed while satisfying the load capacity constraint.

The second phase algorithm involves refining multiple sub-trajectories within the original trajectory π. The proposed algorithm will start from the first source location $s[1]$ in the trajectory and examine each sub-trajectory step by step until the entire trajectory is traversed. After identifying a sub-trajectory, the minimum distance of the sub-trajectory is found in the TSP. Since the TSP is an NP-hard problem, we use an approximation algorithm [13] to solve this problem in polynomial-time complexity. The algorithm first constructs a minimum spanning tree (MST) for all locations in the sub-trajectory with the starting point $s[i]$ set as the root. Subsequently, the desired visiting order can be obtained by traversing this MST using a preorder traversal approach. However, unlike traditional TSP, the sub-trajectory will end at the end point $s[i + 1]$ rather than returning to the starting point $s[i]$. After obtaining the new sub-trajectory, to ensure the feasibility of the trajectory, it is necessary to confirm whether the new sub-trajectory satisfies the load capacity constraint and whether the distance is shorter than the original one. If both conditions are met, replace the original sub-trajectory with the new sub-trajectory.

Additionally, we will keep the current starting point $s[i]$ fixed and change the endpoint $s[i + 1]$ to the next source location $s[i + 2]$. In other words, we can extend the length of the current sub-trajectory to enable the algorithm to find the TSP trajectory that includes more locations. This further shortens the overall trajectory distance. However, when we change the visit order of the source locations, we must satisfy the load capacity constraint. Therefore, extending the sub-trajectory should be terminated when the new sub-trajectory violates the load capacity constraint or has a longer distance than before. When we terminate extending a sub-trajectory at endpoint $s[i + k]$, we change the starting point $s[i]$ to $s[i + k - 1]$, for $1 \leq k \leq m' - i + 1$. This is because the sub-trajectory from $s[i]$ to $s[i+k-1]$ has already been refined, but the subsequent sub-trajectory starting from $s[i + k - 1]$ has not been refined yet. We continue to execute the trajectory refinement algorithm until the starting point is $s[m' + 1]$. The details of the second phase algorithm are summarized in Algorithm 1. An example of trajectory refinement is shown in Fig. 2. The algorithm starts from $s[1]$ to $s[2]$. Since there are no destination locations between them, this sub-trajectory is skipped. Then, it tries from $s[2]$ to $s[3]$ and

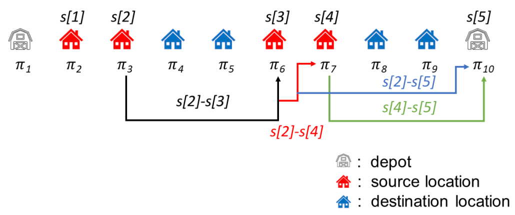

Fig. 2: An example of refining the trajectory.

checks whether the new sub-trajectory is shorter than the original one and the solution is feasible. If not, it attempts the sub-trajectory from $s[3]$ to $s[4]$. Conversely, if the new sub-trajectory is feasible and shorter than before, it tries the sub-trajectory from $s[2]$ to $s[4]$. Subsequently, if the sub-trajectory is feasible and shorter than before, we will try the sub-trajectory from $s[2]$ to $s[5]$. Otherwise, we will try the sub-trajectory from $s[4]$ to $s[5]$. The second phase algorithm refines the trajectory. The refined algorithm starts from the sub-trajectory between the first and second source locations, i.e., $s[1]$ and $s[2]$. It repeatedly attempts to enlarge a sub-trajectory to include more locations as long as possible. In line 2, it executes at most m' times. Additionally, since the length of the sub-trajectory can be up to $2m'$, the complexity of the TSP approximation algorithm used in line 8 is $O((m')^2)$. Therefore, the time complexity of the second phase algorithm is $O((m')^3)$. Finally, the time complexity of the proposed algorithm MPDD is $O(m') + O((m')^3) = O((m')^3)$.

IV. Simulations

A. Simulation Settings

In our experiments, 30 locations are randomly distributed over a 1000 m x 1000 m square area. The depot l_0 is in the bottom-left corner, at coordinates $(0, 0)$. We assume that only some locations require package delivery; that is, the number of source locations accounts for 20% of all locations. The number of destination locations is randomly selected within [3, 4, 5] for each source location. Besides, the weight of each package is set in the range of [0.6, 0.7, 0.8] kg. The maximum load capacity of the UAV is 3 kg [3]. In the first phase of MPDD, according to our experiments, the parameter α of the fitness equation is set to 0.7 to achieve relatively better performance.

For performance comparison, we implement three existing algorithms as baselines, which are the Genetic Algorithm (GA) [14], Ant Colony Optimization (ACO) [15], and Nearest Neighbor (NN). We assume that all algorithms will split multiple packages by adding dummy locations before determining the visiting order of the UAV. The NN is a greedy-based approach that involves adding the nearest location to the current trajectory at each step until all locations are included. Additionally, only feasible locations that satisfy the constraint are added when inserting a new location. Each simulation result is an average of 20 simulations.

B. Simulation Results

In Fig. 3, we show the total flight distance of the UAV with different numbers of locations. We can see that as the number of locations increases, the distance obtained by all algorithms increases. However, when the number of locations is small, the NN approach results in the longest distance, as once it goes to the nearest location, it may find subsequent locations too far and need longer backtracking. Similarly, ACO makes initial pheromone trails with poor choices, leading to premature suboptimal paths that cannot improve due to limited points. MPDD can find the shortest trajectory to achieve better performance when more locations are in the system. Although GA and ACO search for solutions through an iterative process, in GA, due to increased location mutations, swapping, or segment reversal of a path during the search may still lead to trajectories with longer distances.

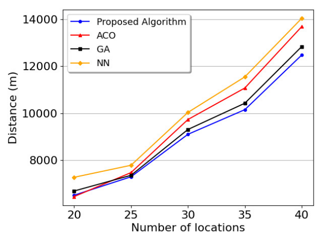

Fig. 3: Total flight distance vs. number of locations.

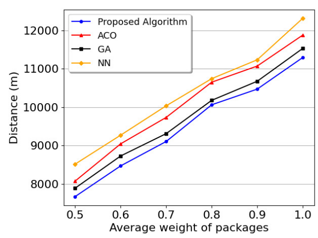

Fig. 4: Total flight distance vs. average weight of packages.

Fig. 4 shows the total flight distance for different package weights. In this experiment, the weight of each package is set at $\pm$ 0.1 kg of the average weight. The average weight of the package varies from 0.5 kg to 1 kg. We can observe that the flight distance of all algorithms increases as the package weight increases. This is because as the weight of the package increases, the number of packages that the UAV can carry decreases, requiring an increase in the number of flights. MPDD consistently outperforms other comparison algorithms in all cases, with the NN approach showing the worst performance. The reason is that the NN only considers the distance between locations and ignores the weight of the packages.

We show the total flight distance for different numbers of destination locations in Fig. 5. In this simulation, the number of destination locations for each source location is randomly chosen within a range of $\pm$ 1 from the average number of destination locations. As the number of destination locations increases, the UAV must spend more time and cover a longer distance to complete the tasks. When there are fewer destination locations, i.e., when each source location has to deliver a small number of packages, the performance differences among the other three algorithms are insignificant except for the NN approach. In contrast, when the number of destination locations increases, indicating an increase in the number of packages to be delivered for each source location, MPDD exhibits better performance, and GA and ACO performance decreases as the number of destination locations increases. This is because more locations must be planned on the trajectory, making it difficult for GA and ACO to find better results.

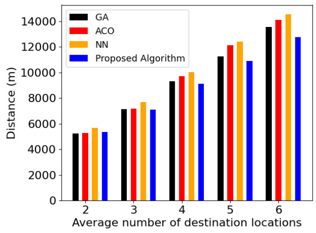

Fig. 5: Total flight distance vs. average number of destination locations.

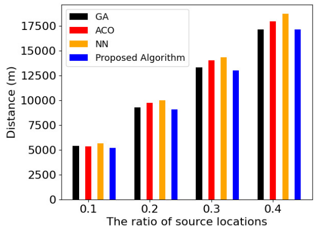

Fig. 6: Total flight distance vs. the ratio of source locations.

In Fig. 6, we show the total flight distance for different ratios of source locations. When the ratio of the source locations increases, it indicates an increase in both the source and destination locations in the system, thereby expanding the scope of UAV service. This leads to a longer total flight distance. We can observe that when the ratio of source location is small, the performance among all algorithms is similar. This is because fewer packages result in a less significant impact on flight distance regardless of the algorithm used for trajectory planning. However, when the ratio of the source location is high, each location may need to handle a large number of packages simultaneously. In this complex scenario, the performance of the NN approach is the worst. MPDD performs similarly to the GA in this case.

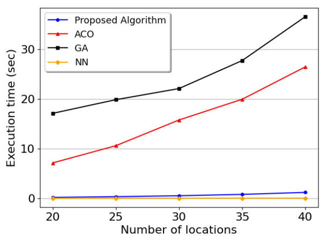

Fig. 7: Execution time vs. number of locations.

In Fig. 7, we compared the execution time of all algorithms for different numbers of locations. We can see that the execution time of all algorithms increases as the number of locations grows. The NN approach requires minimal execution time because it employs a greedy method to find the trajectory. The GA and the ACO require significant execution time as they iteratively update solutions to converge. As for MPDD, the main computational time is spent on solving the TSP. As the number of locations increases, the approximate algorithm used to solve the TSP requires more time, making the execution time slightly longer than that of the NN. Although MPDD shows performance comparable to that of GA and ACO in terms of distance in some cases, it significantly outperforms their execution time.

V. Conclusion

In this paper, we study the problem of using UAVs for package delivery services and minimize the total flying distance. We propose a two-phase heuristic algorithm to address this problem, which is subject to the load capacity constraint of the UAV. First, we construct a trajectory through a greedy-based algorithm. Second, a TSP approximation algorithm optimizes the previously built trajectory to find a shorter flying distance trajectory. The simulation results show that our proposed algorithm outperforms the other algorithms regarding total flying distance. In particular, in the scenario with a large number of package delivery requests, the proposed algorithm performs well and requires significantly less execution time compared to the baselines.

References

[1] M. Atif, R. Ahmad, W. Ahmad, L. Zhao, and J. J. P. C. Rodrigues, "UAV-assisted wireless localization for search and rescue," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3261-3272, 2021.

[2] S. Alfattani, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "Multi-UAV data collection framework for wireless sensor networks," in *2019 IEEE GLOBECOM*, Waikoloa, HI, USA, 2019, pp. 1-6.

[3] 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.

[4] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems—Part I: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, pp. 21–51, 2008.

[5] S.N.Parragh, "A survey on pickup and delivery problems—Part II: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, pp. 81–117, 2008.

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

[7] Y. Li, "A selective many-to-many pickup and delivery problem with handling cost in the omni-channel last-mile delivery," *IEEE Access*, vol. 10, pp. 111284-111296, 2022.

[8] W. Wen, K. Luo, L. Liu, Y. Zhang, and Y. Jia, "Joint trajectory and pickup design for UAV-assisted item delivery under no-fly zone constraints," *IEEE Tran.on Veh. Tech*, vol. 72, no. 2, pp. 2587-2592, 2023.

[9] C. Lei, and Y. Ouyang, "Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems," *Transportation Research Part B: Methodological*, vol. 109, pp. 90-109, 2018.

[10] M. Elhenawy, H. A. Rakha, Y. Bichiou, M. Masoud, S. Glaser, J. Pinnow, and A. Stohy, "A feasible solution for rebalancing large-scale bike sharing systems," *Sustainability*, vol. 13, pp. 1-19, 2021.

[11] T. Mulumba, and A. Diabat, "Optimization of the drone-assisted pickup and delivery problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 181, pp. 103377-103391, 2024.

[12] T. Mulumba, W. Najy, and A. Diabat, "The drone-assisted pickup and delivery problem: An adaptive large neighborhood search metaheuristic," *Computers & Operations Research*, vol. 161, pp. 106435-106462, 2024.

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

[14] B. Ousmane, D. Moustapha, and C. Adama, "Genetic algorithm for the pickup and delivery problem with time window by multi-compartment vehicles," *Transport and Telecommunication Journal*, vol. 22, pp. 343-352, 2021.

[15] A. A. H. Qizilbash, A. K. Yadav, K. Bregler, and W. Kraus, "Ant colony optimization for retail based capacitated vehicle routing problem with pickup and delivery for mobile robots," in *2023 European Conference on Mobile Robots (ECMR)*, Coimbra, Portugal, 2023, pp. 1-6.