Transportation Research Part C 91 (2018) 249–262

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A decomposition-based iterative optimization algorithm for traveling salesman problem with drone☆

Emine Es Yurek*, H. Cenk Ozmutlu

Department of Industrial Engineering, Uludag University, Nilufer 16059, Bursa, Turkey

ARTICLE INFO

Keywords:
TSP
Drone
Drone delivery
Decomposition
Iterative algorithm
MIP

ABSTRACT

This study investigates a new delivery problem that has emerged after the attempts of several e-commerce and logistics firms to deploy drones in their operations to increase efficiency and reduce delivery times. In this problem, a delivery truck that carries a drone on its roof serves customers in coordination with a drone. The drone is considered to complement the truck due to its cost-efficiency and ability to access difficult terrains and to travel without exposure to congestion. This study presents an iterative algorithm that is based on a decomposition approach to minimize delivery completion time. In the first stage of the proposed methodology, the truck route and the customers assigned to the drone are determined. In the second stage, a mixed-integer linear programming model is solved to optimize the drone route by fixing the routing and the assignment decisions that are made in the first stage. Beginning with the shortest truck route, the assignment and the routing decisions are iteratively improved. The solution times of our algorithm are compared with the solution times of the state-of-the-art formulations that are solved by CPLEX. The results demonstrate that our algorithm yields shorter solution times for the instances that we generated with the specified parameters. An optimization-based heuristic algorithm, which obtains solutions for medium-sized instances, is developed by reducing the feasible search area.

1. Introduction

Drones, which are predominantly defined as small unmanned aerial vehicles, are special systems that operate without human operators. Due to the technological developments in these systems, drones have received an increasing amount of attention in various areas, such as the logistics and retail industries, which seek to integrate drones in their operations.

Amazon is the first company to announce an ongoing project to deploy drones for last-mile delivery (Amazon.com). In Amazon's business model, drones are aimed at directly delivering parcels from a depot to the customers. However, due to technical restrictions, such as the battery life limit and parcel weight, drones are restricted to visiting one customer during each flight and returning to the depot after each customer visit (French, 2015). The drone that returns to the warehouse can only perform the next customer visit after its battery is changed and the next customer's package is loaded. In addition to the technical restrictions, which require improvements in drone technologies, federal regulations (Snider and Welch, 2015) and the issues such as safety and violation of privacy, have prompted criticisms and discussions (Paul, 2015). These criticisms and discussions have caused the emergence of a new last-mile delivery concept that suggests synchronization of a truck and a drone (Wohlsen, 2014). In this delivery concept, a drone that is launched from and picked up by a truck that is equipped to carry a drone on top of the truck concurrently serves customers with the

☆ This article belongs to the Virtual Special Issue on Operations & automation.

* Corresponding author.
E-mail address: esyurek@uludag.edu.tr (E. Es Yurek).

https://doi.org/10.1016/j.trc.2018.04.009
Received 1 July 2017; Received in revised form 7 April 2018; Accepted 9 April 2018
Available online 19 April 2018

truck. This new delivery approach has received a substantial amount of interest. Mercedes-Benz has recently announced a "drone-equipped delivery van" concept vehicle that employs "roof-mounted unmanned aerial vehicles (UAVs) and robotic package-sorting devices" (Banker, 2016). UPS, which has attempted to realize this concept in real life, has announced that it successfully tested the coordinated distribution of a drone and a truck by modifying its traditional delivery truck such that it interacts with the drone (Crowe, 2017).

Inspired by this delivery approach, Murray and Chu (2015) introduced the traveling salesman problem with drone (TSP-D), which they refer to as "the flying sidekick traveling salesman problem". Their motivation for this operational logistics problem is to investigate the efficiency of this approach, which combines the advantages of vehicles with contrasting features. The traditional delivery vehicle-the truck-has a large load capacity and does not return to the depot before visiting all customers in the delivery network; however, it may be limited in the number of customers it visits due to the land conditions. In urban areas, congestion is a factor that delays this type of vehicle. Although a drone has a load capacity of exactly one, it can easily handle the land conditions and does not have to track the roads. These factors, which delay a truck, are advantages of a drone.

In this paper, we focus on improving the mathematical models suggested for the TSP-D. Since this problem is a recent variant of the traditional traveling salesman problem, a limited number of studies address this topic. The mathematical models developed for the TSP-D are not capable of solving instances with more than 10 customers in a one-hour computational time limit (Agatz et al., 2016; Murray and Chu, 2015). To improve this performance, we propose an algorithm in which an MIP formulation is iteratively solved. This paper proposes a simple heuristic that improves the solution time.

The main contributions of our paper can be summarized as follows:

- We develop an iterative algorithm by decomposing the problem into two stages and solving an MIP model in the second stage of each iteration.
- We compare the solution times of our algorithm with the CPLEX solution times of the state-of-the-art mathematical models. The main contribution of our algorithm is to solve 12-customer nodes problem instances (uniformly generated) within 15 min on average.
- We propose an optimization-based heuristic that can solve instances with 20 customers.

The remainder of the paper is structured as follows: In Section 2, we present a review of the literature on the TSP-D. In Section 3, we describe the problem. The exact algorithm and the proposed heuristic are explained in Section 4. In Section 5, a detailed analysis of the numerical results is provided. Section 6 presents the conclusion.

2. Literature review

An increasing number of studies investigate the efficiency of delivery systems that deploy drones. However, a limited number of these studies focus on the coordinated delivery of a truck and a drone. Since we address the synchronization of both vehicles in our study, only the studies with similar assumptions are considered.

Murray and Chu (2015) is the first study that introduces the TSP-D. They study two different variants of the problem; one of these variants is the focus of this paper. They propose mathematical models and heuristics for both of the problems. The heuristic that is proposed for the problem, which is the subject of this study, is based on a route first-cluster second heuristic. Murray and Chu (2015) construct their heuristic approach on the TSP solution. By obtaining a route for the truck, they partition the truck route into drone tours regarding the cost savings. They present heuristic solutions for the instances with only 10 customers.

A related study by Agatz et al. (2016) generates combinations of truck and drone routes between each possible launch and pickup nodes. They refer to each combination as an "operation" and propose an operation-based formulation. They develop two heuristics based on local search and dynamic programming. Unlike the assumptions made by Murray and Chu (2015), the truck is allowed to meet with the drone at the starting node of the flight.

Ha et al. (2018) investigate the delivery cost of the TSP-D for the first time. They provide two different heuristics that gain inspiration from the route first-cluster second heuristic, which is based on local search and GRASP. They provide the results obtained for large instances with 50 and 100 customer nodes. In another study by Ha et al. (2015), they propose a cluster first-route second heuristic, in which they initially solve a mathematical model that optimally maximizes the path traveled by the drone and then heuristically build the truck route. Another heuristic approach that they propose is based on a route first-cluster second approach. In this version of the heuristic approach, they initially solve the TSP, then partition the truck route into clusters.

The only study that uses metaheuristic is proposed by Ponza (2016). In his thesis, he applies simulated annealing to the TSP-D.

The literature that we briefly mentioned focuses on the traveling salesman problem with a single drone. Wang et al. (2017) is the first study to consider multi-trucks and multi-drones. They investigate this version of the problem from a theoretical aspect that provides worst-case analysis and bounds for several considerations. As an extension, Poikonen et al. (2017) enhance the proposed theoretical investigation.

Ferrandez et al. (2016) investigate the efficiency of not only delivery time but also energy in a truck and drone coordinated delivery system. They consider a single truck and multiple drones in a different version of the TSP-D. In their consideration, the truck carries multiple drones; when it arrives at the launch node, each drone is launched from the truck for parallel visits and returns to the truck as it completes the visits. The customer at the launch node is served by the truck while it waits for the drones. In their study, they optimize the number of launch nodes by Newton's approach then determine the launch locations utilizing a K-means algorithm. The truck route is determined by a genetic algorithm.

3. Problem description

In the TSP-D, a truck and a drone deliver parcels in coordination. Each customer demand has to be satisfied by either the truck or the drone. Some customers may not be served by the truck due to difficult terrains. Conversely, some customers have to be served by the truck due to regulations or physical restrictions. At the beginning of the delivery, all parcels are loaded into the truck. Since the load capacity of the drone is exactly one and the battery life is limited, the drone has to meet with the truck after each customer visit to load the parcel of the next customer and refresh its battery. Once the vehicles meet, the operator of the truck loads the parcel of the next customer on the drone and refreshes its battery. Then, the drone either immediately travels to the next drone customer or remains on the truck. Between any two customer nodes that are assigned to the truck, the truck travels either alone or with the drone since the truck is equipped to carry the drone. Thus, a customer node is visited either by the drone, the truck or the truck with the drone on top of the truck. While the drone separately performs its task from the truck, the truck keeps moving along its route toward the customer node where they will meet. The truck is allowed to visit customers along its individual route. The first vehicle that arrives at the meeting node waits for the other vehicle. The routes of both vehicles have to start and end at the depot. The objective of this problem is to minimize the delivery completion time.

We illustrate this delivery approach in Fig. 1. This figure involves three types of nodes with different shapes and two types of arcs. The customer nodes in the triangle denote the customers that are assigned to the drone, whereas the nodes that are visited by the truck are denoted by the circle. In addition to the nodes in the circle and triangle, the nodes in the square indicate the nodes at which a drone flight starts or ends. In the remainder of this study, we refer to the nodes given in the square as "combined nodes", which were first defined by Agatz et al. (2016). The drone route is distinguished from the truck route by the dashed line. In the network illustrated in Fig. 1, the vehicles separately start from the depot. Prior to meeting at Node 4, the drone serves Node 10. At Node 4, the truck operator not only serves the customer but also refreshes the battery of the drone and loads the parcel of Node 6 on the drone. After replenishing the drone at Node 4, the truck visits Node 2 and Node 7, respectively, and meets the drone at Node 9. While the truck moves from Node 4 to Node 9, the drone visits Node 6 and returns to the truck at Node 9. The vehicles separately depart from Node 9, and the truck visits Node 1 before arriving at the meeting node-Node 5. The drone travels the tour *9-3-5*. The drone returns to the depot after visiting Node 8, and the truck directly goes to the depot from Node 5. Note that the customers at which the vehicles meet (i.e., Node 4, Node 9 and Node 5) are served by the truck.

The main difficulty of this problem is attributed to the interdependency between the two vehicles. A classification of the synchronization constraints that cause this interdependency is provided by Drexl (2012). According to the given classification, the TSP-D has three types of synchronization constraints. First, "task synchronization" is required because every customer has to be served exactly once by exactly one of the vehicles. Second, "operation synchronization" is another challenge. If the drone is assigned to visit more than one customer, then the drone has to realize multiple tours. Thus, once the drone is launched for a tour, it has to meet with the truck prior to performing the next tour. The starting time of the next tour depends on the arrival time of the truck at the meeting node, which requires coordination between the vehicles in both time and location. Last, the case in which the vehicles may travel separately or in tandem along their routes inspires the "movement synchronization".

Three important decisions are made in this problem:

- Which customers will be served by which vehicle?
- For the customers selected for drone delivery, which customer nodes will be assigned as the launch and the pickup nodes? Which

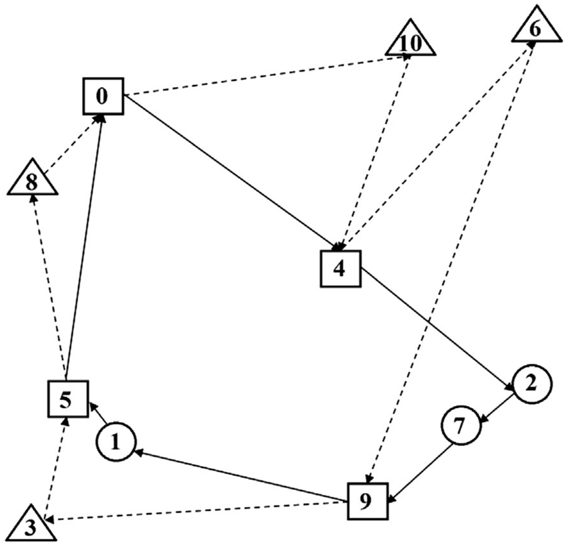

Fig. 1. Illustrative example for the TSP-D solution.

drone tours will be flown?
- In which order will the truck visit the customers that are assigned to the truck?

Murray and Chu (2015) made some assumptions about the operating conditions. These assumptions are also employed in this study:

- A drone tour is only allowed to start and end at customer nodes; the drone is not allowed to depart from/meet with the truck in a position that is not a customer node.
- The drone is not allowed to return to the same customer node it launches from.
- When the drone returns to the depot, it is not allowed to depart for another tour.

Murray and Chu (2015) incorporate replenishment times into their mathematical model. Since these operation times can be easily incorporated in our study, we do not to consider them for simplicity.

4. Methodology

This section presents an iterative algorithm that decomposes the problem into two stages. As we stated in the previous section, three types of decisions are made in this problem. These decisions are not independent of each other due to the synchronization requirements. The interdependence of these decisions enables us to determine the value of the upper-level decision variables in the first stage and then optimize the remaining lower-level problem in the second stage by fixing the upper-level decision variables.

In our two-stage algorithm, we determine the truck route in the first stage and the drone route in the second stage. Determination of the truck route also provides the customer assignments because the customer nodes that are not located along the truck route have to be served by the drone. Therefore, at the end of the first stage, we obtain the assignment decisions and the truck route. In the second stage, we determine the drone route considering the fixed truck route and drone nodes. To determine the drone route, the launch and the pickup nodes have to be determined for each drone node that is obtained in the first stage. This task is achieved by the solution of an MIP formulation.

This two-stage solution approach is illustrated in Fig. 2. Two different solutions for the same network are shown. In both of the solutions, the drone nodes are represented by triangles, whereas the truck-only nodes are represented by circles and the combined nodes are represented by squares. The solution on the left side of Fig. 2 represents the decision made in the first stage. In this stage, in addition to the customer assignments, we also provide the truck route. Fixing the truck route determined in the first stage, we solve the second stage problem, and the drone tours are determined, as demonstrated on the right side of Fig. 2. In the delivery network in Fig. 2, the first-stage problem determines the truck route 0-4-2-7-9-1-5-0, whereas the second-stage problem determines the drone tours 0-10-4, 4-6-9, 9-3-5 and 5-8-0.

4.1. Iterative algorithm

The proposed solution approach is motivated by the consideration that the drone should visit as many customers as possible to obtain a significant decrease in the truck route. In the first stage of each iteration, the truck route is determined. Then, the drone tours are optimized to minimize the truck's waiting time in the second stage of each iteration. However, assigning a maximum number of drone nodes may not always yield optimal solutions. Assigning the maximum number of drone nodes causes truck to visit fewer customers along an individual truck route. In this case, the truck is likely to complete the tour earlier than the drone and then wait for

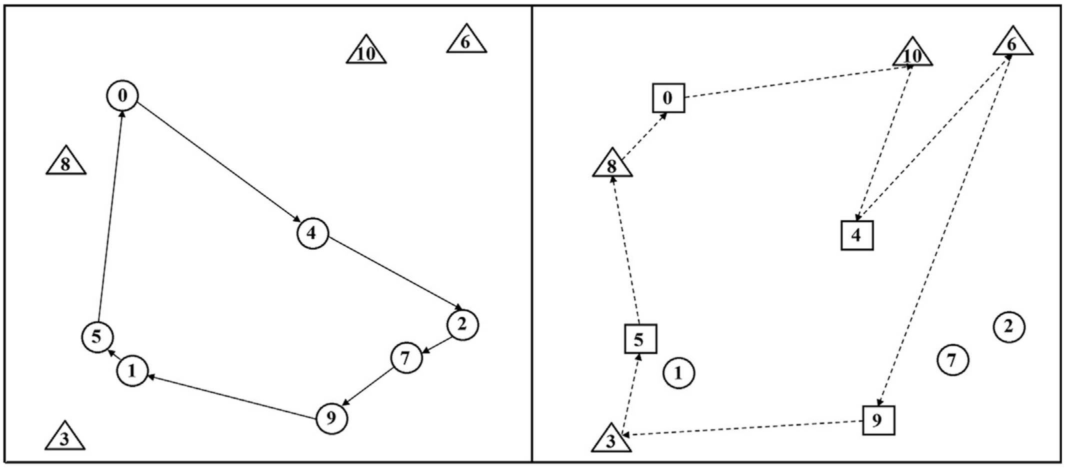

Fig. 2. Illustrative example for the first-and second-stage solutions of the proposed algorithm.

the drone. The truck should visit additional customers on its individual route rather than immediately arriving at the pickup node and waiting for the drone. The related observations obtained from the results of the experimental study are provided in Section 5.2.

To determine the truck route, we generate all possible truck routes instead of solving any formulations in the first stage of each iteration. Excluding each possible combination of drone nodes, we generate all possible truck routes that involve the remaining nodes and constitute R (the set of truck routes). However, we only generate the truck routes in which the index of the last customer node is larger than the index of the first customer node. To illustrate this point, 0-1-2-3-4-5-0 and 0-5-4-3-2-1-0 are equivalent routes with contrasting directions. To reduce the number of feasible truck routes, we eliminate the second route and prevent the iterations from regenerating the same route. With this approach, we only generate one half of the possible truck routes. This approach applies the assumption of symmetric distance and cannot be applied in cases of asymmetric distance. In Section 5.2, detailed results related to asymmetric distance assumption are provided. We retain the generated truck routes with shorter route durations than the global upper bound (*gub*), which is initially the TSP solution, in set R in ascending order of duration. To reduce the number of generated truck routes, we generate the routes in two steps. First, we generate all possible truck routes by excluding the possible combinations of the maximum number of drone nodes, which is denoted by D_{max}. Solving the second-stage problem for the shortest truck route, if the solution is smaller than the TSP solution, we obtain a new global upper bound. Second, we generate the remaining truck routes, which involve more nodes than $(N-D_{max})$, and retain the routes that are smaller than *gub*. Thereby, if *gub* is improved, we retain fewer routes in set R.

The TSP solution provides an upper bound for the total delivery time at the beginning of the algorithm. Therefore, we begin by solving the TSP, and *gub* is initially assigned as the TSP solution. Beginning with the shortest truck route, we sequentially investigate the truck routes in set R. Since the truck routes in R are listed in ascending order, in each iteration, the selected truck route provides a lower bound of the current iteration (lb_{iter}). The total delivery time is composed of the traveling and waiting times of the truck. Therefore, the waiting time in each iteration is obtained for the selected truck route by solving the second stage formulation. If the total delivery time obtained at any iteration is smaller than *gub*, then *gub* is updated. If the truck route that is selected at any iteration has a route duration larger than *gub*, then the optimum solution is obtained because any truck route that is shorter than the current truck route cannot be obtained in subsequent iterations. Otherwise, a better solution is likely obtained. The iterations are performed until the lower bound of the current iteration (lb_{iter}) exceeds the best obtained solution (*gub*). In this case, the algorithm terminates and the optimum solution is obtained.

The mathematical model that we develop to solve the second-stage problem is a tour-based formulation. A drone tour consists of launch and pickup nodes that are involved in the truck route and the drone node that is excluded from the truck route. Therefore, the set of drone tours in each iteration has to be constituted based on the truck route and drone node assignments that are determined in the first stage of this iteration. The battery life limit has to be considered while generating drone tours. Only the drone tours with shorter durations than the battery life limit are retained in the set of drone tours.

A pseudocode of the proposed algorithm is provided in Algorithm 1. For a better understanding, the algorithm is illustrated in an example. Assume that we have a delivery network with four customers. First, we obtain the TSP solution which is 35.56 and *gup* is initially 35.56. Since the maximum number of customers that can be assigned to the drone is two (*floor* $((4+1)/2)$), we generate the truck routes with two customer nodes and retain the routes with smaller delivery durations than *gub* in ascending order in set R. The truck routes that are retained in R are reported in Table 1. The route with the minimum length (11.45), which is the first element of set R (denoted by $R(1)$), is selected, and the second-stage formulation is solved. We obtain the waiting time of the truck, which is 12.02. The total delivery time obtained at the end of the first iteration is 23.47, which is smaller than the TSP solution. The global upper bound is updated as 23.47. Then, the remaining truck routes with three customers are generated. We retain the generated routes that have smaller delivery times than the updated *gub*. Thus, only three truck routes are inserted into set R. The truck routes that will be investigated for better solutions are indicated in the third column of Table 2, with the route number in the second column. Table 2 lists all routes in ascending order. For all iterations and solutions, the lower and global upper bounds and updates are explicitly represented in Table 2. Since the obtained solutions are worse than the best solution, *gub* is not updated at iterations 3 and 5. As shown in bold in Table 2, the optimum solution is obtained at iteration 4 but the algorithm terminates at iteration 6 because the lower bound of iteration 6 is larger than *gub*. This finding indicates that a better solution smaller than 17.66 cannot be achieved because the truck route already exceeds the best solution.

Table 1
Illustrative example of generation of truck routes.

Route no.	Route	Route duration (min)
5	0-2-4-0	11,45
4	0-2-3-0	11,76
6	0-3-4-0	12,49
1	0-1-2-0	27,82
2	0-1-3-0	30,67
3	0-1-4-0	32,56

Table 2
Illustration of the iterative algorithm.

Iteration	Route no.	Route	Route duration (min)	lb_{iter}	Waiting time	Solution	gub	TSP
1	5	0-2-4-0	11,45	11,45	12,02	23,47	23,47	35,56
2	4	0-2-3-0	11,76	11,76	11,49	23,25	23,25	35,56
3	6	0-3-4-0	12,49	12,49	12,17	24,66	23,25	35,56
4	7	0-2-4-3-0	14,45	**14,45**	**3,21**	**17,66**	17,66	35,56
5	8	0-3-2-4-0	16,51	16,51	1,16	17,67	17,66	35,56
6	9	0-2-3-4-0	18,25	18,25	*	*	*	35,56
*	1	0-1-2-0	27,82	*	*	*	*	35,56
*	2	0-1-3-0	30,67	*	*	*	*	35,56
*	3	0-1-4-0	32,56	*	*	*	*	35,56

* Results are not obtained since the algorithm terminates.

Algorithm 1.

Step 1. $iter \leftarrow 1$

Step 2. Solve TSP to obtain the global upper bound, $v(TSP)$
Step 3. $gub \leftarrow v(TSP)$
Step 4. Generate all possible truck routes with $(N-D_{max})$ nodes, and if (route duration $<$ gub),
Step 5. Keep each route in ascending order in set R
Step 6. Set the lower bound as the shortest route duration which is the first route in set R, $lb_{iter} \leftarrow R(iter)$
Step 7. Generate all possible drone tours; set P_{iter}
Step 8. Solve the second-stage mathematical model to obtain the waiting time of the truck $v(Z)$
Step 9. If $(v(Z)+ lb_{iter} < gub)$,
Step 10. $gub \leftarrow v(Z)+ lb_{iter}$
Step 11. Generate all possible truck routes that involve more nodes than $(N\text{-} D_{max})$, and if (route duration $<$ gub),
Step 12. Insert each route in ascending order into set R
Step 13. $iter \leftarrow iter+1$
Step 14. $lb_{iter} \leftarrow R(iter)$
Step 15. If $(lb_{iter} > gub)$
Step 16. Terminate
Step 17. Else
Step 18. Generate all possible drone tours; set P_{iter}
Step 19. Solve Z to obtain optimum drone tours
Step 20. If $(v(Z) + lb_{iter} < gub)$,
Step 21. $gub \leftarrow v(Z)+ lb_{iter}$
Step 22. Go to Step 13

4.2. Mathematical model

The mathematical model that we develop to optimally determine the drone tours is presented in this section. The definitions of the indexes, sets, parameters and decision variables are summarized in Table 3.

In this study, we propose to solve TSP-D in an iterative manner by decomposing it into two stages. In each iteration, we fix the truck route in the first stage and denote it by $\bar{r}_{iter}$. The determination of the truck route also provides the drone assignments, because the customer nodes that are not along the truck route have to be served by the drone. The customer nodes that are assigned to the drone in iteration $iter$ are denoted by D_{iter}. In the second stage of each iteration, the proposed mathematical model is solved to determine the optimum tours. As previously mentioned, a tour is composed of three nodes: launch node, drone node and pickup node. Since a drone flight starts and ends at the truck, the first node that represents the launch node and the last node that represents the pickup node are also included in the truck route. However, the drone node has to be an element of D_{iter}. A drone tour is not allowed to start and end at node i, $i \in D_{iter}$. In addition, the duration of any drone tour cannot exceed the drone battery life. To reduce computational effort, we generate the set of feasible drone tours P_{iter} by considering these feasibility conditions instead of embedding them into the model as feasibility constraints and propose the following tour-based formulation.

The truck and the drone simultaneously depart from the launch node for their individual tours. Both vehicles may arrive at the pickup node either at the same time or at different times depending on the duration of their individual routes. The duration of the truck route will not change because the truck does not wait when the drone arrives before the truck or both vehicles complete their visits at the same time. However, if the truck arrives at the pickup node before the drone, it waits for the drone. This schedule generates additional time for the truck to complete its route and return to the depot. Therefore, our purpose in the second stage is to

Table 3
Indexes, sets, parameters and decision variables.

Indexes	
i, j	Node
k	Position
p	Drone tour
iter	Iteration
Sets	
C	Set of customer nodes
D_{iter}	Set of customer nodes assigned to the drone in the first stage of iteration *iter*
P_{iter}.	Set of drone tours generated at iteration *iter*
Parameters	
d_p	Duration of tour p
f_{ip}	Binary parameter which takes value 1 if tour p starts from node i and 0 otherwise
a_{ip}	Binary parameter which takes value 1 if node i is served by the drone in tour p and 0 otherwise
l_{ip}	Binary parameter which takes value 1 if tour p ends at node i and 0 otherwise
D_{max}	Maximum number of customer nodes that can be assigned to the drone
N	Number of customer nodes in the delivery network
t_i	Arrival time of the truck at node i
m_k	Node assigned to position k in the truck route
$\bar{r}_{iter}$	The truck route decided in the first stage of iteration *iter*
$\{0,C + 1\}$	The depot as the source and the sink nodes
Decision variables	
x_p	Binary variable which takes 1 if tour p is traveled by the drone and 0 otherwise
w_i	Waiting time of the truck at node i

minimize the waiting time of the truck at the pickup nodes. Since the candidate pickup nodes are the customer nodes that are served by the truck, we can represent the objective function of the second stage problem as expressed in Eq. (1). The vehicles are allowed to separately return to the depot. Thus, the depot is a candidate pickup node and is included in the objective function.

$$(Z) \quad \min \sum_{i \in C \setminus D_{iter} \cup \{C+1\}} w_i \tag{1}$$

Eq. (2) computes how long the truck waits for the drone at each pickup node that it visits. Since we fix the truck route in the first stage, the truck's arrival times at each node are exactly known. We can easily compute the waiting times. In Eq. (2), the inner parentheses represent the traveling time of the truck between the launch and the pickup nodes. If the drone tour is longer than the individual truck tour, then the truck waits for the drone for a period as long as w_i.

$$\sum_{p \in P_{iter}} \{d_p l_{ip} x_p - (t_i l_{ip} x_p - \sum_{j \in C \setminus D_{iter} \cup \{0\}} t_j f_{jp} x_p)\} \leqslant w_i, \quad i \in C \setminus D_{iter} \cup \{C+1\} \tag{2}$$

To ensure feasible drone tours, some conditions must be satisfied. The first condition is presented in Eq. (3), such that for each customer node that will be visited by the drone, exactly one tour has to be assigned.

$$\sum_{p \in P_{iter}} a_{ip} x_p = 1, \quad i \in D_{iter} \tag{3}$$

In the second condition, node i, which is an element of route $\bar{r}_{iter}$, can be selected as the launch node of only one tour.

$$\sum_{p \in P_{iter}} f_{ip} x_p \leqslant 1, \quad i \in C \setminus D_{iter} \cup \{0\} \tag{4}$$

Similarly, node i, which is an element of route $\bar{r}_{iter}$, can be selected as the pickup node at most once.

$$\sum_{p \in P_{iter}} l_{ip} x_p \leqslant 1, \quad i \in C \setminus D_{iter} \cup \{C+1\} \tag{5}$$

The last feasibility condition is represented in Eq. (6). Let the truck route fixed in the first stage has n customer nodes. According to the following equation, if a tour starting at node i and ending at node j is selected, a flight cannot start or end at any position k between node i and node j. While on a tour, the drone cannot be re-launched from the truck before returning to the truck. Fig. 3 is provided to illustrate this type of infeasibility, which we overcome in Eq. (6).

$$\sum_{p \in P_{iter}} f_{m_k p} x_p + \sum_{p \in P_{iter}} l_{m_k p} x_p \leqslant 2\left(1 - \sum_{p \in P_{iter}} f_{m_i p} l_{m_j p} x_p\right), \quad i = 0,1,\ldots,n-1, j = i+2,\ldots,C+1, k \in H = \{h | i < h < j\}, \quad i \neq j \tag{6}$$

Definitions of decision variables are provided in Eqs. (7) and (8).

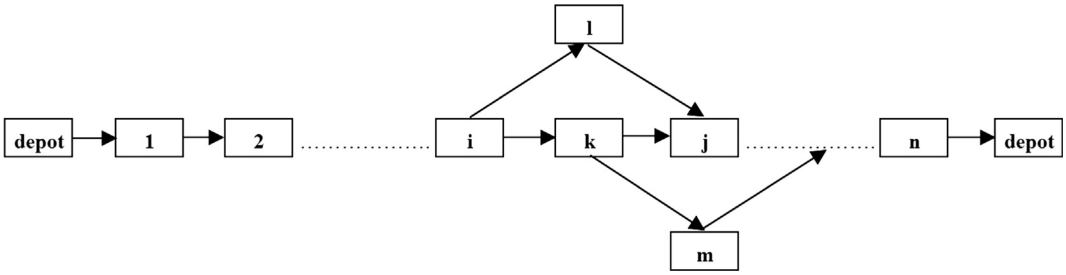

Fig. 3. Illustration of infeasible drone tours.

$$x_p \in \{0,1\}, \quad p \in P_{iter} \tag{7}$$

$$w_i \geqslant 0, \quad i \in C \backslash D_{iter} \cup \{C+1\} \tag{8}$$

4.3. Heuristic algorithm

As we previously expressed, to constitute set R, we exclude all possible drone node assignments and generate all possible truck routes that involve the remaining customer nodes for each assignment. This process entails enumeration, in which the number of elements in set R exponentially increases. Solving the problem will become more difficult as the number of customer nodes increases. Due to the combinatorial structure of this problem, we propose the application of a simple heuristic to reduce the number of generated truck routes. For each possible combination of drone nodes, we generate only one truck route using the nearest neighborhood approach and eliminate the remaining truck routes that involve the same customer nodes in different sequences. Assume that we have five customer nodes. One of the possible drone node assignments is node 1 and node 2. In this case, six different truck routes with the same customers can be generated: 0-3-4-5-0, 0-3-5-4-0, 0-4-3-5-0, 0-4-5-3-0, 0-5-3-4-0 and 0-5-4-3-0. Since we assume symmetric distances, we retain the first three routes in the exact solution algorithm. Instead of retaining all three routes in set R, we generate the truck route that involves nodes 3, 4 and 5 considering the nearest neighborhood approach.

Let N denote the number of customer nodes. In the worst case, without the proposed heuristic, the number of elements in R can be computed as follows:

$$\binom{N}{1}(N-1)!/2 + \binom{N}{2}(N-2)!/2 + \cdots + \binom{N}{D_{max}}(N-D_{max})!/2$$

The number of routes generated by the proposed heuristic can be calculated as follows:

$$\binom{N}{1} + \binom{N}{2} + \cdots + \binom{N}{D_{max}}$$

The dramatic reduction accomplished by this heuristic is distinct. In Section 5.3, the heuristic results are presented, and the performance of the heuristic is evaluated.

5. Computational study

In this section, we present a computational study to evaluate the performance of the proposed algorithm. To perform this evaluation, we generate test instances. The instance generation and the experimental setup are described in Section 5.1. Section 5.2 presents the performance comparison with existing studies in terms of solution time and a detailed analysis of the results obtained by our proposed algorithm for various instance types. In the last part, we evaluate the performance of the proposed heuristic algorithm in terms of both solution quality and time.

5.1. Experimental setup

Since the TSP-D has been recently introduced to the literature, widely accepted benchmark data are not available. Bouman et al. (2015) provide a set of instances for the TSP-D with a maximum of nine customers for small-scale problems. Therefore, we perform computational experiments for a variety of instances that we randomly generated with various problem sizes. We initially create instances with 10 customers. Because each case is optimally solved within a reasonable runtime, we increase the problem size by one until the solution time exceeds the specified time limit, which is one hour. In this way, the instances with 10, 11 and 12 customers are generated for the performance evaluation of the exact algorithm. Additionally, we evaluate the heuristic performance of the instances with 20 customers. Ten instances for each problem size are created.

We conduct our experimental study using three different types of instances, in which customers are variously distributed in the delivery network. In the first type of instance, the coordinates of the customer nodes are uniformly distributed between 0 km and 10 km; therefore, a delivery network of 100 km^2 is constituted. For the generation of the second type of instance (centered) and third

Table 4
Computational results obtained for uniform instances.

	Runtime (sec.)			Opt (min.)											
Case $C = 10$	Murray et al.	Agatz et al.	Es Yurek et al.	*v(Z)*	Travel. time of truck	Travel. time of drone	Wait. time of truck	Wait. time of drone	TSP	Initial LB	Initial UB	# of truck routes	Iter.	Route gen. time	# of D.
1	3600*	899	238	32.04	30.31	29.77	1.73	2.26	49.94	16.39	45.68	27351	1262	3	4
2	3600*	860	27	41.83	41.74	38.61	0.08	3.22	60.99	34.35	49.54	5074	124	2	4
3	3600*	822	161	37.45	35.59	37.45	1.86	0.00	59.04	23.94	53.42	24692	825	4	3
4	3600*	915	51	45.46	43.52	44.20	1.94	1.25	63.02	33.32	63.02	9784	260	1	4
5	3600*	1089	41	39.65	39.65	33.41	0.00	6.24	59.34	29.10	41.03	7126	207	1	3
6	3600*	994	6	41.36	41.36	36.29	0.00	5.07	60.51	35.18	59.67	14020	26	2	4
7	3600*	902	24	31.60	30.60	27.75	1.00	1.32	47.27	27.32	45.64	26245	102	4	4
8	3600*	1148	21	37.84	37.34	33.04	0.50	4.79	60.30	25.86	60.30	39752	62	10	5
9	3600*	1064	10	33.56	33.12	32.56	0.44	1.00	43.26	28.32	45.82	5158	52	2	4
10	3600*	1018	34	37.24	37.24	35.10	0.00	2.14	51.68	32.07	46.68	4674	165	1	3
Avg 1–10		971.1	61.3	37.80	37.05	34.82	0.75	2.73	55.54	28.59	51.08	16287.60	308.50	3	3.8
$C = 11$															
11	3600*	3600*	412	38.12	37.96	33.16	0.16	2.08	47.54	14.55	47.54	28263	2436	8	4
12	3600*	3600*	736	38.70	38.70	28.55	0.00	4.22	59.06	24.47	59.06	168396	1600	479	4
13	3600*	3600*	221	39.86	38.83	32.36	1.03	3.64	56.62	28.42	56.62	58152	1034	32	4
14	3600*	3600*	367	38.37	37.42	36.15	0.95	2.22	56.28	23.82	56.28	94370	1532	119	5
15	3600*	3600*	150	35.68	34.26	32.23	1.43	3.43	53.87	23.45	53.87	79127	434	76	5
16	3600*	3600*	534	36.08	36.08	31.04	0.00	5.03	55.29	22.04	55.29	101657	1901	137	2
17	3600*	3600*	156	37.38	36.54	35.02	0.84	0.63	56.83	27.89	56.83	88984	319	101	4
18	3600*	3600*	184	35.51	33.21	33.08	2.30	2.42	130.98	22.36	47.85	39851	1080	10	4
19	3600*	3600*	172	40.23	38.45	39.28	1.79	0.95	58.43	21.11	58.43	60476	843	37	5
20	3600*	3600*	59	43.60	42.81	40.35	0.80	3.26	60.84	35.07	60.84	27688	241	9	5
Avg 11–20			299.1	38.35	37.43	34.12	0.93	2.79	63.57	24.32	55.26	74696.4	1142	100.8	4.2
$C = 12$															
21	3600*	3600*	514	45.92	45.57	36.20	0.34	9.72	54.72	28.47	54.72	45924	2667	69	5
22	3600*	3600*	686	44.78	44.78	40.25	0.00	4.53	64.15	32.58	64.15	162658	496	575	4
23	3600*	3600*	372	47.65	46.75	37.59	0.90	10.06	63.24	38.04	63.24	88641	1037	162	5
24	3600*	3600*	1316	44.78	42.53	44.30	2.25	0.05	72.00	35.50	60.58	254995	1278	1056	5
25	3600*	3600*	905	43.96	43.18	43.08	0.78	0.88	56.44	26.66	56.44	68729	3776	108	5
26	3600*	3600*	413	34.03	34.03	31.99	0.00	2.05	51.04	27.02	41.82	74498	1671	85	3
27	3600*	3600*	1853	38.37	36.17	38.24	2.20	0.13	52.14	27.47	48.77	260771	5137	1233	5
28	3600*	3600*	413	42.38	39.37	41.11	3.02	0.56	57.89	29.77	57.89	85760	1331	156	3
29	3600*	3600*	216	41.31	41.28	38.85	0.03	2.46	55.27	33.05	55.27	60237	642	88	4
30	3600*	3600*	914	41.95	41.76	37.80	0.19	3.63	58.85	32.00	50.27	63130	4094	73	5
Avg 21–30			760.2	42.51	41.54	38.94	0.97	3.41	58.57	31.06	55.32	116534.3	2213	360.5	4.4

* CPLEX is terminated due to runtime limit. Reported results demonstrate the runtime limit.

type of instance (clustered), we are inspired by Agatz et al. (2016). They generate the depot and the customer nodes such that each node is located at the distance r from the center (0, 0) with the angle a. Thus, the x coordinate of a location is $rcos(a)$, and the y coordinate of a location is $rsin(a)$. We generate the distance r with the parameters N (0, 5) and the angle a with the parameters U (0, 2π). To generate clustered data, we locate two centers at (0, 0) and (10, 0).

We assume that the truck speed is 40 km/h and the drone speed is 56 km/h. The drone's battery life is limited to 20 min. These parameters are consistent with the parameters in the studies of Murray and Chu (2015), Ha et al. (2015) and Ha et al. (2018). To calculate the distance traveled by the vehicles, Murray and Chu (2015) and Ha et al. (2018) employ Manhattan metric for the truck and the Euclidean metric for the drone. Since this approach seems reasonable for distance calculation, we employ the same approach in our study.

The proposed algorithm was coded in C+ +, and ILOG CPLEX Concert Technology 12.6.3 was employed for the mathematical models. All experiments were run on an Ultrabook with Intel Core i7-7500U CPU with 2.90 GHz and 16 GB RAM.

5.2. Performance evaluation of exact algorithm

Two basic formulations are proposed for the TSP-D in the literature. The first formulation is developed by Murray and Chu (2015), and the second formulation is developed by Agatz et al. (2016). The formulation provided by Ponza (2016) is based on the formulation presented by Murray and Chu (2015) with a slight contribution. Ha et al. (2018) and Ferrandez et al. (2016) investigate the problem from a significantly different perspective. Therefore, comparing the solution times of our decomposition-based iterative algorithm with the CPLEX solution times of the formulations provided by Murray and Chu (2015) and Agatz et al. (2016) is

Table 5
Ratio of the truck route generation time to the total solution time.

Case	Route generation time * 100/total time		
	$C = 10$	$C = 11$	$C = 12$
1	1.26	1.94	13.42
2	7.41	65.08	83.82
3	2.48	14.48	43.55
4	1.96	32.43	80.24
5	2.44	50.67	11.93
6	33.33	25.66	20.58
7	16.67	64.74	66.54
8	47.62	5.44	37.77
9	20.00	21.51	40.74
10	2.94	15.25	7.99
Avg	13.61	29.72	40.66

reasonable. To make this comparison, we coded our algorithm and the mathematical models in the related literature and run them for the same instances that we generated. The solutions that we obtained for the uniform cases with 10, 11 and 12 customer nodes are summarized in Table 4. The second, third and fourth columns in the table demonstrate the runtimes that are required to optimally solve the formulation by Murray and Chu (2015), the formulation by Agatz et al. (2016) and our algorithm, respectively, in seconds. However, the runtimes by Murray and Chu (2015) indicate that we terminated the optimization due to violation of the runtime limit, which is an hour. Thus, their formulation yielded the optimum in none of the ten instances within the one-hour runtime limit. The proposed algorithm obtained the optimum solutions in runtimes that were smaller than the runtimes obtained for the other formulations. Although the formulation provided by Agatz et al. (2016) provided the optimum in all instances with 10 customers, the average runtime of their formulation is approximately one quarter of an hour. This time is nearly fifteen times the average runtime of our algorithm, which is approximately one minute. The problem size of 10 customers is the largest size for which we obtained the optimum solutions by solving the mathematical model proposed by Agatz et al. (2016) within the one-hour time limit. In contrast to the formulations provided by Agatze et al. (2016) and Murray and Chu (2015), the algorithm that we proposed obtained the optimum solutions, even for the instances with 12 customers within an average runtime smaller than 15 min. Table 4 lists the results that we obtained for the instances that involve 10, 11 and 12 customers with the assumption that the drone speed is 56 km/h. Our algorithm requires more than one hour to solve the problems with 13 or more customers. For the test cases that we created, the proposed algorithm outperforms the existing studies in terms of the exact solution times.

As clearly listed in Table 4, the average solution time of the algorithm increases with the number of customers. The increase in runtime is primarily attributed to the increase in the number of generated truck routes ("# of truck routes" column), which is dependent on the problem size. The increase in the number of the truck routes causes an increase in the route generation times ("Route gen. time" column) and the number of iterations. As indicated in Table 5, although the average truck path generation time is 13% of the total solution time for the case with 10 customers, this ratio increases to nearly 40% for the case with 12 customers.

To indicate the efficiency of the drone-supported delivery concept, the savings obtained by deploying a drone and a truck is reported in Table 6 (given in percentage). The total delivery completion time ($v(Z)$) and the TSP solution provided in Table 4 are used to calculate the time savings. As explicitly detailed, more than 31% of the TSP solution is saved on average for the cases with 10 and 11 customers, and the average time saving is nearly 27% for the case that involve 12 customers.

We note that our two-stage algorithm is motivated by the consideration that the drone should visit as many customers as possible.

Table 6
Savings compared with the TSP solution.

Case	(TSP-v(Z)) * 100/TSP		
	$C = 10$	$C = 11$	$C = 12$
1	35.84	19.82	16.09
2	31.42	34.46	30.20
3	36.57	29.60	24.65
4	27.87	31.83	37.81
5	33.18	33.77	22.12
6	31.65	34.75	33.32
7	33.16	34.23	26.40
8	37.25	72.89	26.79
9	22.42	31.15	25.26
10	27.93	28.33	28.71
Avg	31.73	35.08	27.13

Table 7
Solution times and number of drone nodes obtained for uniform data when drone speed is 40 km/h.

D_Speed = 40 km/h Instance	C = 10		C = 11		C = 12	
Uniform	Runtime	# of d.	Runtime	# of d.	Runtime	# of d.
1	890	3	847	3	928	5
2	167	3	873	3	1168	3
3	825	4	838	3	610	4
4	154	3	997	3	3600*	**
5	270	3	685	3	643	4
6	173	3	769	2	3600*	**
7	153	2	1191	2	3600*	**
8	134	3	3600*	**	2024	3
9	87	5	565	3	1420	3
10	216	3	177	4	3037	3
Avg.	306.9	3.2	771.33	2.9	1404.3	3.6

* CPLEX is terminated due to runtime limit. Reported results demonstrate the runtime limit.
** Results cannot be obtained due to runtime limit.

However, the number of customer nodes that are assigned to the drone, which is provided in the "# of D." column in Table 4, is smaller than D_{max} in most cases because the truck visits as many customer nodes as possible between the launch and the pickup nodes of any drone tour instead of immediately arriving at the pickup node and waiting for the drone. Thus, fewer customer nodes remain to be served by the drone. With the aim of investigating the impact of drone speed on the number of drone nodes, we also run the uniform instances with an assumption of 40 km/h drone speed. The results indicate that the number of assigned drone nodes is small when the drone speed is 40 km/h. The results of this scenario are provided in Table 7. The average drone nodes are 3.8, 4.2 and 4.4 for the instances that involve 10, 11 and 12 customer nodes, respectively, in the scenario with a 56 km/h drone speed, whereas the average drone nodes are 3.2, 2.9 and 3.6 in the scenario with a 40 km/h drone speed. A consequence of fewer drone assignments is distinct in the waiting times of each vehicle in Table 4. In most cases, the length of time waited by the drone is longer than the length of the time waited by the truck; the waiting time of the truck has a significant effect on total delivery completion time.

Although our algorithm provides significant improvements in the computational time of the TSP-D with uniform data, fewer

Table 8
Solution times obtained for centered and clustered data.

Instance	Centered C = 10	Data		C = 11			C = 12		
	Murray et al.	Agatz et al.	Es Yurek et al.	Murray et al.	Agatz et al.	Es Yurek et al.	Murray et al.	Agatz et al.	Es Yurek et al.
1	3600*	658	576	3600*	3600*	2885	3600*	3600*	3600*
2	3600*	947	77	3600*	3600*	2254	3600*	3600*	3600*
3	3600*	704	84	3600*	3600*	698	3600*	3600*	3600*
4	3600*	624	310	3600*	3600*	2693	3600*	3600*	3600*
5	3600*	594	206	3600*	3600*	257	3600*	3600*	3600*
6	3600*	535	104	3600*	3600*	3600*	3600*	3600*	3600*
7	3600*	719	595	3600*	3600*	264	3600*	3600*	2591
8	3600*	718	639	3600*	3600*	1122	3600*	3600*	3600*
9	3600*	842	245	3600*	3600*	2915	3600*	3600*	3600*
10	3600*	589	124	3600*	3600*	876	3600*	3600*	3600*
Avg.		693	466			1819.6***			2591***
1	3600*	724	51	3600*	3600*	1026	3600*	3600*	3600*
2	3600*	577	552	3600*	3600*	734	3600*	3600*	3600*
3	3600*	403	112	3600*	3600*	196	3600*	3600*	3600*
4	3600*	478	186	3600*	3600*	3600*	3600*	3600*	694
5	3600*	458	188	3600*	3600*	3600*	3600*	3600*	3600*
6	3600*	473	7	3600*	3600*	117	3600*	3600*	3600*
7	3600*	301	109	3600*	3600*	1265	3600*	3600*	3600*
8	3600*	386	441	3600*	3600*	1148	3600*	3600*	3600*
9	3600*	282	138	3600*	3600*	3600*	3600*	3600*	3600*
10	3600*	423	70	3600*	3600*	3600*	3600*	3600*	3600*
Avg.		450.5	185.4			747.6***			694***

* CPLEX is terminated due to runtime limit. Reported results demonstrate the runtime limit.
*** Average runtime does not include runtime limit.

Table 9
Results obtained for uniform data under asymmetric distance assumption.

D_Speed = 56 km/h	C = 10		C = 11		C = 12	
	Total time	# of truck routes	Total time	# of truck routes	Total time	# of truck routes
1	618	54702	1024	56526	1287	91809
2	78	10148	2882	336804	2713	325316
3	401	49384	606	116307	1077	177297
4	133	19565	1213	188740	3600*	**
5	112	14252	612	158257	2102	137459
6	24	28040	1495	203301	973	132968
7	99	52488	714	177968	7303	521542
8	136	79504	458	79702	1284	171520
9	40	10316	546	120951	577	120461
10	100	9348	153	55365	2064	122192
Avg.	174.1	32774.7	970.3	149392.1	2153.3***	200062.7

* CPLEX is terminated due to runtime limit. Reported results demonstrate the runtime limit.
** Results cannot be obtained due to runtime limit.
*** Average runtime does not include runtime limit.

improvements in the computational times of centered and clustered data are obtained. Table 8 presents the solution times that are obtained for these two types of instances for 56 km/h drone speed. Murray and Chu (2015) cannot obtain the optimum solutions within the runtime limit for any of the instances of both types, whereas Agatz et al. (2016) obtain the optimum solutions for only the instances that involve 10 customers. The average solution time of Agatz et al. (2016) is approximately one and a half times larger than our solution time (693/466) for centered data, and the ratio for clustered data is more than two (450.5/185.4). However, our algorithm is terminated due to the runtime limit for one of the centered instances with 11 customer nodes, and the optimum solution for only one centered instance with 12 customer nodes can be obtained. The optimum solutions of the six clustered instances that involve 11 customer nodes and optimum for only one clustered instance with 12 customers can be obtained within the runtime limit. Another parameter that causes significant changes in the computation time of the algorithm is the drone speed. The algorithm runs for a longer period of time in the scenario with a lower drone speed. The optimum solutions for one of the instances with 11 customer nodes and three instances with 12 customer nodes cannot be obtained within runtime limit. The results obtained in the scenario with a 40 km/h drone speed are provided in Table 7. The average solution times increase to 306.9, 771.33 and 1404.3 s from 61.3, 299.1, and 760.2 s for the cases with 10, 11 and 12 customers, respectively, when the drone speed reduces from 56 km/h to 40 km/h.

We note that these solutions are obtained with the assumption that the traveling distances are symmetric. This assumption has a significant advantage in terms of the generated number of truck routes and solution times. When we run the uniform instances with the asymmetric distance assumption, we are capable of obtaining the optimum solutions of all instances in the runtime limit, with the exception of one. Although the number of truck routes is doubled, the rate of increase for the solution times seems to be larger than the rate of increase obtained with the symmetric distance assumption when compared with the results obtained for symmetric distance. The results are listed in Table 9.

5.3. Performance evaluation of heuristic algorithm

This section presents the results and the analysis of the results obtained for the proposed optimization-based heuristic algorithm. To assess the heuristic performance, we run the proposed heuristic algorithm for the cases with 10, 11 and 12 customer nodes with the assumption that the drone speed is 56 km/h and compare the results with the results obtained for the exact algorithm. We obtained the results for the generated instances that involve 20 customers in an average time that is shorter than six minutes. Since the exact algorithm is not capable of solving instances with 20 customer nodes, we compare these results with the TSP solutions to indicate the time percentage saved by delivering goods in coordination with a drone.

Outstanding results are observed for the solution times in Table 10. For the instances with 10, 11 and 12 customers, the optimization-based heuristic algorithm obtained solutions in less than one minute on average. A significant decrease in the number of generated truck paths ("# of truck routes" column) is evident in Table 10. As a consequence of this decrease, the time spent for truck path generation (Route gen. time column) significantly reduces. Despite the advantages of the proposed heuristic, the delivery completion time in minutes (the third column) increases compared with the delivery completion time obtained by the exact solutions. The increase in delivery completion times (given as a percentage) is represented in the "Increase in $v(Z)$%" column. The minimum increase (7.658%) is observed for the average of the instances with 10 customers, whereas the maximum average increase (13.086%) is obtained for the case with 11 customers.

Table 11 lists the test results obtained for the instances with 20 customers. The average time saving in delivery duration is 18.90%. The TSP-D solution is 13.83 min less than the time obtained for the TSP solution.

Table 10
Computational results of the heuristic algorithm for the uniform instances.

Case C = 10	Sol. time	Solution	Increase in v(Z) %	Traveling time of drone	Waiting time of truck	Waiting time of drone	# of truck routes	Route gen. time	# of drone routes
1	5	35.81	11.79	30.99	0.99	0.96	112	1	4
2	2	41.87	0.11	37.46	0.22	4.41	105	1	5
3	10	39.84	6.38	37.15	1.95	2.69	142	1	4
4	2	46.02	1.24	44.20	1.94	1.82	147	1	4
5	3	42.11	6.19	39.74	1.34	2.38	82	1	4
6	1	46.17	11.63	32.71	0.00	7.62	47	1	4
7	2	35.93	13.69	32.11	3.32	0.00	56	1	5
8	2	40.14	6.08	33.63	2.01	4.78	444	1	5
9	1	35.46	5.66	29.18	0.00	6.28	86	1	5
10	3	42.39	13.81	35.51	6.89	0.00	74	1	5
Avg 1–10	3.1	40.57	7.66	35.27	1.87	3.09	129.5	1	4.5
C = 11									
11	12	39.19	2.81	27.47	1.62	6.64	362	1	5
12	7	43.23	11.69	42.88	0.58	0.35	358	1	5
13	11	44.60	11.91	42.32	6.92	2.28	350	1	5
14	10	44.82	16.81	37.49	0.68	7.33	318	1	5
15	4	41.02	14.95	32.34	0.54	8.69	222	1	6
16	16	43.70	21.15	40.56	2.26	3.15	302	1	4
17	8	43.54	16.48	34.74	4.43	8.81	174	2	5
18	9	42.08	18.52	32.81	0.00	9.28	240	1	4
19	5	42.20	4.90	38.89	1.31	3.31	176	1	5
20	3	48.68	11.65	44.62	2.24	4.06	107	1	6
Avg 11–20	8.5	43.31	13.09	37.41	2.06	5.39	260.9	1.1	5
C = 12									
21	4	48.50	5.63	45.56	0.34	2.94	132	1	6
22	12	49.35	10.20	43.12	0.04	6.23	491	1	5
23	13	48.99	2.81	39.01	0.00	6.66	489	1	5
24	16	49.34	10.19	46.02	1.48	3.32	542	1	5
25	11	48.46	10.23	36.14	0.00	10.87	246	1	4
26	16	38.80	14.02	36.23	0.21	2.57	583	1	4
27	95	40.65	5.94	38.70	3.33	0.00	104	94	6
28	27	50.04	18.06	48.85	4.46	1.19	320	1	4
29	8	47.59	15.18	42.25	5.40	5.34	188	1	6
30	7	41.95	0.00	37.80	0.19	3.63	402	1	5
C = 20	20.9	46.37	9.23	41.37	1.55	4.28	349.7	10.3	5

Table 11
Computational results of the heuristic algorithm for the instances with 20 customers.

Case	Sol. time	Solution	TSP	(TSP-Sol.) * 100/TSP	# of truck routes	Route gen. time	# of drone routes
31	364	57.75	72.41	20.25	27823	9	10
32	153	56.92	65.27	12.80	4334	3	9
33	206	56.01	75.71	26.02	34866	10	9
34	87	55.13	75.51	26.99	27120	8	8
35	224	57.66	73.95	22.03	27527	8	9
36	104	66.15	73.80	10.36	3532	3	9
37	75	64.30	79.59	19.21	4565	3	8
38	1089	59.91	70.97	15.59	30141	9	8
39	779	52.24	67.06	22.10	52957	29	9
40	549	64.34	74.54	13.68	29706	9	10
Avg 31–40	363	59.04	72.88	18.90	24257.1	9.1	8.9

6. Discussions and conclusion

In this paper, a new variant of the TSP, which is referred to as the traveling salesman problem with drone (TSP-D), is investigated. Current studies have demonstrated that accomplishing a coordinated delivery by a truck and a drone is more efficient than the traditional delivery concept in terms of delivery time. Due to the increasing interest in this new delivery concept by commercial firms, an extensive study is necessary in this field of study.

We have proposed a decomposition-based iterative optimization algorithm to achieve this purpose. The proposed algorithm was run using randomly generated instances with selected parameters and compared with CPLEX solutions of previous studies, which were proposed by Agatz et al. (2016) and Murray and Chu (2015). The proposed algorithm verified its efficiency by solving the uniform instances with a problem size of 12 customers in a reasonable amount of time, whereas existing studies optimally solved problems with a maximum of 10 customers within the same amount of time. However, fewer improvements were obtained in the solution times of centered and clustered data. These findings are to be expected since the gap between the initial lower and upper bounds are large in centered and clustered data. When real-life applications are considered, it seems difficult to adapt drone delivery in urban areas due to the regulations, high buildings and dense population. The drone delivery is particularly considered for the adaptation to suburban areas due to the ability to access difficult terrains. One challenging area of future research is to develop an efficient solution approach for clustered data. It can be achieved by considering cluster first-solve second approach.

We have contributed to the literature by reducing the search area of the proposed algorithm for the heuristic approach and obtained solutions for medium-sized instances within reasonable runtimes. Although this study improves the exact solution approaches, the runtimes of the proposed algorithm are large, for instances, with problem sizes larger than 12 customers. An extension may include the development of efficient algorithms for truck path generation and drone routing. Hybrid algorithms based on metaheuristics may cause challenging results.

References

Agatz, N., Bouman, P. and Schmidt, M., 2016. Optimization Approaches for the Traveling Salesman Problem with Drone. ERIM Report Series Reference No. ERS-2015-011-LIS.

Amazon.com, Amazon Prime Air. [Online] Available at: https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011 (accessed 25 May 2017).

Banker, S., 2016. The most interesting part of mercedes' drone-equipped delivery van concept has nothing to do with drones. [Online] Available at: https://www.forbes.com/sites/stevebanker/2016/09/08/a-new-mercedes-benz-van-designed-to-facilitate-drone-deliveries/#64e7a1a641b1 (accessed 13 May 2017).

Bouman, P., Agatz, N., Schmidt, M., 2015. Instances for the tsp with drone.

Crowe, S., 2017. UPS Delivery Drone Launches from Truck. Available at: http://www.roboticstrends.com/article/ups_delivery_drone_launches_from_truck (accessed 13 May 2017).

Drexl, M., 2012. Synchronization in vehicle routing-a survey of VRPs with multiple synchronization constraints. Transport. Sci. 46 (3), 297–316.

Ferrandez, S.M., Harbison, T., Weber, T., Sturges, R., Rich, R., 2016. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. J. Indus. Eng. Manage., 9(2), 374–388.

French, S., 2015. 6 myths about Amazon Prime Air and drone delivery, debunked. Available at: http://www.marketwatch.com/story/6-myths-about-amazon-prime-air-and-drone-delivery-debunked-2015-12-02 (accessed 25 May 2017).

Ha, Q.M., Deville, Y., Pham, Q.D. and Ha, M.H., 2015. Heuristic Methods for The traveling Salesman Problem with Drone.

Ha, Q.M., Deville, Y., Pham, Q.D., Ha, M.H., 2018. On the min-cost traveling salesman problem with drone. Transport. Res. Part C: Emerg. Technol. 86, 597–621.

Murray, C.C., Chu, A.G., 2015. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transport. Res. Part C: Emerg. Technol 54, 86–109.

Paul, F., 2015. 10 reasons Amazon's drone delivery plan still won't fly. Available at: http://www.networkworld.com/article/2900317/wireless/10-reasons-amazons-drone-delivery-plan-still-wont-fly.html (accessed 25 May 2017).

Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: extended models and connections. Networks 70 (1), 34–43.

Ponza, A., 2016. Optimization of Drone-Assisted Parcel Delivery. University of Padova.

Snider, M., Welch, W.M., 2015. FAA shoots down Amazon's drone delivery plans. Available at: https://www.usatoday.com/story/tech/2015/02/15/amazon-cool-to-drone-rules/23473791/ (accessed 25 May 2017).

Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: several worst-case results. Optim. Lett. 11 (4), 679–697.

Wohlsen, M., 2014. The next big thing you missed: Amazon's delivery drones could work-they just need trucks. Available at: https://www.wired.com/2014/06/the-next-big-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-of-shipping/ (accessed 1 June 2017).