Traveling Salesman Problem with Truck and Drones: A case study of Parcel Delivery in Hanoi*

Quang Huy Vuong[1] [0000−0003−0776−046X], Giang Thi-Huong Dang[2] [0000−0002−3821−1001], Trung Do Quang[3] [0000−0002−0091−0093], and Minh-Trien Pham[1] ** [0000−0002−2493−8330]

[1] VNU University of Engineering and Technology, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
[2] University of Economic and Technical Industries, 456 Minh Khai, Vinh Tuy, Hai Ba Trung, Ha Noi, Vietnam
[3] Academy of Cryptography Techniques, 141 Chien Thang, Tan Trieu, Thanh Tri, Ha Noi, Vietnam

Abstract. Unmanned Aerial Vehicles (UAVs), commonly known to the public as drones, have recently been utilized for military and many agriculture applications. In the near future, drones are likely to become a potential way of delivering parcels in urban areas. In this paper, we apply a heuristic solution for the parallel drone scheduling salesman problem (PDSTSP) for real-world optimization problems, where a set of customers requiring a delivery is split between a truck and a fleet of drones, with the aim of minimizing the completion time (or the makespan) required to service all of the customers. The study is based on the analysis of numerical results obtained by systematically applying the algorithm to the delivery problem in Hanoi. The results demonstrate that the utilization of drones might reduce the makespan significantly, and our approaches effectively deal with the delivery problem in Hanoi.

Keywords: Parallel drone scheduling · Drone delivery · Heuristic algorithm.

1 Introduction

Recently, drones have received more attention as a new distribution method for transporting parcels. Several companies have put considerable efforts into drone delivery research. A remarkable event occurred in December 2013, a delivery service using drones called Prime Air [2] was first publicly introduced by Jeff Bezos, the CEO and founder of Amazon - the largest online retailer. Then in

* This work was supported by the Domestic Master/ PhD Scholarship Programme of Vingroup Innovation Foundation.

** Corresponding author:
E-mail address: trienpm@vnu.edu.vn. (Minh-Trien Pham)

2016, it said it had made its first successful drone delivery to a customer in Cambridge, England. In 2014, Google began testing its drone delivery service called Google's Wing in Australia. Now Wing's drones are being used to deliver essentials such as medicine, food to residents in lockdown in Virginia, USA during coronavirus pandemic [9]. DHL also launched its drones called parcelcopter in 2016, which could deliver parcels to customers in remote areas such as in the Alps [3]. In 2017, a Silicon Valley start-up Matternet developed their drone delivery system for medical applications in Switzerland [5]. Other similar systems have been launched by many companies such as Alibaba [1], JD.com [4]. Drones can provide significant advantages over traditional delivery systems handled only by trucks. The comparison of trucks and drones is summarized in Table 1 [14].

Table 1. Comparison of truck and drones.

Vehicle	Delivery sapce	Speed	Parcel weight	Parcel capacity	Delivery range
drone	**air**	**fast**	light	one	short
truck	ground	slow	**heavy**	**many**	**long**

Alongside the attention in the industry, in the last few years, several publications in the literature on truck-drone collaboration have been proposed. Khoufi et al. [13] provided a comprehensive survey on this field. Using both truck and drones in the delivery system gives rise to new variants of travelling salesman problems (TSP). In 2015, Murray and Chu [16] was first proposed the problem, named the PDSTSP. In the PDSTSP, a truck and a fleet of drones perform independent tasks. The PDSTSP aims to minimize the makespan (the distance in time that elapses from the start of the delivery process to the end after serving all customers). The authors proposed simple greedy heuristics to obtain solutions but only for small-size instances due to the NP-hard of the problem. Mbiadou Saleu et al. [15] presented an improved algorithm for PDSTSP called two-step heuristic. A dynamic programming algorithm with bounding mechanisms is used to decompose the customer sequence into a trip for the truck and multiple trips for drones. Kim and Moon [14] proposed an extension of PDSTSP named The traveling salesman problem with a drone station(TSP-DS), in which drones launched from a drone station, not the distribution center.

In this paper, we also consider the parallel drone scheduling problem, based on the dynamic programming-based algorithm introduced by Saleu et al. [15] with some modifications. First, we consider the real-world problem with the time-dependent based speed model for the truck. The speed of the truck is affected by traffic conditions. Second, a constructive heuristic approach was applied to solve the TSP for the truck tour, while a parallel machine scheduling algorithm still handles multiple drone tours. Finally, the algorithm is tested with real-world instances in Hanoi with different problem parameters to evaluate the performance and the potential of the algorithms for applying in real-world problems.

The paper is organized as follows. Section 2 provides the problem description of PDSTSP. The heuristic algorithm is described in Section 3. Section 4 shows experimental results and discussions of the results. Finally, Section 5 concludes the paper.

2 Parallel drone scheduling traveling salesman problem

In this paper, we investigate the PDSTSP presented in [16], in which a truck and drones depart and return dependently with no synchronization. There are reasons we decided to investigate this model. First, the PDSTSP with parallel utilization of a truck and drones in a non-synchronized way is more suitable for real-world problems, where the synchronized collaboration between truck and drones (truck carries drones) is challenging to deploy in practical delivery problems. Moreover, we consider the case that the depot locates in a convenient position for drone delivery.

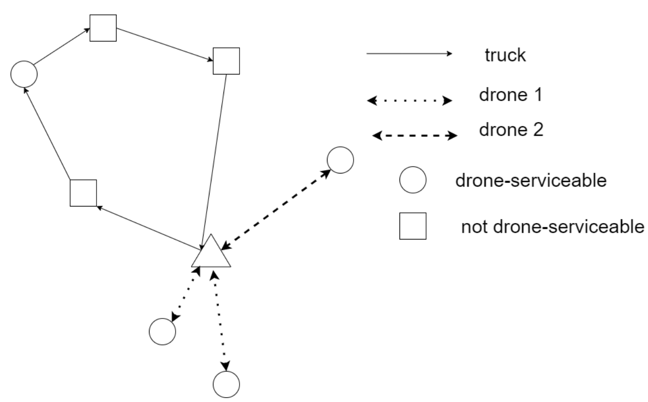

Fig. 1. An illustration of PDSTSP.

2.1 Problem definition

The PDSTSP can be represented as follows. Consider a set of nodes $N = \{0,...,n\}$ represents the set of customers and the depot (index 0). A truck and a fleet of homogeneous drones are available to deliver parcels to the customers from the depot. The truck starts from the depot, services a subset of customers along a TSP route, and returns back to the depot. Drones service customers directly from the depot, then return the depot while servicing a single customer per trip. Not all customers can be served by a drone because of practical constraints like the limited capacity or the limited flight range of a drone. We define D as a set of customers which can be served by a drone. These customers are referred to as drone-serviceable customers in the rest of the paper. Assume that the truck and the drones start from the depot at time 0. The objective of the PDSTSP is to minimize the time required for a truck and drones to return to the depot after servicing all customers (a customer must be serviced exactly once by either the truck or a drone). Since truck and drones work in parallel, the objective is also to find an optimal TSP route for a truck and optimal customer orders for drones. An illustration of PDSTSP is shown in Figure 1.

2.2 Time-dependent speed model

In this section, a model of time-dependent is constructed to capture the congestion in a traffic network. In real-life problems, the travel time between customers strongly depends on the traffic condition of the road network. It means the speed of the truck could vary depending on the time of the day. For example, the truck's speed during rush hour would be multiple times slower than the speed at night. The time dependency is modeled as follows. The time of day is partitioned into L intervals $[T_l, T_{l+1}]$, $l = 0, ..., L-1$. The average value of the truck speed is known. It should be noticed that the truck's speed may differ among arcs when the truck crosses the boundaries of an interval. The time-dependent travel speed of truck can be represented as:

$$v_{ijl} = v_{truck} Rand \in [F_L, F_U] \tag{1}$$

where v_{truck} represents the average speed of the truck, v_{ijl} is the speed of truck among arc (i, j) during the interval l, F_L and F_U are respectively the lower bound and upper bound value of congestion factors. The lower value of $Rand$, the more congestion there will be. During the peak hours, the value of $Rand$ should be close to F_L. The value of $Rand$ should be close to F_U during off-peak hours. To calculate the travel time between two cities (i, j), the distance between these nodes for the truck d_{ij} and the departure time t are needed. The time-dependent travel time value on arc (i, j) if departing from vertex i at time t is computed following Algorithm 1 as proposed by Ichoua et al. [12].

Algorithm 1 Computing the travel time of arc (i, j) at departure time t_0

$t \leftarrow t_0$
$l \leftarrow l_0 : T_{l_0} \le t_0 \le T_{l_0+1}$
$d \leftarrow d_{ij}$
$t' \leftarrow t + (d/v_{ijl})$
while $t' > T_{l+1}$ **do**
  $d \leftarrow d - v_{ijl} \times (T_{l+1} - t)$
  $t \leftarrow T_{l+1}$
  $t' \leftarrow t + (d/v_{ijl+1})$
  $l \leftarrow l + 1$
end while
return $t' - t_0$

3 Heuristic algorithm

3.1 Main algorithm

Based on the definition of the PDSTSP from [16], we follow the general scheme of Saleu et al. [15] to solve this problem. The PDSTSP is considered as a two-stage problem:

- Partitioning stage: splitting customers into two sets, a set for the truck and a set for the fleet of drones.
- Optimizing stage: solving a TSP for the truck and a parallel machine scheduling (PMS) problem for drones.

Our heuristic approach iterates over these stages. After the optimizing stage, the algorithm is repeated until the termination condition is met. Our heuristic method is described as follows with pseudocode:

1. Given a TSP tour T visiting the depot and all of the customers, a truck and a fleet of M drones start from the depot, the tour T is initialized with a greedy algorithm.
2. Update current solution: assign all the customers to the truck in order of sequence T, that means no customer is assigned to the drones.
3. $BestSolution = Solution$
   $BestCost = Cost(BestSolution)$
4. Initial tour T is split into two complementary subtours: T_{truck} for the truck and T_{drones} for the fleet of drones.
5. Tours improvement:
   - Truck tour T_{truck} is then repoptimized using TSP algorithm and improved using an improvement heuristic.
   - A set of tours for drones $T_{drone_1}, .., T_{drone_M}$ is obtained by a PMS algorithm from T_{drones}.
6. Update new solution with optimized tours for truck and drones.
7. If $Cost(Solution) < BestCost$ then $BestCost = Cost(Solution)$ and $BestSolution = Solution$
8. Drones tour $T_{drone_1}, .., T_{drone_M}$ are inserted into T_{truck} with Nearest Insertion algorithm to form a new tour T.
9. Check the termination criterion: the process is terminated if the exit criterion is met (typically computation time is reached), otherwise comeback to step 4.

Algorithm 2 illustrates the pseudo-code of the introduced heuristics.

A solution is presented as a set consisting a tour for the truck in order of T_{truck} and M tours for fleet of M drones in order of $T_{drone_1}, .., T_{drone_M}$. The cost of the truck is denoted by $TruckCost$ that indicates the completion time for the truck after servicing all customers in order T_{truck}. $DronesCost$ denotes the completion time for the fleet of drones. Therefore, the cost of a solution $Cost(Solution)$ is equal to $max(TruckCost, DronesCost)$, which indicates the completion time of the final vehicle returning to the depot after servicing all

Algorithm 2 Main algorithm

$T \leftarrow InitializeTSP()$
$BestSolution = Solution$
$BestCost = Cost(BestSolution)$
while $isTerminate = False$ **do**
    $Split(T)$
    $T_{truck} \leftarrow ImprovementHeuristic()$
    $T_{drone_1}, .., T_{drone_M} \leftarrow PMSalgorithm()$
    if $Cost(Solution) < BestCost$ **then**
        $BestCost = Cost(Solution)$
        $BestSolution = Solution$
        $T \leftarrow reInsertion(T_{truck}, T_{drone_1}, ..., T_{drone_M})$
    end if
end while

customers. A solution is considered to be better than others if its cost is smaller. If two solutions have the same cost value, the solution that has the smaller sum of $TruckCost$ and $DronesCost$ is the better solution.

3.2 Customers partitioning

The tour T is separated into two complementary parts T_{truck} and T_{drones}. We adapted an effective split procedure from Saleu et al. [15] with a remarkable change of cost calculation. It should be noticed that the cost of the truck now is affected by the traffic congestion model described in Section 2.2, since the result of the split procedure is also affected. Given a set of customers $N = \{0, ..., n\}$ and an additional node $n + 1$ represents the copy of depot 0. As mentioned in the previous section, a truck can serve all customers, but drones can only serve customers in a drone-serviceable subset $D \subseteq N$. The objective of partitioning phase is to find a partition between T_{truck} and T_{drones} that minimizes the $max(TruckCost, DronesCost)$. The details of the split procedure can be found in [15]. We briefly describe the algorithm with some modifications as follows:

1. The algorithm checks every node from depot 0 to destination node $n + 1$ in order of tour T. At any node i, a list of $(TruckCost, DronesCost)$ is induced by adding arc cost for every solution from node 0 to node i (for node j, two different solutions can occur when we decide whether node j is assigned to the drone or not).
2. With every arc (i, j), a cost vector (c^1_{ij}, c^2_{ij}) is generated. The component c^1_{ij} represents the cost incurred for the truck if it travels directly from i to j: $c^1_{ij} = d_{ij}$. The corresponding cost induced for the drone c^2_{ij}. If the truck travels directly from i to j, all drone-serviceable customers k in-between i and j are assigned to the drone: $c^2_{ij} = \sum \hat{d}_k$.
3. To reduce the number of solutions, before adding a solution to the list of $(TruckCost, DronesCost)$, all existing solutions are checked with the new solution to decide which solution should be removed. The best decomposition

(T_{truck} and T_{drones}) is retrieved from the best solution found in the list of cost at destination node $n+1$.
4. Before running the procedure, the departure time is set. The cost for the truck is calculated based on the given truck cost distance matrix, and the speed of the truck is affected by the time-dependent traffic congestion model.

3.3 Subtours Improvement

Truck tour improvement Truck tour T_{truck} retrieved from partitioning procedure is then reoptimized using Christofides algorithm [10]. After that, 2-opt heuristic [11] is used to improve it. For the TSP improvement phase, the congestion index is relaxed because it does not affect the result.

Christofides heuristic Christofides' algorithm [10] is a well-known tour construction heuristic for travelling salesman problem . For the given inputs, it starts with finding the minimum spanning tree T and then a minimum matching M on the odd-degree vertices. A new graph H is formed by adding M to T. Every vertex now has even degree. Therefore, H is Eulerian. Finally, a TSP tour is obtained by skipping visited nodes (using shortcuts). The algorithm is described as follows:

1. Find a minimum spanning tree MST (T)
2. Find vertexes in T with odd degree (O) and find minimum weight matching (M) edges to T
3. Form an Eulerian graph using the edges of M and T
4. Obtain a Hamiltonian path by skipping repeated vertexes (using shortcuts)

Two-opt local search The 2-opt algorithm was first proposed by Croes in 1958 [11]. The 2-opt algorithm examines all possible pairs of edges in the tour, removes and reconnects them to form a new tour. This transition is called a 2-opt move. If the new tour is longer or equal to the original one, we undo the swap. Otherwise, the move resulted in a shorter tour. In the 2-opt, when removing two edges, there is only one alternative feasible solution. The swap continues until it no longer improves the path.

Drone scheduling The customers sequence T_{drones} is decomposed to subtours for M drones. The objective of this phase is to find the minimum makespan of the schedule. The simple Longest Processing Time (LPT) algorithm is used. LPT assigns the customers with the longest cost to the drone with the earliest end time so far.

4 Experimental results

The heuristic approach has been tested with different benchmark sets. The first set is introduced by Saleu et al. [15] that generated from TSPLIB library with some parameters for sensitivity analysis. The second real-life instance set is generated using geodata from OpenSreetMap [7]. The details of instances generation are later described in Section 4.1. All computational works were conducted on an Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, and the algorithm is implemented in C++.

4.1 Instances and parameters setting

In this section, two data sets are used to evaluate the performance of the algorithm. The first set of instances used was adapted from [15]. The instances were generated from the classic TSPLIB library [8]. The customers are represented by coordinates x and y. The original TSP instances were modified for the PDSTSP problems:

- The percentage of drone-serviceable customers ranging from 0% to 100%.
- The Manhattan distances d_{ij} are calculated for the truck if it travels from i to j and Euclidean distances $\hat{d}_k$ for the drones if a drone serves customer k.

The second instances were generated from OpenStreetMap project [7]. A customer node is represented by coordinates at a geographic coordinate system with latitude and longitude units. The real travel distances are generated from Openrouteservices API [6].

The parameters that are inserted for the experiments are shown as follows:

- Fixed distance cost matrices of truck and drones.
- For the instances generated from TSPLIB, we indicate the speed factor $sp = v_{drone}/v_{truck}$. The cost of the drone is divided by the speed factor sp, and the speed of the drone is set to be one unit of distance per unit of time.
- The depot is located near the center of all customers.
- The number of drones M = 2.
- The planning horizon is divided into five intervals $L = 5$, the value of F_L and F_U are set to 0.5 and 1, respectively.
- The termination criterion (computing time limit) is set to 5 minutes.

4.2 Results and discussions

Results obtained from first data set of TSPLIB are shown in Table 2 - 4, where completion time indicates the cost value of the solution. The completion time is then compared to the optimal cost of the traditional TSP tour provided by $gap\%$ value. Columns labeled with $\%ds$ indicate the percentage of drone-serviceable customers.

In general, the results show that the completion times are reduced with the introduction of drones working in parallel. In addition, completion times are also improved when the percentage of drone-serviceable customers increases. It can be easily explained that when the percentage of drone-serviceable increases, assigning customers to the drones can reduce the completion time for the truck.

Table 2. Results of the algorithms on the TSPLIB instance att48

Instance name	%ds	CT	gap %
att48	0	47170	0
	20	40121	-14.9
	40	34761	-26.3
	60	32867	-30.3
	80	30221	-35.9
	100	30085	-36.2

Table 3. Results of the algorithms on the TSPLIB instance berlin52

Instance name	%ds	CT	gap %
berlin52	0	11175	0
	20	10013	-10.3
	40	9120	-18.4
	60	7914	-29.2
	80	6914	-38.1
	100	6592	-41.0

Table 4. Results of the algorithms on the TSPLIB instance eil101

Instance name	%ds	CT	gap %
eil101	0	988	0
	20	838	-15.2
	40	721	-27.0
	60	651	-34.1
	80	623	-36.9
	100	596	-39.7

Table 5. Impact of the speed factor (80% drone-serviceable customers)

Instance	Without sp factor		With sp factor	
	CT	DC	CT	DC
att48	34324	10	30221	18
berlin52	7760	11	6914	18
eil101	690	15	623	28

However, when the completion time of the truck and drones are nearly the same, increasing drone-serviceable customers does not affect the completion time too much.

We conduct a sensitivity analysis as shown in Table 5 to investigate the impact of the speed factor. Columns labeled with DC indicate the number of customers assigned to the drones. When the speed factor affects the completion time of the truck, the drone is able to serve more customers. The completion time of all three instances is also improved. Actually, the completion time also depends on the percentage of drone-serviceable customers. The drone is not able to visit more customers if the percentage of drone-serviceable customers is low.

Table 6. Comparison of results on TSPLIB instances (80% drone-serviceable customers)

Instance	gap%	
	Proposed algorithm	**Greedy approach**
att48	-35.9	-22.6
berlin52	-38.1	-24.1
eil101	-36.9	-23.5

In Table 6, the results obtained are compared with a baseline algorithm with a greedy strategy. The algorithm tried with all different combinations of drone-serviceable customers to be assigned to the truck route. Then, the TSP is solved by Path Cheapest Arc (PCA) algorithm, and the PMS is based on Shortest Job First (SJF). In Table 6, it is shown that the proposed algorithm shows significantly more efficiency, demonstrating the reliability of the algorithm for applying to practical problems. The baseline greedy algorithm can be easily trapped in local optimums in the complex search space of this problem.

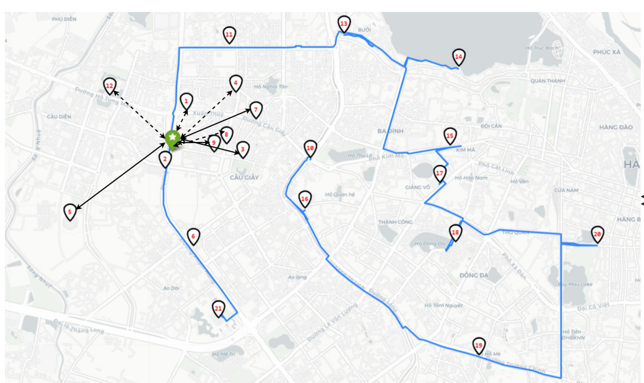

Fig. 2. An illustration of the result for real instance with 20 customers.

Table 7. Results for real instances (50% drone-serviceable customers)

Instance		%gap	DC
NC	ND		
20	1	-24.2	6
	2	-45.1	9
	3	-48.2	10
40	1	-19.1	14
	2	-36.5	18
	3	-36.5	18
80	1	-18.5	22
	2	-32.1	28
	3	-38.5	31

For the real delivery problem, we analyzed instances in which the number of nodes ranging from 20 to 100. The default truck speed v_{truck} is set to 40 km/h and is affected by the time-dependent traffic congestion model in Section 2.2. In addition, we vary the number of drones. An illustration of the results for real instances is shown in Figure 2. The results obtained from real instances are shown in Table 7. Columns labeled with NC and ND indicate the number of customers and the number of drones, respectively. The results show that the completion time of the real delivery problem is reduced when increasing the number of drones and the drones are able to visit more customers. However, in some cases, increasing the number of drones does not improve the solution, or not too much improvement. Therefore, the number of drones needs to be selected reasonably to save the operating cost. Finally, compared to the traditional delivery service (only truck), the effect of traffic congestion is reduced by the utilization of drones. The customers tend to be assigned to the drone to balance the completion time of both the truck and the drones.

5 Conclusions

In this paper, we applied a heuristic solution for the traveling salesman problems with a truck and a fleet of drones for real-world delivery service. A model of traffic congestion is used to explore the advantages of using a truck and drones in parallel. In addition, when applying to real-world applications, the introduction of drones would bring substantial improvements in the logistic operations of last-mile delivery in Hanoi. The results indicated that we could overcome congestion situations and significantly reduce the delivery completion time compared to traditional delivery services.

References

1. Alibaba uses drones to deliver products to Chinese islands - Geospatial World, https://www.geospatialworld.net/news/alibaba-uses-drones-deliver-products-chinese-islands/. Last accessed 13 July 2021.
2. Amazon.com: Prime Air, https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8node=8037720011. Last accessed 13 July 2021.
3. DHL drone delivery and parcelcopter technology — Discover DHL, https://discover.dhl.com/business/business-ethics/parcelcopter-drone-technology. Last accessed 13 July 2021.
4. JD.com makes drone deliveries as coronavirus cuts off usual modes - Nikkei Asian Review, https://asia.nikkei.com/Spotlight/Coronavirus/JD.com-makes-drone-deliveries-as-coronavirus-cuts-off-usual-modes. Last accessed 13 July 2021.
5. Matternet to launch medical drone delivery in Switzerland — Internet of Business, https://internetofbusiness.com/matternet-drone-delivery-switzerland/. Last accessed 13 July 2021.
6. Openrouteservice, https://openrouteservice.org/. Last accessed 19 May 2021.
7. OpenStreetMap, https://www.openstreetmap.org/. Last accessed 19 May 2021.
8. TSPLIB, http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html. Last accessed 19 May 2021.
9. Wing drones from Google deliver shopping during coronavirus pandemic, https://www.dezeen.com/2020/04/15/google-wing-drone-delivery-coronavirus-virginia/. Last accessed 13 July 2021
10. Christofides, N.: Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. (February) (1976)
11. Croes, G.A.: A Method for Solving Traveling-Salesman Problems. Operations Research **6**(6), 791–812 (1958). https://doi.org/10.1287/opre.6.6.791
12. Ichoua, S., Gendreau, M., Potvin, J.Y.: Vehicle dispatching with time-dependent travel times. European Journal of Operational Research **144**(2), 379–396 (2003). https://doi.org/10.1016/S0377-2217(02)00147-9
13. Khoufi, I., Laouiti, A., Adjih, C.: A Survey of Recent Extended Variants of the Traveling Salesman and Vehicle Routing Problems for Unmanned Aerial Vehicles. Drones **3**(3), 66 (2019). https://doi.org/10.3390/drones3030066
14. Kim, S., Moon, I.: Traveling salesman problem with a drone station. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2019). https://doi.org/10.1109/TSMC.2018.2867496
15. Mbiadou Saleu, R.G., Deroussi, L., Feillet, D., Grangeon, N., Quilliot, A.: An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. Networks **72**(4), 459–474 (2018). https://doi.org/https://doi.org/10.1002/net.21846, https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21846
16. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem : Optimization of drone-assisted parcel delivery. TRANSPORTATION RESEARCH PART C **54**, 86–109 (2015). https://doi.org/10.1016/j.trc.2015.03.005, http://dx.doi.org/10.1016/j.trc.2015.03.005