2019 20th IEEE International Conference on Mobile Data Management (MDM)

A VLOS Compliance Solution to Ground/Aerial Parcel Delivery Problem

Ji Zhang[1], Ting Shen[1], Wenlu Wang[1], Xunfei Jiang[2], Wei-Shinn Ku[1], Min-Te Sun[3], and Yao-Yi Chiang[4]

[1]*Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA*
{jizhang, tzs0033, wenluwang, weishinn}@auburn.edu
[2]*Department of Computer Science, Earlham College, Richmond, IN, USA*
jiangxu@earlham.edu
[3]*Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan, ROC*
msun@csie.ncu.edu.tw
[4]*Spatial Sciences Institute, University of Southern California, Los Angeles, CA, USA*
yaoyic@dornsife.usc.edu

***Abstract*—This paper presents an exact solution and a heuristic solution to a UAV-assisted parcel delivery problem, in which UAVs can only be operated in Visual-Line-Of-Sight (VLOS) areas. In our proposed problem, we assume that trucks travel on road networks, and UAVs move in Euclidean spaces and can launch at any locations on roads. We first demonstrate the overview of our exact solution that iterates all permutations of destinations for an optimal delivery route. Given a specific delivery order, an intuitive approach needs to check all possible locations on roads in the VLOS areas and find a globally optimal location for every destination if UAVs are used for delivery. To avoid high computational cost of searching the optimal location at run-time, we propose an advanced index-based alternative, which computes optimal delivery routes in a pre-processing stage. Due to the nature of NP-hard problems, we also propose a heuristic approach that utilizes delivery groups for the proposed problem of practical size. All proposed solutions are evaluated through extensive experiments.**

***Index Terms*—UAV, Visual-Line-Of-Sight, Parcel Delivery Problem**

I. Introduction

The Travelling Salesman Problem (TSP) finding an optimal route for a salesman, who plans to travel to each of a list of cities exactly once and return to the home city, has been extensively studied and applied in the parcel delivery service and other real applications [14] [8]. On the other hand, Unmanned Aerial Vehicles (UAVs), or drones, are developed for assisting traditional delivery vehicles (such as trucks) since the delivery could be completed in a shorter time period with lower maintenance cost by using UAVs in specific circumstances. Many UAV-assisted delivery projects have been initialized. For example, Prime Air is designed to deliver packages to customers in 30 minutes or less at Amazon [3]. DHL will start a project for delivering medications and other urgently needed goods to the North Sea island of Juist by DHL parcelcopters [10]. The project Wing targets delivering aid to isolated areas for disaster relief [9].

As UAVs are not limited by established infrastructure (e.g., roads), a new delivery model, a truck and a UAV, was proposed for parcel delivery [18]. In the model, every truck is equipped with a UAV and all packages can be delivered by either of the two. Fig. 1 and Fig. 2 display examples of TSP and the UAV-assisted parcel delivery problem. The delivery route starts at a distribution center h (indicated by a box), and eventually returns to the distribution center after reaching five destinations ($\{d_1, d_2, d_3, d_4, d_5\}$ indicated by circles). The delivery truck (indicated by the triangle) could stop at any locations on the road. In the example of the UAV-assisted parcel delivery problem in Fig. 2, the truck stops on road r and the UAV flies to d_4 for delivery.

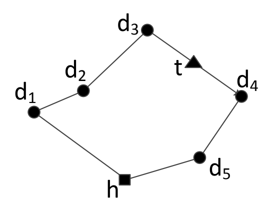


Fig. 1. An example of TSP in a Euclidean space. The line segments indicate a delivery route.

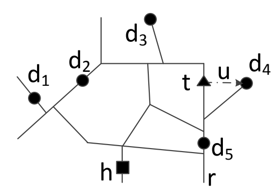


Fig. 2. An example of UAV-assisted parcel delivery problem. The line segments indicate the road network.

The travelling salesman problem is a well-known NP-hard problem in the field of combinatorial optimization [19], and the new UAV-assisted parcel delivery problem is more challenging. First of all, the fastest delivery route among destinations is usually assumed to be computed in a pre-processing stage in TSP. However, if packages could be delivered by either the truck or the UAV, the truck can stop at any places on the road and the UAV can be used for delivery. So, the fastest route among locations where the truck stops cannot be pre-determined, and may greatly vary case by case. Second, only one search space (either a Euclidean space or a road network) is usually considered in TSP, but the UAV-assisted problem assumes that the truck travels on road

2375-0324/19/$31.00 ©2019 IEEE
DOI 10.1109/MDM.2019.00-56

networks while the UAV moves in Euclidean spaces. Thus, the fastest delivery route may consist of paths in both search spaces, which significantly increases the complexity of the problem. Third, many variants of UAV-assisted problems have been investigated [18] [11] [21]. Nevertheless, none of these studies takes UAV regulations into account. For example, the Federal Aviation Administration (FAA) does not allow UAVs to be operated beyond the Visual-Line-Of-Sight (VLOS) of operators in the United States [1]. Existing solutions are not applicable to any VLOS compliance problems because the optimal delivery route may vary greatly by the VLOS distance and speeds of the truck and the UAV.

Therefore, we propose a novel Ground/Aerial Parcel Delivery Problem (GAPDP) with consideration of VLOS compliance. We develop an exact solution and a heuristic solution for the problem in this paper. In our exact solution, we check all permutations of destinations for the fastest delivery route. To calculate the fastest delivery route for destinations in a given order, we present an index-based exact approach that relies on an index over the VLOS areas for reducing the cost of the route calculation. This approach finds the fastest delivery route for every destination from all its entrances to its exits at a pre-processing stage. The fastest routes can be retrieved from the index, and our method can "jump" from one destination to the other without computing routes in the VLOS areas. Additionally, as the proposed problem is NP-hard, we also propose a heuristic solution that utilizes delivery groups for the proposed problem of practical size. All proposed solutions are evaluated through extensive experiments.

The contributions of this study are summarized below:

- We propose and formally define a new Ground/Aerial Parcel Delivery Problem (GAPDP), in which packages could be delivered by either a truck or a UAV.
- We develop an index-based exact solution that pre-computes the fastest delivery routes in VLOS areas of destinations.
- We propose a heuristic solution that produces an approximation for problems of practical size.
- We evaluate our solutions through extensive experiments over a real-world road network.

The rest of this paper is organized as follows. Section II surveys related work. The new UAV-assisted parcel delivery problem is formally defined in Section III. Our two solutions are illustrated in Sections IV and V. The experimental validation of our solutions is presented in Section VI. Section VII concludes this paper.

II. Related Work

In this section, we review previous studies related to the travelling salesman problem and UAV-assisted parcel delivery problems.

A. Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is an NP-hard problem [19]. An intuitive method for an exact solution of the problem is to check all permutations of destinations, and return the shortest route among those permutations, the computational complexity of which is $O(n \cdot n!)$, where n denotes the number of destinations. Held and Karp reduced the computational cost to $O(n^2 \cdot 2^n)$ by utilizing dynamic programming [12]. Jones and Adamatzky approximated TSP in a Euclidean space by shrinking "blob" [17]. In their method, a "blob" of material emerging from low-level interactions of particles is placed over destinations. The shrinkage process automatically and morphologically adapts to the configuration of destinations. And finally, a TSP tour is captured by tracing the perimeter of the blob among the destinations. Moreover, there are many studies that have investigated heuristic solutions for TSP [8]. A hybrid heuristic approach was developed for the multi-commodity pickup-and-delivery travelling salesman problem [13]. In the pickup-and-delivery TSP, each customer could be either a producer or a consumer of a package, or both. In other words, a package is assumed to be picked up at a customer, and then delivered to another customer in the delivery process. GELS-GA is a hybrid metaheuristic algorithm for multiple Travelling Salesman Problem (mTSP) [15]. Rather than the traditional TSP, mTSP has two or more salesmen, and all of them must return to the places where they start. GELS-GA combines Gravitational Emulation Local Search (GELS) algorithm and Genetic Algorithm (GA) for achieving the global optimum in a high possibility. Xu *et al.* also proposed a two-phase heuristic algorithm for mTSP [22]. Their method specifically considers workload balance and minimizes the overall travelling cost. They improved the K-means algorithm by grouping visited cities based on their locations and capacity constraints, and then designed a route planning algorithm for a delivery route. However, aforementioned studies are different from our proposed GAPDP problem; the shortest delivery routes among destinations in GAPDP depend on speeds of trucks and UAVs, the delivery order, the delivery method for each destination, UAV regulations, and more.

B. Drone-assisted Parcel Delivery Problem

Agatz *et al.* studied the Travelling Salesman Problem with Drones (TSP-D), formulated the problem as an Mixed Integer linear Programming (MIP) model, and developed route first-cluster second heuristic approaches based on local search and dynamic programming [2]. Ha *et al.* proposed two heuristic methods, either of which utilizes route-first cluster-second or cluster-first route-second strategies to solve TSP-D [11]. They used a new mixed integer programming formulation in the cluster step for both heuristics. Wang *et al.* focused primarily on the worst case of a vehicle routing problem with drones [21]. They assumed that a truck could be equipped with many drones, and their investigation showed that the results of the worst case depend on the number of drones on the truck and the relative speed of the drones to the truck. Murray and Chu proposed the flying sidekick TSP problem, in which customers can be served by either a driver-operated delivery truck or a UAV [18]. They solved the problem by utilizing an MIP formulation. Moreover, they also proposed a heuristic approach for parallel drone scheduling TSP problem. Ferrandez *et al.* explored the delivery time and energy consumption of a truck-drone delivery network [7]. Their proposed algorithm aims at minimizing the time of delivery by utilizing K-means clustering methods, and determining the optimal number of launch sites and drones per truck. Dorling *et al.* proposed two solutions for multi-trip vehicle routing problems [6]. Specifically, they first demonstrated an energy consumption model of drones, in which the energy consumption is approximately linear to the payload and battery weight. Then, they developed a cost function by using the energy consumption model and drone reuse. Finally, they proposed a method that minimizes costs by considering the limit of delivery time, and a method that minimizes the delivery time under a budget constraint. A Randomized Variable Neighborhood Descent (RVND) heuristic method was proposed to TSP-D problem [5]. In the method, practical restrictions, such as the flying time limit of drones and the limit of payload, are considered, and the RVND heuristic is used in local searches to find an optimal delivery route. Their experimental results show that the help of drones can save up to 20% of time in the last mile delivery.

However, all methods mentioned above are not applicable to our proposed problem due to the difference in assumptions of the problems. None of these studies considers the VLOS restriction. The optimal delivery route may vary greatly by setting different VLOS distances or speeds of the truck and the UAV. Many studies output an approximation [2] [11] [18]; while we propose an exact solution to the VLOS-compliance UAV-assisted parcel delivery problem. Moreover, Murray and Chu assumed that the truck could move on road networks while the UAV is in flight; but the truck and the UAV cannot be operated simultaneously in our problem because the driver can only either drive the truck or operate the UAV at a particular time. In addition, in many countries, UAVs are not allowed to land at or takeoff from a moving vehicle. Agatz *et al.* assumed that the UAV can only land on and depart from the truck while it is parked at a customer location, while the truck is allowed to park at any locations on roads in our problem. More than one UAV can be used with a truck in the problems proposed by Wang *et al.* and Ferrandez *et al.*; while there is only one UAV available with a truck for delivery in our problem, as each UAV requires a dedicated operator to handle.

III. Problem Definition

Given a road network G =(V, E), where V denotes a set of vertices and E is a set of edges, a spatial object is defined by a tuple $<x_1,\ x_2>$in a Euclidean space R. Every vertex v=$<x_1,\ x_2>$is also in R. $D^R(.,.)$ and $D^G(.,.)$ represent the distance between two spatial objects in the Euclidean space R and on the road network G, respectively.

Definition Given a set of destinations $D = \{d_1, ..., d_n\}$, there are a truck t and a UAV u used for parcel delivery. Every destination is served exactly once either by the truck or the UAV. The Ground/Aerial Parcel Delivery Problem (GAPDP) targets at finding an optimal route that minimizes the delivery time of serving all destinations once and returning to the distribution center. The delivery time can be calculated by

$$A = A_{h,d_1} + A_{d_1,d_2} + ... + A_{d_i,d_{i+1}} + ... + A_{d_{n-1},d_n} + A_{d_n,h} \quad (1)$$

where $A_{d_i,d_{i+1}}$ represents the time between the completion of delivery to d_i and to d_{i+1}, and A_{h,d_1} and $A_{d_n,h}$ are the delivery time of d_1 from the distribution center, and the time returning to the distribution center from d_n.

To make the proposed problem more realistic, there are assumptions in the problem as follows.

- The truck travels on road network at a fixed speed t^v and the UAV moves in Euclidean space at a fixed speed u^v. The truck can stop at any locations on the road for delivery.
- The driver is allowed to operate both truck and UAV for delivery. However, the truck and UAV cannot move simultaneously.
- UAV carries one package at a time. UAV returns to the truck immediately upon the completion of delivery.
- There is only one package for a destination. If there are two or more packages to a destination, it is conceptually equivalent to setting up a destination at the same location for each package.
- The delivery method can be explicitly selected by customers (because they may prefer using the truck for delivery, the package may exceed the payload capacity of the UAV, or a signature may be required), which indicates that a package could be delivered only by the truck or the UAV. If the delivery method is not explicitly selected, both the truck and the UAV can be used by default.
- The power of the UAV is sufficient for a round trip of delivery (a round trip from the truck to a destination); the time of battery replacement is negligible in this research.
- Due to the requirements of the FAA, the distance between the UAV and the driver/operator cannot be greater than a threshold, which is called "Visual-Line-Of-Sight" (VLOS) distance.
- All destinations can be at any locations on the road network.

IV. Design

In this section, we first present that the proposed GAPDP problem is NP-hard, and provide an overview of our exact solution in subsection IV-A. Then, we illustrate our index-based exact solution in subsection IV-B.

A. Overview

To achieve an exact solution to the travelling salesman problem, all permutations of destinations have to be visited, and the route that provides the shortest travelling time is the optimal route as an exact solution. The computational complexity of this intuitive method is $O(n \cdot n!)$, where n indicates the number of destinations. An improvement by utilizing dynamic programming techniques had been proposed, which could reduce the computational cost to $O(n^2 \cdot 2^n)$ [12].

Algorithm 1 GAPDP($\mathbb{D}$, h)

1. shortestDeliveryTime = ∞
2. **for** $D' \in \{\text{all permutation of } \mathbb{D}\}$ **do**
3. shortestDeliveryTime = min(shortestDeliveryTime, ShortestTime(D', h));
4. **end for**
5. **return** shortestDeliveryTime;

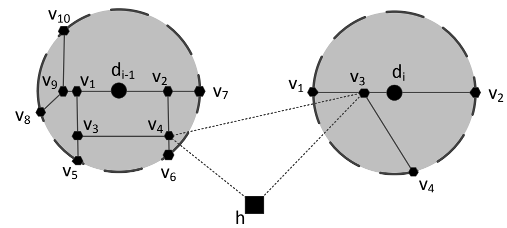


Fig. 3. An example of selecting the fastest delivery route for two consecutive destinations.

Both approaches assume that the shortest distance between any two destinations can be calculated in constant time ($O(1)$). However, this assumption is not applicable to our proposed GAPDP problem because the location where the truck stops for delivery may greatly vary by the delivery order of destinations, speeds of the truck and the UAV, the VLOS distance, and more. Additionally, the optimal delivery route may consist of the fastest paths that connect destinations in Euclidean spaces and on road networks, which makes the GAPDP problem more challenging than the TSP problem.

Therefore, we propose an exact solution to address the GAPDP problem. Because GAPDP is NP-hard, shown in Theorem IV.1, our solution follows a fundamental idea that finds an optimal delivery route by checking all permutations of destinations. The details are described in Alg. 1, which receives a set of destinations $\mathbb{D}$ and a starting distribution center h, and returns the fastest delivery time for $\mathbb{D}$. For a specific permutation of $\mathbb{D}$ in the FOR loop at lines 2 to 4, the function ShortestTime(D', h) calculates the shortest delivery time of destinations D' in a specific order. The variable $shortestDeliveryTime$ always maintains the shortest time of the delivery. To find the fastest route of destinations in a given order, we propose an index-based approach, which pre-computes the fastest delivery route in VLOS areas of destinations. Consequently, we could "jump" from an entrance of the VLOS area of a destination to its exit without route searching at run-time.

Theorem IV.1. ***GAPDP is NP-hard.***

Proof. GAPDP is NP-hard, since the Travelling Salesman Problem (TSP) is NP-hard [19], and TSP is a special case of the GAPDP problem when all packages are restricted to be delivered by the truck. □

B. The Fastest Route for Destinations in a Given Order

A major challenging problem in GAPDP is that the fastest delivery route of destinations in a given order may greatly vary under different settings of speeds of the truck and the UAV, delivery methods of destinations, and the VLOS distance. In this subsection, we illustrate an approach that finds the fastest delivery route for destinations in a given order. The fastest route is represented by a sequence of stop locations, where the truck is stopped and UAV is launched for delivery.

Since the operating range of UAVs is restricted by the Visual-Line-Of-Sight (VLOS) distance from operators [1], the truck must stop and the UAV must be launched at a location in the VLOS area of a destination for delivery. Thus, an intuitive algorithm visits all roads in the VLOS areas of destinations, and finds a sub-optimal stop location on each road. Then, the best route for serving destinations can be easily derived from the best sub-optimal stop locations. It is worth noting that there is a special case, in which the delivery time of a destination by truck is equal to the delivery time of stopping at the destination and using UAV for delivery because the delivery distance for UAV is 0. With this observation, our algorithm assumes that the UAV is used for all delivery for simplifying the problem. If the delivery by truck is the optimal method or the truck is a preferred method for a destination, our algorithm returns an equivalent case of using the UAV (the delivery distance of the UAV is 0), and the fastest delivery route and the shortest delivery time are the same with the methods considering delivery by truck.

Fig. 3 displays an example of serving two consecutive destinations d_{i-1} and d_i. We only display one route $\{h$, $d_{i-1}.v_4$, $d_i.v_3$, $h\}$ for better illustration. Initially, the truck starts from h, and we calculate the travelling time from h to all vertices in the VLOS area of d_{i-1}, where $d_{i-1}.\mathbb{V}$ = $\{d_{i-1}.v_1$, ..., $d_{i-1}.v_{10}\}$. Then, we set those vertices as starting points, and calculate the delivery route for d_i. Each of these routes ends at a vertex in the VLOS area of d_i. So, the fastest delivery route to a vertex of d_i can be found by examining all combinations of vertex of d_{i-1} and d_i. Then, in the next iteration, the vertex of d_i will be a starting point used for calculating the delivery time of the next destination.

In addition to checking all permutations of destinations, the intuitive approach also needs to search the network space for calculating the shortest network distance between two locations, and iterate all roads to find an optimal stop location in VLOS areas, the computational cost of which would be considerably high due to large number of possible stop locations and complex road networks. To avoid the computation at run-time, we turn to an alternative that pre-computes optimal delivery routes in the VLOS areas. We observe that the truck has to enter a VLOS area before delivery, and leave the area for the next if the VLOS areas of two destinations do not overlap with each other. If this is the case, the optimal delivery routes can be calculated at a pre-processing stage, and we can "jump" over the VLOS areas by retrieving the optimal routes from index.

Therefore, we develop an index-based approach, in which routes from entrances of destinations to their exits are pre-

Algorithm 2 ShortestTime($\mathbb{D}$, h)

1\. **for** $d_i \in \mathbb{D} = \{d_1, ..., d_n\}$ **do**
2\.   Let $d_i.map$ be the mapping from entrances to exits;
3\.   **if** $d_i = d_1$ **then**
4\.   **for** $e \in d_1.Exit$ **do**
5\. $d_1.time[e]$ = min($D^R(h, e')/t^v$ + $d_1.map[e'][e]$ — $e' \in d_1.Exit$);
6\.   **end for**
7\.   **else**
8\.   Let d_{i-1} be the last visited destination;
9\.   **for** $e \in d_i.Exit$ **do**
10\. $d_i.time[e]$ = min($d_i.map[e'][e]$ + $D^R(e', e'')/t^v$ + $d_{i-1}.time[e'']$ — $e' \in d_i.Exit$, $e'' \in d_{i-1}.Exit$);
11\.   **end for**
12\.   **end if**
13\. **end for**
14\. Let d_n be the last destination in D;
15\. **for** $e \in d_n.Exit$ **do**
16\. $d_n.time[e]$ += $D^R(e, h)/t^v$;
17\. **end for**
18\. **return** min($d_n.time[e]$);

built. We denote $d_i.Exit$ to be the set of intersections of the road network and the VLOS circle of destination d_i. Here $d_i.Exit$ is also called the entrances or exits of d_i in this paper.

The details of the index-base approach are described in Alg. 2. We iterate all destinations in the given order in the FOR loop from lines 1 to 13. Then, given a specific destination d_i, if d_i is the first destination d_1, we calculate the shortest travelling time from the distribution center h to every exit of $d_1.e$ at lines 4 to 6. $D^R(h, e')/t^v$ indicates the travelling time from h to an exit $d_1.e'$, and $d_1.map[e'][e]$ is the shortest delivery time from the entrance $d_1.e'$ to the exit $d_i.e$ after serving d_1. So, the shortest time of serving d_1 and stopping at $d_1.e$ is kept in $d_1.time[e]$. Then, we calculate the shortest delivery time of an intermediate destination d_i in the FOR loop at lines 7 to 12. For a specific exit $d_i.e$, the shortest delivery time after d_i is served and the truck stops at $d_i.e$ is calculated in line 10, where $d_i.map[e'][e]$ represents the shortest delivery time of the route from the entrance $d_i.e'$ to $d_i.e$. Here $d_{i-1}.time[e'']$ denotes the shortest time when the truck stops at an exit of the last destination d_{i-1}, and $D^R(e', e'')/t^v$ is the travelling time from an exit $d_{i-1}.e'$ to an entrance $d_i.e''$. Finally, the travelling time from exits of the last destination to the distribution center is added to $d_n.time[e]$, and the shortest delivery time of $d_n.time[e]$ is the solution. It is easy to see that we have to check all exits of every destination. The computational complexity of Algorithm 2 is $O(n \cdot |d_i.Exit|^2)$, where $|d_i.Exit|$ denotes the number of exits of a destination d_i and n is the number of destinations. The computational complexity of our VLOS-index-based solution is $O(n^2 \cdot 2^n \cdot |d_i.Exit|^2)$.

Moreover, there is a special case that the Euclidean distance between two continuous destinations is shorter than the VLOS

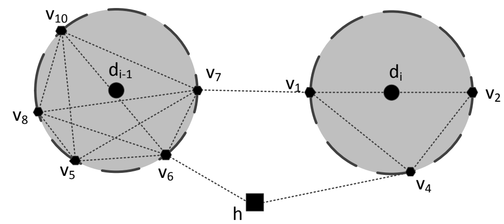


Fig. 4. An example of selecting delivery routes in Alg.2.

distance. In this case, the two VLOS areas overlap; the exit(s) of the VLOS area of a destination may be in the VLOS area of the second destination. This indicates that the parcel truck may already be in the VLOS area of the second destination after exiting the VLOS area of the first destination. The VLOS-based index over single VLOS area cannot be used for route searching in this case. This problem can be solved by merging the VLOS areas of two or more destinations if they overlap. For example, given two destinations d_{i-1} and d_i, a VLOS-based index can be constructed to contain all optimal routes from entrances to exits of the union of VLOS areas of d_{i-1} and d_i. All these routes start from an entrance of d_{i-1} and stop at an exit of d_i.

An example of our second approach is displayed in Fig. 4. The road networks in the VLOS areas of destinations are not accessed in our approach. Instead, the fastest delivery route and time have been calculated and are available in VLOS-based index. Accordingly, take the delivery route in Fig. 4 for example, we enter the VLOS area of d_{i-1} from entrance $d_{i-1}.v_6$, and jump to the exit $d_{i-1}.v_7$ to leave the VLOS area. Similarly, we jump over from the entrance $d_i.v_1$ to the exit $d_i.v_4$ in the VLOS area of d_i.

Next, we will present our method of finding the optimal route for one destination. This sub-problem can be formally defined as follows.

Definition Given two destinations d_{i-1} and d_i, d_{i-1} has been served, and d_i is the next destination for delivery. Let $d_i.\mathbb{V}$ be the set of vertices in the VLOS area of d_i, then this problem finds an optimal delivery route for d_i, which could start from any vertex in $d_{i-1}.\mathbb{V}$ and end at any vertex in $d_i.\mathbb{V}$. Here d_i must be served by the truck or the UAV exactly once on every route. All assumptions in Definition III are also applied to this sub-problem.

The fundamental idea of our method is to check every road in the VLOS area of d_i, and find an optimal stopping location on a road as a sub-optimum. The global optimum to serve d_i is the best of these sub-optima. Given a starting location v and a road r where the truck stops, there are four possible delivery routes starting from v and ending at either of the two end points of road r. As listed below, o denotes the optimal stop location on road r for delivery, and l_1 and l_2 represent the two end points of road r.

- Route 1: the truck first arrives at o through l_1, and then

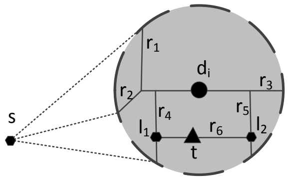


Fig. 5. An example of calculating the fastest delivery from from s to d_i.

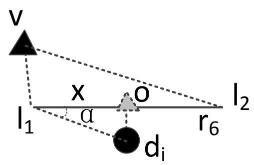


Fig. 6. An example of finding the optimal stop location on a road segment.

goes back to l_1 after delivery.

- Route 2: the truck first arrives at o through l_1, and then moves to l_2 for the next delivery.
- Route 3: the truck first arrives at o through l_2, and then moves to l_1 for the next delivery.
- Route 4: the truck first arrives at o through l_2, and then goes back to l_2 after delivery.

The total delivery time of these four candidate routes are

$$\begin{cases} \frac{D^R(v,\ l_1)+2\cdot D^R(l_1,\ o)}{t^v} + \frac{2\cdot D^G(o,\ d_i)}{u^v} & Route\ 1 \\ \frac{D^R(v,\ l_1)+D^R(l_1,\ l_2)}{t^v} + \frac{2\cdot D^G(o,\ d_i)}{u^v} & Route\ 2 \\ \frac{D^R(v,\ l_2)+D^R(l_1,\ l_2)}{t^v} + \frac{2\cdot D^G(o,\ d_i)}{u^v} & Route\ 3 \\ \frac{D^R(v,\ l_2)+2\cdot D^R(l_2,\ o)}{t^v} + \frac{2\cdot D^G(o,\ d_i)}{u^v} & Route\ 4 \end{cases} \tag{2}$$

where

$$D^G(o, d_i) = \sqrt{D^G(l_1, d_i)^2 + D^R(l_1, o)^2 - 2\cdot D^G(l_1, d_i)\cdot D^R(l_1, o)\cdot \cos\alpha} \\ 0 \le D(l_1, o), D(l_2, o) \le D(l_1, l_2) \tag{3}$$

It is easy to observe from Equ. 2 that the delivery times of Routes 2 and 3 only depend on $D^G(o,\ d_i)$, which is the distance between o and d_i in Euclidean space. $D^R(v, l_1)$, $D^R(v, l_2)$ and $D^R(l_1, l_2)$ are the network distance among s, l_1, and l_2, which are constants in the equation. Therefore, for the shortest delivery times of Routes 2 and 3, o should be selected at a location closest to d_i on the road r. If d_i is on r, then o should be at the location of d_i. If d_i is not on the road r and the projection of r from d_i is on r, then the projection is the optimal stop location for the delivery. If the projection is out of r, the end point of r closer to the projection is the optimal stop location.

For Routes 1 and 4, let $x = D^R(l_1,\ o)$, then the delivery times of the two routes can be represented by

$$\begin{cases} \frac{D^R(v, l_1)+2\cdot x}{t^v} + \frac{2\cdot\sqrt{D^G(l_1,d_i)^2+x^2-2\cdot\cos\alpha\cdot x\cdot D^G(l_1,d_i)}}{u^v} \quad Route\ 1 \\ \frac{D^R(v, l_2)+2\cdot(D^R(l_1,\ l_2)-x)}{t^v} + \frac{2\cdot\sqrt{D^G(l_1,d_i)^2+x^2-2\cdot\cos\alpha\cdot x\cdot D^G(l_1,d_i)}}{u^v} \quad Route\ 4 \end{cases} \tag{4}$$

where α is the angle of the road r and the line segment from l_1 to d_i, and $0 \le x \le D(l_1, l_2)$. $D^R(v,\ l_1)$, $D^G(l_1, d_i)$, $D^R(l_1,\ l_2)$, t^v, and u^v are constants. Due to the space limit, the optimal locations for the two routes can be found at [20].

It is worth noting that if there exist two or more optimal stopping locations on one road, the total delivery time in Equ. 2 includes the time of moving to an end of the road twice. Theoretically, in this case, we have to partition the road into two or more road segments in such a manner that there is at most one stopping location on one road segment.

Fig. 5 displays an example of six road segments $\{r_1,\ r_2,\ r_3,\ r_4,\ r_5,\ r_6\}$ in the VLOS area of a destination d_i and v is a vertex in the VLOS area of d_{i-1}. Fig. 6 displays the method of calculating the optimal stop location o on r_6. We set x to be the distance between $r_6.l_1$ and o, and other variables are constants in Equ. 4.

The computational complexity of our solution is $O(n^2 \cdot 2^n \cdot |V|^2)$, where $|V|$ denotes the average number of vertices in the VLOS areas of destinations. If all destinations are restricted to truck-delivery only, then $|V|$ becomes 1, and the proposed GAPDP becomes the travelling salesman problem, the computational complexity of which is $O(n^2 \cdot 2^n)$.

V. A HEURISTIC FOR GAPDP

From our experimental results (see Section VI), the proposed exact solutions require hundreds of seconds to determine an optimal delivery route of 15 destinations. The execution time grows exponentially due to the nature of NP-hard problems. In this section, we propose a heuristic approach for solving the GAPDP problem of practical size.

In general, our approach is derived from heuristic approaches for the travelling salesman problem (TSP). If all destinations are restricted to truck delivery, a heuristic TSP approach can be used to address the GAPDP problem. While, if UAV delivery is allowed at one or more destinations, our heuristic method prefers 1) serving all these destinations by UAV and 2) stopping at a location to serve as many destinations as possible.

Specifically, our method first generates destination groups, each of which contains destinations that can be served by UAV from one location. The VLOS areas of the destinations in a destination group must overlap on road networks, so that the truck can stop at a location in the overlapping area, and each destination can be served by UAV in a round trip between the stopping location and the destination. If a destination is served by truck only or its VLOS area does not overlap with the ones of others, the destination forms a destination group of itself. The details of the first step are described in Alg. 3. DS denotes the set of delivery group. In the while loop (lines 3-15), delivery groups are produced in iterations until $\mathbb{D}$ is an empty set.

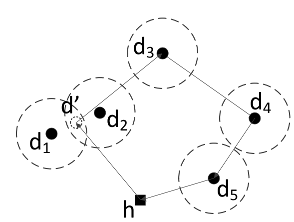


Fig. 7. An example route of delivery group.

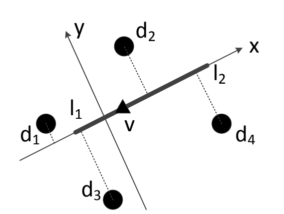


Fig. 8. An example of computing the optimal stopping location on a road segment for delivery.

Algorithm 3 GAPDP($\mathbb{D}$, h)

1. DS = $\emptyset$;
2. RoD^* = $\emptyset$;
3. **while** $\mathbb{D} \neq \emptyset$ **do**
4. Let d_i be an element of $\mathbb{D}$;
5. **if** d_i is served by truck only **then**
6. DS = $DS \cup \{\{d_i\}\}$; $\mathbb{D}$ = $\mathbb{D} \backslash \{d_i\}$;
7. **else**
8. **if** $\exists\, d_j \in \mathbb{D}$, $i \neq j$, VLOS(d_i)$\cap$ VLOS(d_j) $\neq \emptyset$ **then**
9. Let $G = \{d_k \in \mathbb{D} | VLOS(d_i) \cap VLOS(d_k) \neq \emptyset\}$;
10. DS=$DS \cup \{G\}$; $\mathbb{D}$ = $\mathbb{D} \setminus G$;
11. **else**
12. DS = $DS \cup \{\{d_i\}\}$; $\mathbb{D}$ = $\mathbb{D} \backslash \{d_i\}$;
13. **end if**
14. **end if**
15. **end while**
16. $RoDS^*$ = TSP(DS, h);
17. **for** $ds \in RoDS^*$ **do**
18. **if** $|ds|$ = 1 **then**
19. RoD^* = $RoD^* \cup ds$;
20. **else**
21. RoD^* = $RoD^* \cup$ DeliveryRouteInGroup(ds);
22. **end if**
23. **end for**
24. **return** RoD^*;

A heuristic TSP approach is applied to search an optimal visiting order for the delivery groups. If a delivery group has two or more destinations, we use the central location of the destinations as the delivery location of the delivery group. At line 16 of Alg 3, $RoDS^*$ represents the visiting order of delivery groups generated by a heuristic TSP solution. Then, we iterate every delivery group in $RoDS^*$ in order, and compute the delivery order of destinations in a specific group by using Alg. 4. Finally, the delivery order of destinations RoD^* is returned as the result, and the delivery route and cost can be computed by using Alg. 2.

Fig. 7 displays an example, in which the VLOS areas of d_1 and d_2 overlap. The delivery group of d_1 and d_2 and its central location d' is found and used as the delivery location of the group in Alg. 3. Then, a heuristic TSP approach is applied to a delivery problem of $\{h, d', d_3, d_4, d_5\}$, and Alg. 4 is used to compute the delivery order of $\{d_1, d_2\}$ in the group.

Algorithm 4 DeliveryRouteInGroup(ds)

1. RoD^* = $\emptyset$;
2. **while** $ds \neq \emptyset$ **do**
3. $D = \{d_i \in ds \mid \bigcap_{VLOS(d_i)} \neq \emptyset\ \}$
4. Iterate all roads in $\bigcap_{VLOS(d_i)}$, and find an optimal stopping location for delivery to D;
5. RoD^* = $RoD^* \cup D$; ds = $ds \backslash \{D\}$;
6. **end while**
7. **return** RoD^*;

Alg. 4 computes a delivery order in a delivery group. In each iteration, D denotes a set of destinations, each of which overlap with all others. So, D is equal to or is contained in ds. The optimal stopping location for serving all destinations in D by UAV can be computed in O($|D| \times |R|$), where $|R|$ indicates the number of roads in the overlapping area ($\{\bigcap_{VLOS(d_i)} | d_i \in D\}$). At line 4, we iterate all roads in the overlapping area, and find the best location among sub-optimal stopping location on each road for delivery. The sub-optimal location on each road can be determined in O($|D|$). Take Fig. 8 for example, assume that the truck stops at $v = \{x_v, 0\}$ on road $l_1 l_2$ for serving $\{d_1, d_2, d_3, d_4\}$. The delivery cost is $\sum_{i=1}^{4} \sqrt{(d_i.x - x_v)^2 + d_i.y^2}$ ($l_1 \leq x \leq l_2$), the minimum value of which can be easily obtained by using the derivative of the function.

VI. Experimental Validation

In this section, we evaluated the performance of our proposed solutions to the novel GAPDP problem over the road network of Oldenburg, Germany (containing 7K roads and 6K nodes [4]). The road network was normalized to the space of $[0, 10^4]^2$. The destinations were randomly selected on the road network. Our proposed algorithms were implemented in the C programming language. In our experimental results, "Exact Solution" refers to our exact algorithm that maintains the fastest route in a VLOS-based index. We also developed our heuristic method that applies the nearest neighbor travelling salesman solution. All data were loaded into the main memory during the execution of simulations.

The VLOS-based index was pre-built and the road distance between every pair of nodes was pre-computed in a preprocessing stage. This computation time is not included in the response time reported in our experimental results.

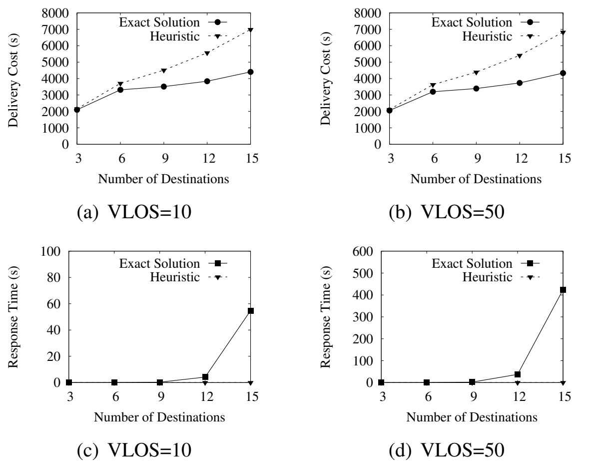
All the experiments were conducted on a Ubuntu Linux server equipped with two Intel Xeon E5-2670 v3 2.30 GHz processors and 256 GB of memory.

A. Effect of Number of Destinations

We varied the number of destinations from 3 to 15 in the first group of experiments. We fixed the normalized VLOS distance to 10 and 50 units. All destinations can be served by either the truck or the UAV. The normalized speeds of the truck and the UAV were fixed at 40 units per second. The experimental results are displayed in Fig. 9 and Table I.

The cost of delivery produced by the heuristic method grows faster than our exact solution, but the key point is that the heuristic method can complete the computation in milliseconds in all cases of this group of experiments while the exact solution needs at least 60 seconds or 400 seconds in cases of 15 destinations. This verifies that GAPDP is NP-hard; the response time of the exact solution increases exponentially.

Moreover, the exact method took longer against queries with larger VLOS areas. For example, it required 54.7 seconds when the VLOS distance was equal to 10 units; while the response time increased to 423 seconds if the VLOS distance was set to 50 units. This is because the VLOS-based index method needed to visit more entrance-exit pairs of VLOS



(a) VLOS=10 (b) VLOS=50

(c) VLOS=10 (d) VLOS=50

Fig. 9. Vary the number of destinations.

TABLE I Response Time of our Heuristic Method.

VLOS=10	Number of Destinations	20	50	100
	Response Time (s)	0.003	0.007	0.019
VLOS=50	Number of Destinations	20	50	100
	Response Time (s)	0.019	0.040	0.099

areas. In this group of experiments, the VLOS-index can be built, on average, in 9.7 seconds.

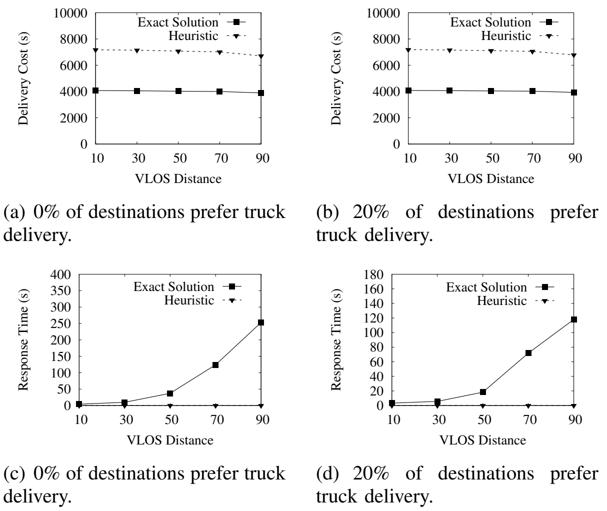
Table I displays the response time of our heuristic method in cases of up to 100 destinations (A FedEx driver typically has 75-80 stops per day [16]). The results show that our heuristic method helps us solve the real delivery problems in 0.1 second, which cannot be achieved by our exact solutions.

B. Effect of VLOS distance

We took a closer look at the effect of VLOS distance to our proposed solutions by increasing the normalized VLOS distance from 10 to 90 units. The number of destinations was fixed at 12. The normalized speeds of the truck and the UAV were fixed at 40 units per second. We evaluated two cases, in which 100% and 80% of destinations can be served by either the truck or the UAV. The experimental results are displayed in Fig. 10.

The exact method ran longer for queries with larger VLOS areas. According to our analysis in Section IV, the computational cost of the method is quadratic to the VLOS distance, since the number of road segments and the VLOS areas increase quadratically to the VLOS distance. Moreover, queries can be completed in a shorter period of time if 20% of destinations were restricted to truck delivery only. If a destination prefers truck delivery, there is no need to search optimal routes in the VLOS areas. Instead, the truck has to stop at the destination, and the fastest route to the destination on the road network is the optimal route for delivery.

Fig. 10 also compares the delivery cost by varying the VLOS distance. There is a slight drop when VLOS areas become larger because there could be a better delivery route found in a larger VLOS area.



(a) 0% of destinations prefer truck delivery. (b) 20% of destinations prefer truck delivery.

(c) 0% of destinations prefer truck delivery. (d) 20% of destinations prefer truck delivery.

Fig. 10. Vary the VLOS distance.

C. Effect of Percentage of Destinations Preferring Truck Delivery Only

We also evaluated the effect of delivery methods in experiments by varying the percentage of destinations preferring truck delivery only. We fixed the number of destinations at 12, and the normalized speeds of the truck and the UAV were fixed at 40 and 60 units per second, respectively. The normalized VLOS distance was set to 50 and 90 units. The experimental results are displayed in Table II.

From Table II, we observe that the delivery cost of our exact method increases and the benefit of utilizing the UAV for delivery decreases as more destinations prefer truck delivery. Moreover, we can save more delivery time in larger VLOS areas. In the case of VLOS areas equal to 90, if all destinations allow UAV or truck for delivery, we can save 12% in the delivery cost compared to the case of all truck delivery only.

VII. Conclusion and Future Work

In this research, we formulated a novel problem that utilizes both a truck and a UAV for parcel delivery. We also considered the VLOS regulation that restricts the UAV operating range. Then, we proposed an exact solution that pre-computes the fastest delivery routes in VLOS areas for the problem. Due to the high computational cost of the exact solution, we also developed a heuristic solution for the proposed problem in practical size. We demonstrated the performance of the proposed two solutions through extensive simulations.

TABLE II Delivery cost of our exact solution by varying the percentage of truck delivery only.

	Percentage of destinations preferring truck delivery			
	0	20	80	100
VLOS=50	1089	1110	1153	1172
VLOS=90	1045	1083	1155	1172

For future work, we will apply more practical restrictions/assumptions, such as UAV regulations or delivery with help of two or more UAVs, in problems in order to make our solutions useful for more applications.

REFERENCES

[1] F. A. Administration. Summary of Small Unmanned Aircraft Rule (Part 107). https://www.faa.gov/uas/media/Part_107_Summary.pdf.

[2] N. Agatz, P. Bouman, and M. Schmidt. Optimization Approaches for the Traveling Salesman Problem with Drone. *ERIM Report Series*, 2016.

[3] I. Amazon.com. Amazon Prime Air. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011. Accessed on May 1, 2019.

[4] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.

[5] J. C. de Freitas and P. H. V. Penna. A Randomized Variable Neighborhood Descent Heuristic to Solve the Flying Sidekick Traveling Salesman Problem. *Electronic Notes in Discrete Mathematics*, 66:95–102, 2018.

[6] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2017.

[7] S. M. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management*, 9(2):374–388, 2016.

[8] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328, 2014.

[9] I. Google. Project Wing. http://www.bbc.com/news/technology-28964260. Accessed on May 1, 2019.

[10] D. P. D. Group. http://www.dhl.com/en/press/releases/releases_2014/group/dhl_parcelcopter_launches_initial_operations_for_research_purposes.html. Accessed on May 1, 2019.

[11] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà. Heuristic methods for the Traveling Salesman Problem with Drone. https://pdfs.semanticscholar.org/59b4/8e77e710917d85facb5d2cebf2e2ebd5dfae.pdf, 2015. Accessed on May 1, 2019.

[12] M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

[13] H. Hernández-Pérez, I. Rodríguez-Martín, and J.-J. Salazar-González. A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 251(1):44–52, 2016.

[14] K. L. Hoffman, M. Padberg, and G. Rinaldi. Traveling Salesman Problem. *Encyclopedia of Operations Research and Management Science*, pages 1573–1578, 2013.

[15] A. A. Hosseinabadi, M. Kardgar, M. Shojafar, S. Shamshirband, and A. Abraham. Gels-ga: hybrid metaheuristic algorithm for solving multiple travelling salesman problem. In *ISDA*, pages 76–81, 2014.

[16] R. Inman. A Day in the Life of a FedEx Driver. https://www.aol.com/2010/05/28/fedex-home-delivery-driver/. Accessed on May 1, 2019.

[17] J. Jones and A. Adamatzky. Computation of the travelling salesman problem by a shrinking blob. *Natural Computing*, 13(1):1–16, 2014.

[18] C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86 – 109, 2015.

[19] C. H. Papadimitriou. The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

[20] T. Shen, J. Zhang, W. Wang, X. Jiang, W.-S. Ku, M.-T. Sun, and Y.-Y. Chiang. Optimal Stop Location Selection for two specific cases in GAPDP. *Technical Report, CSSE17-04, Auburn University*, 2017.

[21] X. Wang, S. Poikonen, and B. Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, pages 1–19, 2016.

[22] X. Xu, H. Yuan, M. Liptrott, and M. Trovati. Two phase heuristic algorithm for the multiple-travelling salesman problem. *Soft Computing*, pages 1–15, 2017.