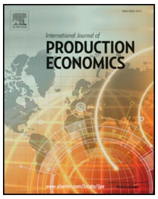
International Journal of Production Economics 214 (2019) 220–233

Contents lists available at ScienceDirect

International Journal of Production Economics

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



Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones

Ho Young Jeong[a], Byung Duk Song[b,*], Seokcheon Lee[a]

[a] School of Industrial Engineering, Purdue University, 315 N. Grant St., West Lafayette, IN 47907, USA
[b] Department of Industrial and Management Systems Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do, Republic of Korea

ARTICLE INFO

Keywords:
Routing
Flying sidekick
Drone
Traveling salesman
Mixed integer programming
Heuristic algorithm

ABSTRACT

The truck-drone hybrid delivery system uses a truck as a station for drones in addition to its delivery function and is getting attention because the strengths of these individual vehicles can be selectively and synergistically exploited. In this study, we extended the previous vehicle routing models to the hybrid delivery systems by taking into account two important practical issues: the effect of parcel weight on drone energy consumption and restricted flying areas. The flight range of the drones is heavily susceptible to the loaded weight due to the limited battery life. Drones are also not allowed to fly over sensitive facilities regulated by the Federal Aviation Administration (FAA) or temporarily in certain areas due to weather-related conditions. We developed a mathematical model that incorporates these issues and propose a two-phase constructive and search heuristic algorithm to provide computational efficiency of the real-world cases problems. The result of the numerical study demonstrates the effectiveness and efficiency of the proposed algorithm.

1. Introduction

As a new delivery method for crossing the last-mile, drones have been receiving much attention. For instance, a partnership between UPS, Gavi, and the Vaccine Alliance is exploring the use of drone technology to deliver medical supplies to remote areas (UPS, 2016). In December 2013, Amazon announced Prime Air – a delivery service using drones (Banker, 2013) and in December 2016, they launched their first trial of the Prime Air in the UK (Perlow, 2016). DHL also made a successful test for medical supplies delivery by drones called parcelcopter (Bryan, 2014). A Silicon Valley start-up Matternet is another company developing small drones for the delivery of lightweight parcels (Dillow, 2015).

Although drones can provide distinct advantages such as mobility, flexibility, and low cost, they have a fundamental drawback with their limitations in flight duration and flight restriction, making it difficult to use them in a large area operation. One way to overcome this barrier is the use of a hybrid delivery system consisting of a large ground vehicle and a small aerial vehicle. The hybrid delivery system has significant advantages compared to traditional delivery since the strengths of individual vehicles can be selectively and synergistically exploited. The first truck-drone delivery system was reported in 2014 and developed by University of Cincinnati researchers and AMP Electric Vehicles (HorseFly, 2014). Subsequently, the Workhorse Group developed and tested the HorseFly drone's ability to make deliveries from a Workhorse truck in 2016 (Workhorse, 2016).

Mathematical models for hybrid truck-drone delivery systems have been developed as well. The first model was suggested by Murray and Chu (2015), named the "Flying Sidekick Traveling Salesman Problem (FSTSP)". In the FSTSP, a drone moves along with the truck and launches from the truck to provide a delivery while the truck continues to serve its customers in different locations. Once the drone completes its service to one customer, it needs to fly back to the truck. The objective of FSTSP is to develop a truck and drone routes such that the completion time of all deliveries is minimized. The unique approach of using the truck as a drone station as well as a delivery vehicle has several advantages. First, by transporting the drone closer to a customer, the drone can save the energy, and more customers become accessible by drone. Second, the truck can be used for deliveries or supportive delivery for the situations where the use of the drone is limited due to a payload limit or flight restrictions. Consequently, these two delivery methods complement each other to decrease overall delivery time compared to regular truck delivery, as shown in Fig. 1.

There are practical limitations of drones that have not taken into consideration in research, leading infeasible routes of the hybrid delivery system. First, the flight range of the drone may directly be limited by the weight of payload because of the typically finite battery capacity and its battery-powered nature. Second, the drone may not be used in

[*] Corresponding author.
E-mail address: bdsong@kaist.ac.kr (B.D. Song).

https://doi.org/10.1016/j.ijpe.2019.01.010
Received 11 May 2018; Received in revised form 24 September 2018; Accepted 7 January 2019
Available online 08 January 2019

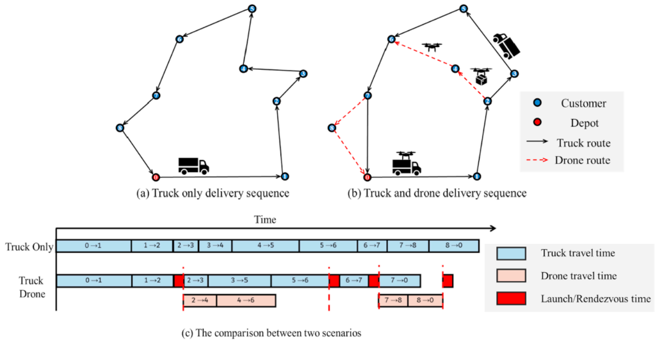


Fig. 1. Effect of truck-drone hybrid delivery. (a) The truck only delivery route, (b) truck-drone delivery route, and (c) comparison of route performance.

'no-fly zones' as announced by the Federal Aviation Administration (FAA, 2017). The 'no-fly zones' includes not only sensitive facilities such as airports, heliports, and national parks but also places like stadiums which require temporary restrictions during certain time periods. Besides, severe weather conditions will make certain areas into no-fly zones during certain periods due to the safety of the drone, civil properties, and citizen.

The primary contribution of this study is the development of a robust FSTSP that implements energy consumption and no-fly zone (therefore called FSTSP-ECNZ). This approach provides academic and practical contributions by developing a new mixed integer linear programming (MILP) formulation, solution algorithm and providing the practical usability of the hybrid delivery system. The performance of the proposed solution algorithm was tested and compared with the well-known metaheuristics, genetic algorithm (GA), particle swarm optimization (PSO), simulated annealing (SA) and the nearest neighbor (NN).

2. Literature review

The problem that we are dealing with can be considered an extension of the Carrier-Vehicle TSP (CV-TSP) that has been extensively studied by Garone et al. (2008, 2010, 2011,2014) in which marine carriers and aircraft work as a team to conduct a rescue mission by visiting a set of locations. In such a system, the marine carrier does not visit the rescue area but instead carries the aircraft near the area, and the aircraft works as a rescue operator by itself. The truck and trailer routing problem (TTRP) also considered the simultaneous use of two delivery means and in the TTRP, customers are defined with two groups that may be visited by a truck or a trailer (Chao, 2002; Scheuerer, 2006; Lin et al., 2009; Derigs et al., 2013; Drexl, 2011, 2014). The main difference in TTRP is that the trailer cannot serve customers without the truck, while our model enables both vehicles to work separately.

The collaborative truck and drone delivery team has received much attention recently and there are several studies that addressed this collaborative system (Murray and Chu, 2015; Ferrandez et al., 2016; Ponza, 2016; Agatz et al., 2018; Ha et al., 2018; Mathew et al., 2015). The concept of this system was first described by Murray and Chu (2015), and the authors named the system as FSTSP optimizing the delivery schedule of a single truck-single drone scenario. Ferrandez et al. (2016) extended the FSTSP to include multiple drones and investigated the effectiveness of a truck-drone delivery. They suggested a design of algorithm utilizing K-means clustering to find launch locations, as well as the GA to find the truck route. Other literature (Ponza, 2016), proposed a heuristic method based on Simulated Annealing to provide solutions for the FSTSP with different types of drones. Another heuristic based on Randomized Variable Neighborhood Descent (RVND) is proposed and evaluated (de Freitas and Penna, 2018).

Agatz et al. (2018) suggested Traveling Salesmen Problem with Drone (TSP-D) that shares the same structure as the FSTSP but assumes drone travel on the same road networks as the truck. The authors provided a lower bound on the optimal solution over the truck-only system while relinquishing the advantage of taking shortcuts via Euclidean distances. The route first-cluster second heuristic was proposed, and the heuristic utilizes the strength of the lower bound of truck-only solution that was provided before clustering. In Ha et al. (2018), two heuristic algorithms were proposed to solve TSP-D: Greedy Randomized Adaptive Search Procedure (GRASP) and a heuristic adapted from the work of Murray and Chu (2015) called TSP-LS.

Another related problem, the Heterogeneous Delivery Problem (HDP) was designed on a physical street network and allowed the truck to launch a drone at every endpoint of an arc (Mathew et al., 2015). The difference between this study and the HDP is that a truck cannot achieve a direct delivery to the customer and it only launches a drone to service the customers. For the solution approach, the authors transform the problem to a Generalized Traveling Salesman Problem (GTSP), and use the Nood-Bean Transformation available in Matlab to reduce a GTSP to a TSP which is then solved by a heuristic. Table 1 presents an overview of the above studies and briefly summarizes in terms of problem natures and solution approaches.

In terms of analysis of this hybrid system, Carlsson and Song (2017) used theoretical analysis to demonstrate In terms of analysis of this hybrid system, Carlsson and Song (2017) used theoretical analysis to demonstrate that the potential benefit of using the hybrid system is proportional to the square root of the velocity ratio between the vehicle and the drone. Wang et al. (2017) introduced the Vehicle Routing Problem with Drone (VRP-D) with multiple trucks and drones and conduct worst-case scenarios analysis to identify the benefits of drone usage. The development of this study is described by Poikonen et al. (2017) not only explicitly considers limits on battery life and cost objectives, but also extends worst-case boundaries to distance/cost metrics.

The existing researches for collaborative delivery have been studied without considering the effect of parcel weight on drone flight duration and flight regulations. However, these constraints are common and must be considered together because of their interdependency rooted from the flight time and route configuration for the practical use. For example, in the general VRP problem of ground vehicles, Xiao et al.

Table 1
Overview of the cooperative team delivery problem.

Problem type	Literature	Number of		Parcel weight	No-fly zone	Solution approach
		trucks	drones			
FSTSP- ECNZ	This study	Single	Single	o	o	MILP formulation Heuristic
FSTSP	Murray and Chu (2015)	Single	Single	x	x	MILP formulation Heuristic
	Ferrandez et al. (2016)	Single	Multiple	x	x	GA and K-means
	Ponza (2016)	Single	Single	x	x	SA
	de Freitas & Penna (2018)	Single	Single	x	x	Heuristic
TSP-D	Agatz et al. (2018)	Single	Single	x	x	MILP formulation Heuristics
	Ha et al. (2018)	Single	Single	x	x	MILP formulation Heuristics
HDP	Mathew et al. (2015)	Single	Single	x	x	Reduction to GTSP Reduction to TSP Heuristic

(2012) and Zhang et al. (2015) provided a fuel consumption model according to loading weight. Obviously, drones are much more vulnerable to parcel weight. So it is essential to address this issue in the problem of aerial vehicles that are susceptible to weight. Constraints regarding flight time, speed, coverage distance, and payload weight have been taken into consideration for drones' transportation (San et al., 2016; Chowdhury et al., 2017). However, the actual relationship between drone weight and energy consumption was modeled in (Di Franco and Buttazzo, 2015) to estimate its realistic coverage. By formulating a drone energy consumption model that differs by payload weight, the vehicle routing problem has been solved with multiple trips (Dorling et al., 2017).

The restricted flight area is another issue that has not been considered although it strongly limits a drone's service coverage. However, there are similar studies that have proposed avoiding obstacles by using path planning and suggest heuristic approaches to finding solutions in dynamic obstacle environments (Lozano-Pérez & Wesley, 1979; Canny and Reif, 1987; Zhuoning et al., 2010; Raja and Pugazhenthi, 2012).

In this study, a new problem called FSTSP-ECNZ is suggested to enable the realistic implementation of the hybrid delivery team for real world last mile delivery service. To achieve the purpose, we developed; 1) an energy consumption model that considers the relationship between energy consumption and loaded parcel weight, and 2) a detour method to avoid time-dependent restricted flight areas. These characteristics are presented in a linear form and included in the proposed mathematical model. Also, a new heuristic called the Two-Phase Construction and Search Algorithm (TPCSA) is developed to derive vehicle schedules of FSTSP-ECNZ. The performance of the proposed heuristic is evaluated via comparison with other well-known heuristic algorithms.

3. Problem definition and formulation

In this section, we propose the mathematical formulation of FSTSP-ECNZ to accurately handle the drone's flight limits by developing the energy consumption model of drone flight and suggesting detour method for time-dependent no-fly zones. The following general assumptions are adapted to determine the nature of the problem for the proposed study.

- The drone is launched and retrieved from the truck at only a depot or a customer location.
- The battery of the UAV is replaced without being charged, and the replacement time is included in the launch and rendezvous time.
- Drones fly at a steady speed, and the fluid density of air and gravity are assumed to be fixed.
- The shape of a no-fly zone is a circle, and the zones do not overlap.

3.1. Payload-energy dependency and No-Fly zones

3.1.1. Energy-payload consumption model

Due to UAV's limited batteries, its energy estimation is of paramount importance and more, an up-and-coming research area (Tokekar et al., 2016). We propose a model that uses loading weight to determine the energy expenditure of drones which can be used to estimate the possible flight duration. In the process of developing the model, the data of a drone called the “MK8-3500 standard” is adopted, and it is specified in MikroKopter. (2017).

In the equation (2) of Dorling et al. (2017), the power equation is developed with n-rotors drone as shown in equation (1). From this equation, the power P (watts) of a drone can be calculated with steady speed, with W (kg) frame weight and payload weight w (kg), the fluid density of air ρ (kg/m^3), the area of the spinning blade disc ς (m^2), and gravity g (N).

$$P = (W + w)^{\frac{3}{2}}\sqrt{\frac{g^3}{2\rho\varsigma n}} \tag{1}$$

The linear regression is applied to equation (1) to derive linear equation (2), which has two variances, β_0 and β_1, representing the required power for the drone flight and the payload weight w respectively. The average percent error of 0.0083% and the largest difference was 0.011 kW. Therefore, the power demand increases near linearly as the payload increases.

$$p(w) = \beta_0 + \beta_1 w \tag{2}$$

Since we are interested in the change of flight time due to the power variance, equation (3) was derived which provides the estimated flight time. In equation (3), η, C, V_n and P stand for the energy conversion efficiency, battery capacity (mAh), nominal battery voltage with n number of cells (V) and power consumption (W), respectively. The energy conversion efficiency η is assumed to be 0.7.

$$FlightTime = \frac{\eta \cdot C \cdot V_n}{P} \tag{3}$$

By substituting the equation (2) into P in the equation (3), the flight time is calculated, and it is compared with the result of a real experiment in MikroKopter. (2017). Four different versions of the “MK8-3500” drones with different specifications are applied to the comparison test. Fig. 2 demonstrates that the proposed energy consumption model provides realistic values that are analogous to the experiment result. We also have evaluated the model statically using a one-sided independent t-test, which revealed that they are relevant with a p-value of 5%. As a result, equation (2) is considered to provide a realistic energy consumption value based on the loaded weight and is used for the drone energy constraint (11) in section 3.2.

3.1.2. Time-dependent detour method

Areas called 'no-fly zone' prohibit drone operation at a specific time or indefinitely. When these zones are encountered, a drone detour may be required for such areas, or sometimes not even be servable by drone. To address this issue, the time-dependent detour method is developed. In the process, the detour decision is determined first, and then detour distance is calculated as required.

Fig. 3 depicts how the detour decision will be arrived at considering six possible cases. Drone's arrival time in node i, j are indicated T_i^d, T_j^d, and the start, the end time of the no-fly zone h are denoted as st_h, et_h. If the drone's arrival time at node i(j) is earlier than the end (start) time of zone h, $z_{i,h}^e$ ($z_{j,h}^s$) is equal to 1, otherwise 0. The value of $z_{i,h}^e$ - $z_{j,h}^s$ determines the occurrence of the detour around the restricted area h when traveling from node i to node j (refer to equations 9(a)–(d) in section 3.2 for details). When a drone moves from node i to j, if the flying time of the drone overlaps with the prescribed time of no-fly zone h, the detour is necessary like case 2 to 5 in Fig. 3. If not, the drone travels as usual as

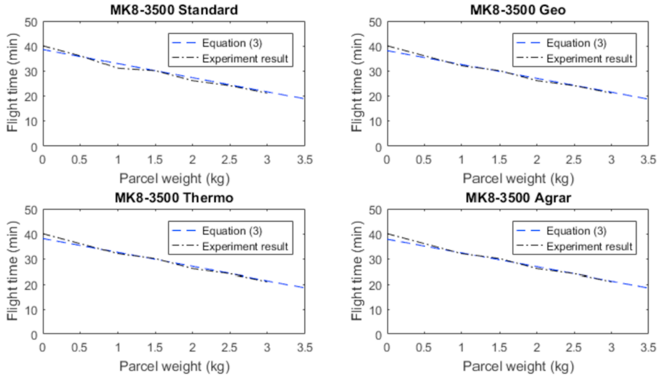
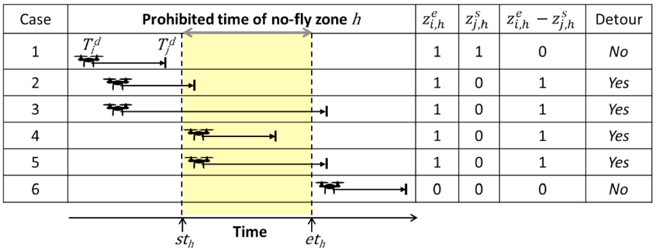
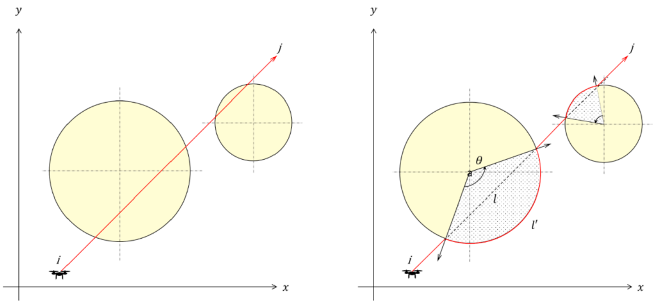


Fig. 2. The graphical comparison of equation (3) and experiment result in MikroKopter. (2017).



Case	Prohibited time of no-fly zone h	$z^e_{i,h}$	$z^s_{j,h}$	$z^e_{i,h} - z^s_{j,h}$	Detour
1		1	1	0	*No*
2		1	0	1	*Yes*
3		1	0	1	*Yes*
4		1	0	1	*Yes*
5		1	0	1	*Yes*
6		0	0	0	*No*

Fig. 3. The operating principle for making a detour decision according to prohibited time.



(a) Without detour

(b) With detour

Fig. 4. Path of a drone without or with a detour.

case 1 and 6.

In the case of a detour, as in cases 2–5, the detour route should be determined. Since path planning in the presence of multiple obstacles was found to be NP-Hard (Canny and Reif, 1987), the complexity of the problem will often prohibit the use of pure exact algorithms when combined with TSP which is also NP-Hard. Therefore, we assume that drones avoid obstacles and restricted flight areas by rotating the region which assures a good solution in many cases (Lozano-Pérez & Wesley, 1979). Fig. 4 graphically shows the rotating detour maneuver.

$$d_{i,j,h} = l' - l = (\frac{2\pi r_n}{360}\min(\theta_{i,j,h},\, 360 - \theta_{i,j,h}) - l) \tag{4}$$

Without detour, drones directly fly over the time-dependent no-fly zone, as shown in Fig. 4(a). However, should a detour be required, drones should rotate the area. In this case, the detour distance is using equation (4). In equation (4), $d_{i,j,h}$ indicates the additional distance generated by rotating no-fly zone h with a radius of r_h to travel from node i to j. The Euclidean distance between two nodes is l, and the length of the rotation route is l' with the central angle θ. The difference between these two distances indicates the extra flight to rotate the zone. As a result, when a detour is determined, equation (4) is applied to derive the additional distance through the rotation. In the case of a drone launching from a no-fly zone, which is infeasible, we put a large number in $d_{i,j,h}$ to prevent the infeasible case. Please refer equations 9(a)-(d) in section 3.2 to observe equation (4) in greater detail.

3.2. Mathematical formulation

With the concepts and models developed above, the FSTSP-ECNZ is formulated to derive delivery routes of truck and drone with the objective of minimizing the arrival time at ending depot after serving all customers. Suppose N is the set of n customer nodes and depot nodes. Also, set N is divided into subsets of the departing node N_d and arriving node N_a, which ensure that no vehicle leaves the ending depot and enter the starting depot. The mathematical model assumes one truck, and one drone scenario. The following notations are defined to formulate the FSTSP-ECNZ.

Indices	
i, j, k, l, m, n	Represent customers and depot, totally $c + 2$nodes
t, d	Represent truck (t) and drone (d) respectively
h	Represent no-fly zones
Sets	
N	Set of nodes $\mathrm{N} = \{0, 1, \ldots, c+1\}$
N_d	Set of departing nodes $\mathrm{N_d} = \{0, 1, \ldots, c\}$, $\mathrm{N_d} \subset \mathrm{N}$
N_a	Set of arriving nodes $\mathrm{N_a} = \{1, 2, \ldots, c+1\}$, $\mathrm{N_a} \subset \mathrm{N}$
C	Set of customers $\mathrm{C} = \{1, 2, \ldots, c\}$
C_d	Set of drone delivery eligible customers $\mathrm{C_d} \subseteq \mathrm{C}$
F	Set of tuples of the form $< \mathrm{i, j, k} >\ \forall\, \mathrm{i} \in \mathrm{N_d}, \mathrm{j} \in \{\mathrm{C_d}: \mathrm{j} \neq \mathrm{i}\}, \mathrm{k} \in \{\mathrm{N_a}: \mathrm{k} \neq \mathrm{j}, \mathrm{k} \neq \mathrm{i}\}$
NZ	Set of no-fly zones
Parameters	
$\tau^t_{i,\ j}$	Truck travel time from node i to j $\forall$ $i, j \in N$
$\tau^d_{i,\ j}$	Drone travel time from node$i$ to j $\forall$ $i, j \in N$
$\mathrm{d_{i,\ j,\ h}}$	Additional detour time for no-fly zone h when drone travels from node i to j $\forall$ $i \in N_d$, $j \in C_d$, $h \in NZ$
st_h, et_h	The starting and ending time of a no-fly zone h, $\forall$ $h \in NZ$
L	Launching time
R	Rendezvous time
w_i	Weight of parcel for customer i $\forall$ $i \in C$
E	Energy capacity of the drone
η	An energy efficiency factor of the drone's battery
M	Large positive constant
Decision Variables	
$x_{i,\ j}$	Binary decision variable, 1 if truck traverses from node i to node j; otherwise 0 $\forall$ $i \in N_d$, $j \in N_a$, $i \neq j$
$p_{i,\ j}$	Auxiliary decision variable, 1 if i is visited before j, otherwise 0 $\forall i \in N$, $j \in C$, $i \neq j$
u_i	Auxiliary decision variable, equal to the position of node i in truck's path $1 \le u_i \le c + 2$ $\forall_i \in N_d$
$y_{\ i,\ j,\ k}$	Binary decision variable, 1 if drone traverses from node i to node j to k; otherwise 0 $\forall$ $< i, j, k > \in F$
T^t_i	Real number decision variable, arriving time of truck at node i including launch/rendezvous time, $T_0{}^t = 0$ $\forall$ $i \in N$,
T^d_i	Real number decision variable, arriving time of drone at node i include launch/rendezvous time $T_0{}^t = 0$ $\forall$ $i \in N$
$z^s_{i,\ h}$	Binary decision variable, 1 if the drone's arriving time at node i is earlier than starting time of no-fly zone h, otherwise 0 $\forall$ $i \in N$, $h \in NZ$
$z^e_{i,\ h}$	Binary decision variable, 1 if the drone's arriving time at node i is earlier than ending time of no-fly zone h, otherwise 0 $\forall$ $i \in N$, $h \in NZ$

With the above notations, the FSTSP-ECNZ is developed and described via 5(a)-13.

$$\sum_{i \in N_a} x_{0,i} = 1 \tag{5(a)}$$

$$\sum_{i \in N_d} x_{i,c+1} = 1 \tag{5(b)}$$

$$\sum_{i \in N_d} x_{i,j} - \sum_{j \in N_a} x_{j,k} = 0 \quad \forall\, j \in C \tag{5(c)}$$

$$u_j - u_i \ge 1 - (c + 2)(1 - x_{i,j}) \quad \forall\, i \in C, j \in N_d, i \neq j \tag{5(d)}$$

Constraints 5(a)-(d) limit the truck routes. Constraint 5(a) states that the truck departs from the starting depot, while constraint 5(b) indicates that it arrives at the ending depot. Constraint 5(c) refers to the fact that if the truck visits node j, then it must depart from j. Constraint 5(d) works as a subtour elimination for a truck's route.

$$\sum_{j \in C_d} \sum_{k \in N_a} y_{i,j,k} \le 1 \quad \forall\, i \in N_d \tag{6(a)}$$

$$\sum_{i \in N_d} \sum_{j \in C_d} y_{i,j,k} \le 1 \quad \forall\, k \in N_a \tag{6(b)}$$

$$u_k - u_i \ge 1 - (c + 2)(1 - \sum_{j \in C} y_{i,j,k}) \quad \forall\, i \in C, k \in \{N_a: i \neq k\} \tag{6(c)}$$

Constraints 6(a)-(c) restrict the drone routes. Constraints 6(a)-(b) indicate that the drone may launch and rendezvous at any departing and arriving node respectively, and may only do so once. Constraint 6(c) is developed for the subtour elimination of a drone.

$$\sum_{i\in N_a} x_{i,j} + \sum_{i\in N_d}\sum_{k\in N_a} y_{i,j,k} = 1 \quad \forall\, j\in C \tag{7(a)}$$

$$2y_{i,j,k} \le \sum_{l\in N_d} x_{l,i} + \sum_{m\in C} x_{m,k}$$

$$\forall\, i\in C,\, j\in\{C_d\colon j\neq i\},\, k\in\{N_a\colon <i,j,k>\in F\} \tag{7(b)}$$

$$y_{0,j,k} \le \sum_{i\in N_d} x_{i,j} \quad \forall\, j\in C,\, k\in\{N_a\colon <i,j,k>\in F\} \tag{7(c)}$$

Constraints 7(a)-(c) are developed to connect both vehicles routes. In constraint 7(a), each customer should be visited exactly once by either drone or truck. Constraint 7(b) states that if the drone launches from node *i* and lands in node *k*, then the truck must visit both nodes *i* and *k*. Constraint 7(c) is a special case of constraint 7(b) when the drone is launched from the starting depot.

$$T_k^t \ge T_n^t + \tau_{n,k}^t + L\Big(\sum_{l\in C_d}\sum_{m\in N_a} y_{n,l,m}\Big) + R\Big(\sum_{i\in N_d}\sum_{j\in C_d} y_{i,j,k}\Big) - M(1 - x_{n,k}) \quad \forall\, n\in N_d,\, k\in N_a,\, n\neq k \tag{8(a)}$$

$$T_k^t \le T_n^t + \tau_{n,k}^t + L\Big(\sum_{l\in C_d}\sum_{m\in N_a} y_{n,l,m}\Big) + M\Big(\sum_{i\in N_d}\sum_{j\in C_d} y_{i,j,k}\Big) + M(1 - x_{n,k}) \quad \forall\, n\in N_d,\, k\in N_a,\, n\neq k \tag{8(b)}$$

The truck arrival time is calculated through constraints 8(a)-(b). Constraint 8(a) accounts for the arrival time of the truck in node *j* which is computed by taking the sum of the arrival time in node *i*, the travel time from *i* to *j* and the launch and rendezvous time for the drone if applicable. Constraint 8(b) fixes the arrival time of the truck if the drone does not rendezvous at that point.

$$T_j^d \ge T_i^d + \tau_{i,j}^d + \sum_{h\in NZ} d_{i,j,h}(z_{i,h}^e - z_{j,h}^s) + L - M\Big(1 - \sum_{k\in N_a} y_{i,j,k}\Big) \quad \forall\, i\in N_d,\, j\in C_d,\, i\neq j \tag{9(a)}$$

$$T_j^d \le T_i^d + \tau_{i,j}^d + \sum_{h\in NZ} d_{i,j,h}(z_{i,h}^e - z_{j,h}^s) + L + M\Big(1 - \sum_{k\in N_a} y_{i,j,k}\Big) \quad \forall\, i\in N_d,\, j\in C_d,\, i\neq j \tag{9(b)}$$

$$T_k^d \ge T_j^d + \tau_{j,k}^d + \sum_{h\in NZ} d_{j,k,h}(z_{j,h}^e - z_{k,h}^s) + R - M\Big(1 - \sum_{i\in N_d} y_{i,j,k}\Big) \quad \forall\, j\in C_d,\, k\in N_a,\, j\neq k \tag{9(c)}$$

$$T_k^d \le T_j^d + \tau_{j,k}^d + \sum_{h\in NZ} d_{j,k,h}(z_{j,h}^e - z_{k,h}^s) + R + M\Big(1 - \sum_{i\in N_d} y_{i,j,k}\Big) \quad \forall\, j\in C_d,\, k\in N_a,\, j\neq k \tag{9(d)}$$

$$T_i^d \ge st_h - Mz_{i,h}^s \quad \forall\, i\in N,\, h\in NZ \tag{9(e)}$$

$$T_i^d \le st_h + M(1 - z_{i,h}^s) \quad \forall\, i\in N,\, h\in NZ \tag{9(f)}$$

$$T_i^d \ge et_h - Mz_{i,h}^e \quad \forall\, i\in N,\, h\in NZ \tag{9(g)}$$

$$T_i^d \le et_h + M(1 - z_{i,h}^e) \quad \forall\, i\in N,\, h\in NZ \tag{9(h)}$$

The drone arrival time is calculated by considering a detour of no-fly zones through constraints 9(a)-(h). In constraints 9(a)-(b), the arrival time for the drone from launch to delivery is computed by summing the travel time, launching time, and detour time if required. Under the same logic, the arrival time from delivery to rendezvous is computed in 9(c)-(d). For the detour decision, the constraints from 9(e) to 9(h) are used to compare the drone's arrival time T_i^d with the starting st_h (or ending et_h) time of the zone *h* and determine the value of the decision variables $z_{i,h}^s$ and $z_{i,h}^e$. Fig. 3 in section 3.1.2 demonstrates how these decision variables determine whether to take a detour or not. However, in 9(e)–(h), the drone's arrival time may conflict with the st_h or et_h. Therefore, constraints 9(b) and (d) were developed to fix the value of the arrival time.

$$T_i^d \ge T_i^t - M\big(1 - \textstyle\sum_{j\in C_d}\sum_{k\in N_a} y_{i,j,k}\big) \quad \forall\, i\in N_d \tag{10(a)}$$

$$T_i^d \le T_i^t + M\big(1 - \textstyle\sum_{j\in C_d}\sum_{k\in N_a} y_{i,j,k}\big) \quad \forall\, i\in N_d \tag{10(b)}$$

$$T_k^t \ge T_k^d - M\big(1 - \textstyle\sum_{i\in N_d}\sum_{j\in N_a} y_{i,j,k}\big) \quad \forall\, k\in N_a \tag{10(c)}$$

Constraints 10(a)-(c) link the arrival time of both vehicles when they meet. If a truck carries a drone, the arrival time for the drone should be updated before its launch according to the truck's arrival time 10(a), 10(b). Constraint 10(c) tries to synchronize the arrival time of the truck and drone. If a drone arrives later than the truck, the truck should wait and the arrival time of the truck needs to be fixed with that of the drone.

$$p(0)(T_k^d - T_j^d) + p(w_j)(T_j^d - T_i^d) \quad \forall <i,j,k>\in F$$
$$\le \eta E + M\big(1 - \textstyle\sum_{k\in N_a} y_{i,j,k}\big) \tag{11}$$

Constraint (11) ensures that a drone does not fly if the tour exceeds its battery capacity.

$$p_{0,i} = 1 \quad \forall\, i\in C \tag{12(a)}$$

$$p_{i,j} + p_{j,i} = 1 \quad \forall\, i\in C,\, j\in\{C\colon i\neq j\} \tag{12(b)}$$

$$u_j - u_i \le -1 + (c+2)p_{i,j} \quad \forall\, i\in C,\, j\in\{C\colon i\neq j\} \tag{12(c)}$$

$$u_j - u_i \ge 1 - (c+2)(1 - p_{i,j}) \quad \forall\, i\in C,\, j\in\{C\colon i\neq j\} \tag{12(d)}$$

$$T_l^t \ge T_k^t - M\left(3 - \sum_{j\in C_d} y_{i,j,k} - \sum_{\substack{m\in C_d\\ m\neq i\\ m\neq k}}\sum_{\substack{n\in N_a\\ n\neq i\\ n\neq k}} y_{l,m,n} - p_{i,l}\right)$$

$$\forall\, i\in N_d,\, k\in\{N_a\colon k\neq i\},\, l\in\{C\colon l\neq i,\, l\neq k\} \tag{12(e)}$$

Constraints 12(a)-(e) ensure the order of the routes. If the truck departs from node 0, constraint 12(a) forces $p_{0,i}$ for every $i\in C$ equal to 1. Constraints 12(b)-(d) ensures that the truck does not visit the same route from a different direction to make sure there are no redundant visits. Constraint 12(e) prevents the drone from launching while it is flying.

The objective function seeks to minimize the time required to service all customers and return both vehicles to the depot (13). Although T_{c+1}^t indicates a truck's arrival time at the depot, it matches our intended objective function because the truck returns carrying the drone.

$$\text{Minimize } T_{c+1}^t \tag{13}$$

4. TPCSA

The TSP was proven to be NP-Hard (Korte and Vygen, 2012), and the FSTSP is the NP-Hard as well (Murray and Chu, 2015). Thus, it is obvious that the proposed FSTSP-ECNZ also belongs to NP-hard class. Though small instances can be solved through the proposed mathematical model, solving large-scale instances requires a computationally inexpensive solution approach. Therefore, we propose an evolutionary-based heuristic algorithm consisting of the construction phase and search phase, the TPCSA. It has been proven that this combined approach not only further improves the solution quality but also saves computation time in scheduling problems (Liu and Reeves, 2001).

The TPCSA starts by generating initial truck routes, which undergo seeding for potential improvement. Based on these initialized truck routes, the main loop proceeds to phase I by assigning customers to the drone selectively and constructing routes for the drone. The total route that is completed in phase I is evaluated and the current best solution is

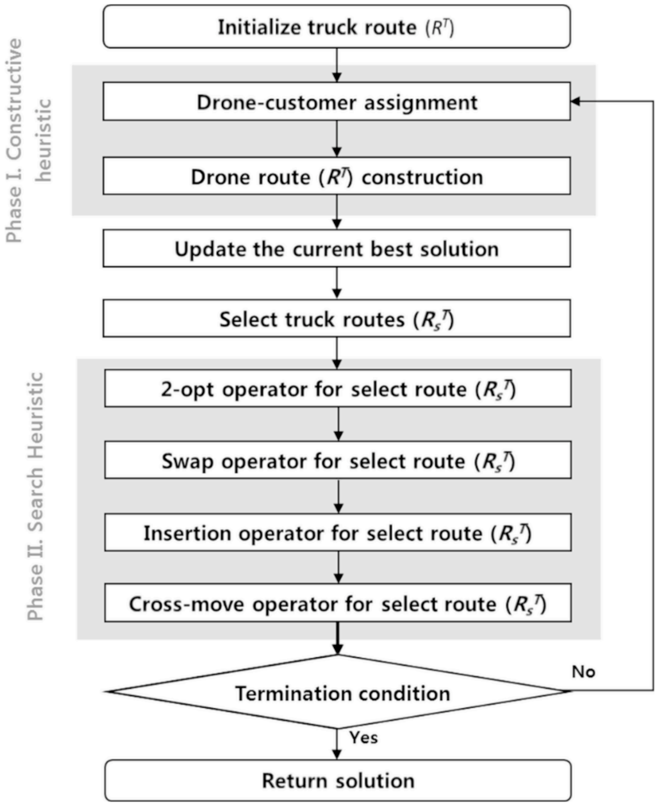

Initialize truck route (R^T)
Drone-customer assignment
Drone route (R^D) construction
Update the current best solution
Select truck routes (R_s^T)
2-opt operator for select route (R_s^T)
Swap operator for select route (R_s^T)
Insertion operator for select route (R_s^T)
Cross-move operator for select route (R_s^T)
Termination condition
No
Yes
Return solution
Phase I. Constructive heuristic
Phase II. Search Heuristic

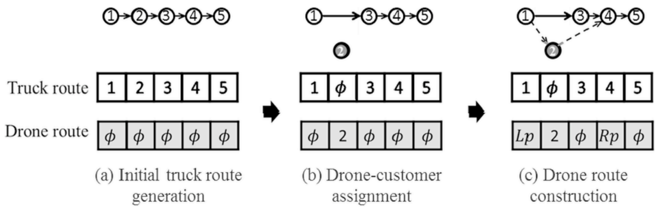
Fig. 5. Flowchart of TPCSA.

updated. After updating the solution, a portion of the truck routes is selected and undergo search operators in phase II. These search operators consist of 2-opt, Swap, Insertion to search for locally optimal solutions, and Cross-move to diversify the search. Until the termination condition is satisfied, the loop continues exploring solutions. Fig. 5 shows the flowchart of the algorithm.

This combinational approach takes advantage of both heuristics. It adopts the constructive heuristics for computational inexpensively and the search heuristics for exploration capability. This algorithm is also incorporated into problem-specific optimality conditions for an efficient search process.

4.1. Initial solution generation

The procedure begins by generating an initial solution of a full route that utilizes the truck only R^T. Fig. 6(a) shows an example of initialized routes. We use the centrality policy that was recently studied by Lee (2012, 2013; 2014). In general, if a chosen customer is located away from the rest of the customers, the next delivery times will tend to increase. In the centrality policy, therefore, the centrality of customer i (c_i) is computed as equation (14) to prioritizes customers in terms of the efficiency of their location. After computing the centrality of all customers, we calculate the fitness value of i-j with the centrality weighted by the parameter a as shown in equation (15). By differing the value of a from 0 to 2 for each population, various seeds can be obtained. By choosing customer j to maximize the fitness value, we build the path of the truck R^T. Algorithm 1 provides the pseudo code of the initial truck route generation.



(a) Initial truck route generation

(b) Drone-customer assignment

(c) Drone route construction

Fig. 6. An example of drone route construction.

$$c_i = \sum_{\substack{j \in C \\ i \neq j}} \frac{1}{1+\tau_{i,j}} \quad \forall\, i \in C \tag{14}$$

$$f_{i,j} = \frac{c_i^a}{1+\tau_{i,j}} \quad \forall\, i, j \in C \tag{15}$$

Algorithm 1
Initial truck route generation.

```
For all customers $i$ do
  Calculated centrality $c_i$ using equation (14)
$a \leftarrow 0$
For $p = 1$ to $PopulationSize$ do
  $a \leftarrow a + 2/PopulationSize$
//Calculating fitness value
  For all customers $i, j$ ($i \neq j$) do
    Calculated centrality fitness $f_{i,j}$ using equation (15)
  For $i = 1$ to $c$ do
    Add customer $j$ with max($f_{i,j}$) to the end of the $R_p{}^T$
    Update the customer $j$ as visited
Return ($R^T$)
```

4.2. Phase I- construction heuristic

The first phase begins by assigning drone-customers by following the constructive rules through Algorithm 2. For every drone eligible customer C_d, we evaluate suitability S for the drone by comparing the spent time between adhered nodes when using the drone (S^D) or the truck (S^T). If the drone takes a shorter time than to the truck, the suitability is positive, and we designate the customer to the drone. When drone routes are duplicated, we keep the better one and dispose of the other. Fig. 6(b) shows an example of an output of this function in which customer 2 is designated to the drone.

Algorithm 2
Drone-customer assignment ($R_p{}^T$).

```
$R_p{}^{T\prime} \leftarrow R_p{}^T$
//Score drone delivery suitability
For $i \in C_d$ do
    $S_i^T \leftarrow \tau_{i\text{-}1,i}{}^t + \tau_{i,i+1}{}^t$
    $S_i^D \leftarrow \max\,(\tau_{i\text{-}1,i+1}{}^t,\ \tau_{i\text{-}1,i}{}^d + \tau_{i,i+1}{}^d)$
//get suitability
    $S_i \leftarrow S_i^T\text{-}S_i^D$
//if the suitability is negative, dispose
  If $S_i < 0$
    $S_i \leftarrow \phi$
//if Suitability is positive, designate the customer to drone
  else $S_i > 0$
    $R_{p,i}{}^D \leftarrow R_{p,i}{}^{T\prime}$
//if there is overlapped, keep better one
  If $R_{p,i}{}^D \neq \phi$ && $R_{p,i\text{-}1}{}^D \neq \phi$
    If $S_i > S_{i\text{-}1}$
      $R_{p,i\text{-}1}{}^{T\prime} \leftarrow R_{p,i\text{-}1}{}^D$
    else
      $R_{p,i}{}^{T\prime} \leftarrow R_{p,i}{}^D$
Return ($R_p{}^D$, $R_p{}^{T\prime}$)
```

In the following procedure, based on the assigned drone-customers and the truck route, launch and rendezvous points, *LP* and *RP*, respectively are chosen and overall delivery time is evaluated. The launch and rendezvous points are selected based on maximum time-saving compared to the truck. After deciding these points, we check the energy limit and switch the customer to the truck if the travel is overloaded for the drone. Fig. 6(c) shows an example of the generated drone route.

With the constructed drone routes and truck routes, the arrival time to the ending depot is evaluated and recorded in T_p. The pseudocode for this process is shown in Algorithm 3.

Algorithm 3
Drone route construction ($R_p^{T\prime}$, R_p^{D}).

```
//evaluation the truck route first
For $i = 2$ to $n$ do
    $T_i^t \leftarrow T_{i-1}^t + \tau_{i-1,i}^t$
//evaluation the drone route first
For $i = 2$ to $n$-$1$ do
  If $R_{p,i}^D \neq \phi$
//calculate the drone arrival time at delivery point
    For $j \in R_p^T$ $(i > j > RP_{i-1})$ do
      $T_i^d = T_j^d + \tau_{j,i}^d + L$
      For h $\in NZ$ do
        If $(T_j^d < et(h))$ && $(T_i^d > st(h))$
          $T_i^d = T_i^d + d_{j,i,h}$
//calculate the drone arrival time at rendezvous points
    For $k \in R_p^T$ $(i < k < O_{i+1})$ do
      $T_k^d = T_i^d + \tau_{ik}^d + R$
      For h $\in NZ$ do
        If $(T_i^d < et(h))$ && $(T_k^d < st(h))$
          $T_k^d = T_i^d + d_{i,k,h}$
//Find the best launch and rendezvous point, $LP_i$, $RP_i$, with maximum saving
    $LP_i$, $RP_i \leftarrow j$, $k$ with max $((T_i^d - T_j^t + T_k^d - T_i^d) - (T_k^t - T_j^t + L + R))$
//Check the energy limit
  If $p(w_i)\ (T_i^d - T_j^t) + p(0)\ (T_k^d - T_i^d) > \eta E$
    $R_{p,j}^{T\prime} \leftarrow R_{p,j}^D$
    For $j = i$ to $n$ do
      $T_j^t \leftarrow T_{j-1}^t + \tau_{j-1,j}^t$
  Else
//Evaluate arrival time after rendezvous
    $T_{RPi}^d \leftarrow$ max $(T_{RPi}^t, T_{RPi}^d)$
    For $j = RP_{i+1}$ to $n$ do
      $T_j^t \leftarrow T_{j-1}^t + \tau_{j-1,j}^t$
//Recode the arrival time to the ending depot
$T_p \leftarrow T_n^t$
Return $(T_p)$
```

4.3. Phase II-Search heuristic

In the second phase, the result of phase I is improved through several search operations (Algorithm 4). We implemented different operators to diversify the search area by systematically perturbing the current solution. These operators work for truck route R^T only because the drone routes are decided by the truck routes in phase I.

The operators work with two customers *I,* and *J* that were randomly selected from the truck routes. The 2-opt operator, which has been shown to perform well as a local search, especially in routing problems (Ting and Liao, 2013), reverses the sequence of visits to a set of customers. The Swap operator swaps two customers that are selected as *I* and *J,* and the Insertion operator moves the chosen node *I* next to node *J*. These three operators are shown in Fig. 7. The last operator, Cross-move, exchanges routes from *I* to *J* (step 1). For the duplicated nodes after the exchange, the inverse replacement is applied outside of the *I*, *J* (step 2). This idea is based on a partially matched crossover studied by Oliver et al. (1987) and Goldberg (1989), and its procedure is depicted in Fig. 8.

Algorithm 4
Search Operation (R^T, T).

```
$R_s^T \leftarrow$Roulette wheel selection $(R^T, T)$
//generate random point $I$, $J$ $(I < J)$
$I$, $J \leftarrow$random point $I < J$
$R_{(1\ \mathrm{to}\ PopulationSize/5)}^T \leftarrow R_s^T$
$R_{(PopulationSize/5+,2\cdot PopulationSize/5)}^T \leftarrow$ 2-opt routes between points $I$, $J$ of $R_s^T$
$R_{(2\cdot PopulationSize/5+,3\cdot PopulationSize/5)}^T \leftarrow$ Swap routes between points $I$, $J$ of $R_s^T$
$R_{(3\cdot PopulationSize/5+,4\cdot PopulationSize/5)}^T \leftarrow$ Insertion routes between points $I$, $J$ of $R_s^T$
$R_{(4\cdot PopulationSize/5+,5\cdot PopulationSize/5)}^T \leftarrow$ Cross-move routes between points $I$, $J$ of $R_s^T$
Return $(R^T)$
```

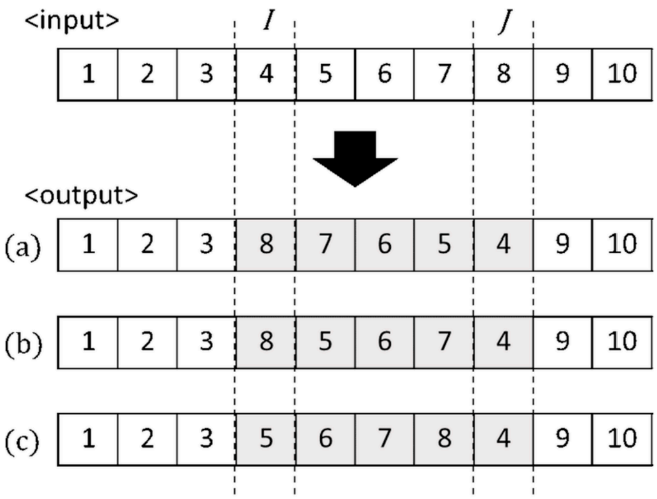


Fig. 7. Search operations (a) 2-opt, (b) Swap and (c) Insertion.

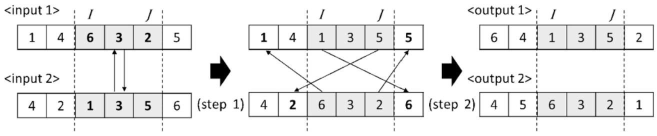


Fig. 8. Cross-move procedure.

4.4. Termination conditions

If the algorithm cannot improve a current best solution in a predefined number of iterations or it reaches the maximum number of allowed iterations, the algorithm terminates.

5. Computational results

A series of numerical experiments were conducted to evaluate the performance of the proposed heuristic, TPCSA and to gain insights into its potential applicability. The TPCSA was evaluated in a collection of generated problems, adopted by Murray and Chu (2015), which has customers randomly distributed across 8 miles squared region. The computation was performed on a 3.1 GHz computer with 4 GB RAM. Where applicable, MILP models were solved via IBM ILOG CPLEX 12.7. The TPCSA and other comparing heuristics were coded in MATLAB version R2016a.

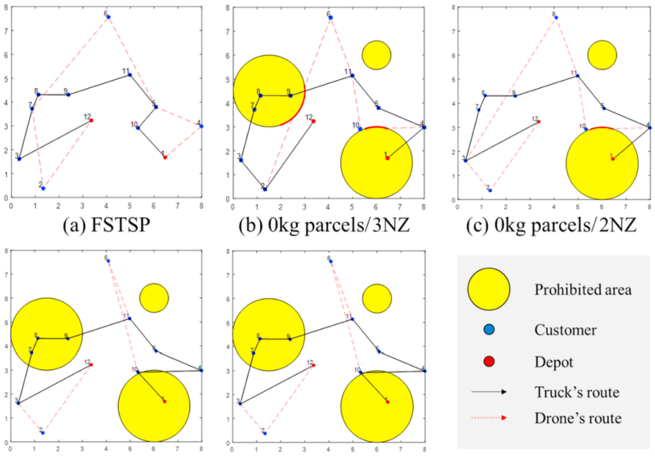
5.1. Solution comparison of FSTSP and FSTSP-ECNZ

This section examines the differences between the existing model FSTSP (Murray and Chu, 2015) and FSTSP-ECNZ to determine the impact of parcel weights and no-fly zone considerations. The test problem was generated to contain 10 customers that were randomly distributed across 8 miles squared region. The weight of the customers' packages differed from 0 kg to 4 kg, and three no-fly zones were generated. The first area was designed to have a 0.6-mile radius, and two others had a 1.5-mile radius. For a clear comparison, all no-fly zones were set to be valid all the time ($st = 0$, $et = \infty$) and all customers were designed to be drone-eligible. We used the 'MK8-3500' drone, as mentioned in section 3.1.1. The speed of the UAV and truck were selected as 35 miles/h with a Euclidean path and 25 miles/h with the Manhattan metric, respectively. The value of big-M is set to 350 that effectively renders all corresponding constraints, taking into account data such as travel time and detour time.

Table 2 shows the computational results of FSTSP (Murray and Chu, 2015) and FSTSP-ECNZ obtained by MILP. The "Gap (%)" indicates the percentage difference between the optimal solution of FSTSP, 63.62,

Table 2
Computational results of FSTSP and FSTSP-ECNZ.

Test set	FSTSP (Murray and Chu, 2015)		FSTSP-ECNZ											
			Without No-fly zone			With 1 No-fly zone			With 2 No-fly zones			With 3 No-fly zones		
weight	Obj. value	CPU time	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)
0 kg	63.62	29516	63.62	42003	0.00	63.62	58524	0.00	68.55	48357	7.75	74.58	43023	17.22
2 kg	N/A	N/A	63.62	31680	0.00	63.62	46526	0.00	69.46	45575	9.19	74.58	31780	17.22
3 kg	N/A	N/A	63.62	30261	0.00	64.28	32513	1.05	70.50	37517	10.81	76.85	29291	20.80
4 kg	N/A	N/A	64.28	25640	1.05	64.28	17675	1.05	71.41	45235	12.25	77.91	24049	22.46



(a) FSTSP (b) 0kg parcels/3NZ (c) 0kg parcels/2NZ

(d) 3kg parcels/ 3NZ (e) 4kg parcels/3NZ

Fig. 9. Optimal route of FSTSP and FSTSP-ECNZ.

and each solution obtained by FSTSP-ECNZ. Obviously, FSTSP (Murray and Chu, 2015) is applicable only for 0 kg weight and without the no-fly zone. The result shows that the objective function increases consistently with a rise in shipping weights or the number of no-fly zones.

As shown in Table 2 and Fig. 9, the FSTSP-ECNZ and FSTSP have a distinct difference in their route and objective value. Since the FSTSP is a special case of FSTSP-ECNZ with no parcel weight and the no-fly zone, it can be defined as a generalized version of the FSTSP-ECNZ. In other words, the solution obtained by the FSTSP can only be used in limited circumstances that are rarely found in reality. The FSTSP-ECNZ, meanwhile, is suitable for practical application.

5.2. Optimality comparison of solution approaches

In this section, the proposed algorithm (TPCSA) is evaluated on a collection of 30 test problems. These problems were generated by a similar setting to section 5.1 with a different delivery weight and no-fly zones. The weights were randomly selected following a normal distribution with a mean of 1.6 pounds and a standard deviation of 3.9 pounds, such that 86% were below 5 pounds as with an Amazon delivery (Wang, 2016). The three no-fly zones from section 5.1 were generated to be valid for 60 min, randomly beginning from 0 to 30 min delivery began.

To evaluate the performance of TPCSA, four different well-known meta-heuristics were also tested. The first three heuristics were the GA, PSO, and SA taken from San et al. (2016), Wang et al. (2003), and Ponza (2016). The last heuristic is the NN approach, which represents a simple local search. The initial truck routes of the first three heuristics are generated by the NN. Refer to algorithm 5, 6, 7 and 8 in Appendix A-D for a detailed description with pseudo code for comparing meta-heuristics. For each algorithm except NN which has a single iteration, population size and maximum iterations were set to 100 and 10,000 and stopped searching if no improvement occurred within 100 iterations. Detail parameters for the proposed algorithms are described in Table 3.

Table 4 illustrates performance for the various approaches after running 30 different test set by each algorithm, where “MILP” represents the use of the MILP formulation via CPLEX and “TPCSA”. “GA”, “PSO”, “SA” and “NN” represent the heuristics evaluated. The “GAP (%)” in Table 4 indicates the percentage difference between the optimal solution obtained by the MILP and each heuristic in terms of optimality gap. The big-M was set as 350 which is adequately large and effectively renders the corresponding inequality. Fig. 10 provides the box plots of the percentage gaps.

We observe that the use of “TPCSA” yielded the nearest-optimal results with an average gap of 1.64%. Furthermore, it generates 9 optimal solutions in 30 instances. The “GA” followed closely with an average 5.44% gap and 3 optimal solutions obtained. Other approaches showed relatively inferior performances with around 10% average gaps and failed to derive any optimal solutions. Fig. 10 shows that the “TPCSA” solution has a lower gap and a small standard deviation. On the other hand, “SA” and “NN” show relatively unstable solution quality but take considerably less computing time. “TPCSA” and “GA” show shorter computation times than “PSO”.

5.3. Performance comparison with large size problems

In this section, accessibility for the large size problems was tested. We ran each heuristic for 10 replications and increased the size of the problem by changing the number of customers while maintaining the same setting as in section 5.2. Since the MILP takes a long computation time even in 10-customer instances at over 8 h on average, in this

Table 3
Parameters for the proposed algorithm.

	Basic Parameters	Maximum iteration	Population size	Terminate condition
TPCSA	Weight value a: 0-2	10,000	100	100
GA	Crossover rate: 0.7 Mutation rate: 0.3	10,000	100	100
PSO	Local/global learning factor:2 Weight for previous solution: 0.9	10,000	100	100
SA	Start temperature: 0.110 End temperature: 0.008	10,000	-	-
NN	-	-	-	-

Table 4
The performance of the approaches in 10-customer instances.

Test set	MILP		TPCSA			GA			PSO			SA			NN		
	Obj. value	CPU time	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)
1	65.86	25477	67.46	6.16	2.43	68.52	1.14	4.04	73.84	4.88	12.11	69.34	3.65	5.28	81.37	0.05	23.55
2	76.99	35412	76.99	5.09	0.00	77.31	3.70	0.41	83.62	7.92	8.61	78.98	1.90	2.58	93.47	0.06	21.40
3	69.78	33857	69.78	5.48	0.00	73.19	2.20	4.88	72.88	4.04	4.45	72.88	1.87	4.44	88.79	0.05	27.24
4	80.82	44765	80.82	2.80	0.00	80.82	2.01	0.00	86.64	4.27	7.20	82.79	1.70	2.44	89.69	0.03	10.97
5	80.81	51453	82.88	3.00	2.56	86.56	4.68	7.12	90.59	3.35	12.10	81.83	1.97	1.26	100.43	0.06	24.27
6	75.22	25153	77.03	4.20	2.41	78.21	2.68	3.98	85.56	5.43	13.74	89.46	2.04	18.92	89.95	0.06	19.58
7	58.69	9735	59.06	2.34	0.63	62.56	1.37	6.59	62.56	2.90	6.59	62.56	2.01	6.59	69.51	0.03	18.43
8	74.22	28127	75.86	1.97	2.21	78.59	2.65	5.89	77.40	3.71	4.28	74.32	1.68	0.14	88.56	0.03	19.32
9	85.01	30566	85.01	2.39	0.00	85.01	1.97	0.00	87.00	4.26	2.34	86.16	1.65	1.35	89.58	0.02	5.38
10	57.34	17064	57.66	2.92	0.55	62.63	2.03	9.23	60.98	4.51	6.35	64.75	2.03	12.92	65.60	0.03	14.40
11	60.14	19136	60.83	1.92	1.15	64.77	1.25	7.70	70.49	4.07	17.22	71.65	1.90	19.14	72.74	0.03	20.95
12	74.13	22072	75.74	2.11	2.17	78.56	2.61	5.97	85.58	3.09	15.44	86.12	1.87	16.17	94.32	0.02	27.23
13	89.20	34278	91.04	2.18	2.06	94.45	2.81	5.89	97.58	3.07	9.39	93.68	1.72	5.03	106.13	0.03	18.98
14	60.57	33338	61.68	3.17	1.84	62.68	2.11	3.49	64.17	3.98	5.94	73.61	1.76	21.53	74.56	0.03	23.10
15	72.05	24682	72.23	1.48	0.25	79.02	1.30	9.68	79.75	3.56	10.69	83.86	1.98	16.40	84.80	0.04	17.70
16	64.60	16634	65.78	1.46	1.82	67.89	4.51	5.10	70.84	3.59	9.66	68.09	1.75	5.40	68.74	0.03	6.41
17	78.37	38963	80.05	3.68	2.15	84.32	3.54	7.59	84.25	5.80	7.50	78.90	1.87	0.68	99.72	0.02	27.24
18	81.28	45107	81.28	3.04	0.00	82.11	3.88	1.02	83.99	4.06	3.34	82.62	1.58	1.65	91.04	0.05	12.01
19	86.56	26140	88.20	1.04	1.89	91.29	2.54	5.46	101.79	2.80	17.59	87.09	1.76	0.62	87.91	0.02	1.56
20	87.26	34167	89.12	5.12	2.13	90.54	4.12	3.76	100.42	4.44	15.08	96.37	1.56	10.44	104.12	0.05	19.32
21	69.14	34210	70.55	2.78	2.04	72.15	5.23	4.35	80.15	3.44	15.92	76.96	2.26	11.31	78.42	0.02	13.42
22	67.35	28099	67.35	3.44	0.00	75.12	3.87	11.54	75.44	3.96	12.01	74.99	1.78	11.35	80.32	0.03	19.26
23	80.53	40526	82.12	1.87	1.97	80.53	6.33	0.00	89.69	3.89	11.37	83.59	1.87	3.79	95.64	0.03	18.76
24	76.44	30758	77.64	1.98	1.57	89.42	3.35	16.98	99.11	2.98	29.66	78.01	1.79	2.06	90.41	0.02	18.28
25	83.00	34937	84.21	2.60	1.46	84.21	2.20	1.46	87.08	3.22	4.92	83.31	1.79	0.38	85.47	0.02	2.98
26	69.91	23997	69.91	1.90	0.00	75.32	2.64	7.74	80.63	4.80	15.33	77.90	2.00	11.43	78.60	0.03	12.43
27	53.99	24027	57.55	2.12	6.59	56.74	2.42	5.09	63.56	4.42	17.73	68.92	1.98	27.66	69.57	0.01	28.86
28	64.12	21344	70.00	1.35	9.17	68.51	2.50	6.84	68.43	4.46	6.72	67.50	1.61	5.27	68.70	0.04	7.14
29	77.89	25750	77.89	1.76	0.00	81.46	2.93	4.58	83.96	2.19	7.79	78.54	1.51	0.84	84.23	0.02	8.14
30	53.30	18477	53.30	1.08	0.00	57.00	2.29	6.95	59.71	3.30	12.03	60.66	1.90	13.80	61.36	0.02	15.12
Avg	72.49	29275	73.63	2.75	1.64	76.32	2.90	5.44	80.26	4.01	10.77	77.85	1.89	8.03	84.46	0.03	16.78

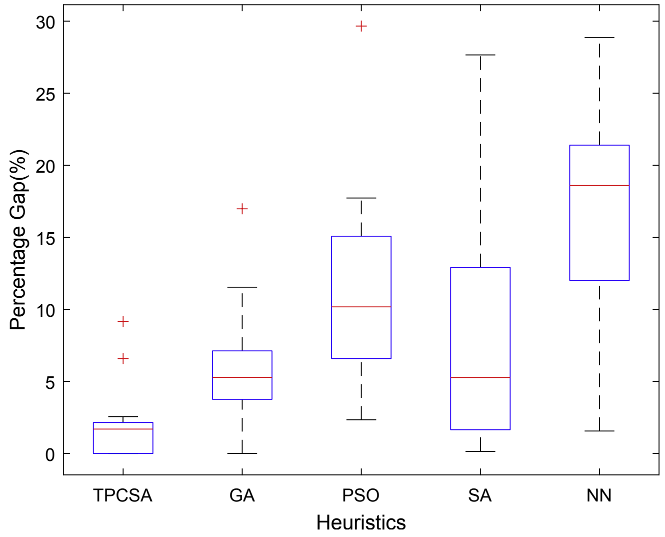
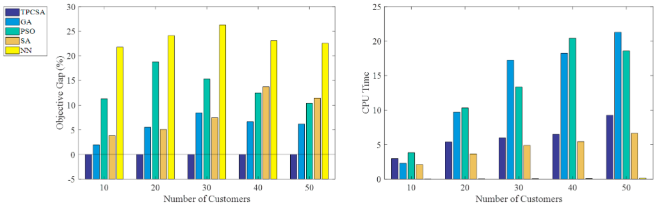


Fig. 10. Results of the heuristics for 10 customers.



(a) Objective gap (%)

(b) CPU Time

Fig. 11. Results of heuristics with the different number of customers.

section, we explored the effectiveness of the proposed algorithm (TPCSA) by comparing it with other heuristics only. Fig. 11(a) illustrates the percentage gap between "TPCSA" and the other heuristics. In Fig. 11(b), the CPU time of each approach for different problem size is compared. Table 5 shows the objective value and CPU time of each heuristics and the percentage gap between "TPCSA" and other heuristics for further comparison.

There was a marked difference in solution quality among the different heuristics. In particular "TPCSA" outperformed all others for every problem size and is followed closely by "GA" with less than 7% average gap. In 10 to 30-customers instance, "SA" performed better than "PSO" but not competitive in larger instances. Beside, "PSO" and "NN", were over 10% compared to "TPCSA", which indicates the effectiveness of the proposed algorithm in the problem.

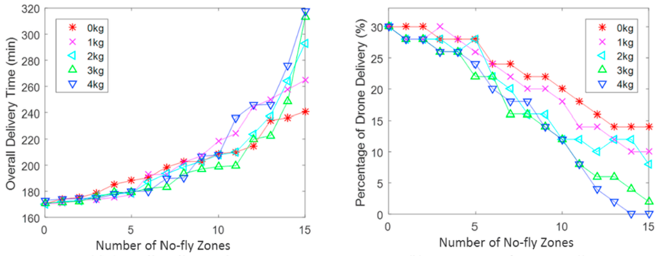
Although "NN" has a relatively poor objective performance, when observing the runtime in Table 5, we note that it is significantly faster than other approaches to solving the problem. Therefore, it may be reasonable to solve a problem with a large size and limited computation time by using "NN". Except "SA" and "NN", "TPCSA" derived a solution in less time than other two approaches and an average of 9.27 s computation time to derive 50 customers' schedules is reasonable considering its solution quality.

5.4. Sensitivity analysis

This section aims to present the sensitivity of the overall delivery time and percentage of drone delivery through different parcel weights and no-fly zones. The "TPCSA" was run on a 50 customers problem for 10 replications, and the best-obtained solutions are reported in Fig. 12. The weight of the packages differed from 0 kg to 4 kg and fifteen no-fly zones that represent category B was generated with no overlap at the same time. Note that increasing the weight and the number of non-flying zones will not always delay the delivery or reduce the delivery

Table 5
A summary of the heuristics' performance for various approaches with the different number of customers (refer Appendix E for detail).

No. of customer		TPCSA		GA			PSO			SA			NN		
		Obj. value	CPU time	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj.value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)
10	Avg	77.34	2.98	78.78	2.30	1.95	86.19	3.84	11.33	80.04	2.10	3.85	94.30	0.03	21.78
	Std	11.34	1.43	11.22	0.94	2.24	14.56	0.74	6.10	10.17	0.59	3.43	16.25	0.02	8.84
20	Avg	109.80	5.40	115.59	9.73	5.56	129.87	10.35	18.76	115.15	3.65	5.09	136.14	0.04	24.11
	Std	6.77	2.02	6.52	4.02	7.49	11.16	4.22	13.01	7.13	0.64	6.74	12.22	0.01	9.96
30	Avg	146.66	6.00	158.87	17.22	8.46	169.00	13.33	15.29	157.00	4.89	7.48	185.67	0.06	26.27
	Std	10.82	0.84	12.92	5.45	6.56	17.41	4.18	9.09	11.75	0.68	9.63	23.48	0.02	8.88
40	Avg	168.48	6.51	179.85	18.24	6.69	189.89	20.42	12.50	190.36	5.44	13.70	206.69	0.07	23.10
	Std	14.09	0.44	18.27	6.94	5.04	22.31	10.29	6.56	20.74	0.29	15.23	15.13	0.02	9.10
50	Avg	194.31	9.27	206.09	21.27	6.21	214.26	18.57	10.42	216.04	6.65	11.48	237.88	0.11	22.56
	Std	10.87	1.86	14.39	7.23	7.31	18.74	7.72	9.54	11.43	0.41	8.00	12.31	0.03	5.25



(a) Overall Delivery Time

(b) Percentage of Drone Delivery

Fig. 12. Results of the sensitivity analysis on parcel weight and the number of no-fly zones.

ratio of UAVs. This discrepancy is due to performance variations caused by the randomness inherent in the algorithm. Although there are some discrepancies, the results clearly show that the increased number of no-fly zones led to an increase in overall delivery time and a decrease in the number of drone sorties. Likewise, the increasing weight of parcels has the same result because the massive weight shortens the drones' flight time. However, from 0 to 5 no-fly zones, the effects of weight on the number of drone sorties is much less than the change in of no-fly zones. Such observations from the sensitivity analysis can guide decision makers to make more economical decisions when utilizing the drone-truck delivery system in various situations.

6. Conclusion

The drone has tremendous potentials, and many international companies are developing and testing real-world uses. However, drones have a fundamental weakness in its flight time and limited loadable products. Since these restrictions limit the application of drones, the simultaneous use of trucks and drones is proposed. By launching a drone from a truck and returning it to a truck, the drone travels with a truck and shares its delivery tasks with relaxed restrictions for the drone.

In this study, the FSTSP was considered to account for drone energy consumption and restricted flight areas in the interest of creating substantial and realistic measures. A mathematical model was developed, and an optimal solution was obtained using the commercial optimization software CPLEX. However, due to the intrinsic complexity of the proposed problem, CPLEX consumes enormous computation time, and there was a limit on deriving delivery schedules promptly. To solve this issue, a heuristic algorithm called TPCSA, which combines two different heuristics, constructive heuristics, and search heuristics, was proposed and evaluated by comparing the performance with other heuristics for TSP. The results indicated that the TPCSA obtains promising results over other heuristics in a reasonable time frame. Sensitivity analysis shows that the increase in package weight and no-fly zones reduce the efficient use of drones because it limits their flight range, especially when the two factors are combined.

The proposed study has several future research directions. In this study, battery exchange time was assumed to be fixed and included in launching time and rendezvous time. However, considerations for dynamic battery charging and energy consumption optimization will provide more realistic routing schedules. Single truck and single drone scenarios can be extended to multiple vehicles and drones. Moreover, there are opportunities to explore the problem with different scales or shapes of no-fly zones and by finding other flight detour paths accordingly. Considering different objective functions could be another interesting topic for handling conflicting criteria in the system. On the other hands, the managerial perspectives of FSTSP can be addressed in the future study to derive managerial guidelines and insights for FSTSP. For examples, the number of trucks and drones can be quantitatively derived to serve customers. Also clustering of FSTSP service area can be designed via mathematical optimization approaches. Comprehensive analyses on the systemic factors of such problems will provide managerial and operational insights.

Appendix A. Pseudo code description of Genetic Algorithm (GA)

We develop the GA with multi-dimension chromosome form. The first and second rows consist of truck route and drone route respectively. Each column represents the sequence of routes. Crossover rate and mutation rate are set to 0.7 and 0.3, respectively.

```
Algorithm 5. Genetic Algorithm (GA)
//initialize truck route and evaluate
  $R^T$, $R^D$ ←Generate initial truck and drone routes
  While (termination condition is not satisfied)
    for $p = 1$ to PopulationSize do
//evaluate total routes
      ($T_p$) ←Evaluation ($R_p{}^T$, $R_p{}^D$)
//select truck route using Roulette Wheel
    $R_s{}^T$, $R_s{}^D$ ←Roulette wheel selection ($R^T$, $R^D$)
//Update best solution
    if $T_{best}$ < BestSolution
      BestSolution ←$T_{best}$
//Crossover
    ($R^D$, $R^T$) ←Crossover ($R^T$, $R^D$, $T$)
//Mutation
    ($R^D$, $R^{T\prime}$) ←Mutation ($R^T$, $R^D$, $T$)
  Return (BestSolution)
```

Appendix B. Pseudo code description of Particle Swarm Optimization (PSO)

In the PSO the parameters of local learning factors c_1 and global learning factor c_2 are set as 2, and weight for the previous solution is set as 0.9. Local and global best positions, P_{local} and P_{global}, are derived from evaluation which also keeps the same process with TPCSA. Velocities for truck and drone routes, v^T, v^D, are obtained by the above parameters, local/global best positions and random number r_1, r_2 as (B.1). With the velocities, swarms are updated and evaluated.

$$v = w\cdot v + c_1\cdot r_1\cdot(P_{local} - R) + c_2\cdot r_2\cdot(P_{global} - R) \qquad \text{(b.1)}$$

```
Algorithm 6. Particle Swarm Optimization (PSO)
//initialize truck route and evaluate
  $R^T$, $R^D$ ←Generate initial truck and drone routes
  Update $P_{local}$ and $P_{global}$
  While (termination-condition is not satisfied)
    for $p = 1$ to PopulationSize do
//update velocity
      ($v_p{}^T$) ← Update velocity ($P_{local}{}^T$, $P_{global}{}^T$, $R_p{}^T$)
      ($v_p{}^D$) ← Update velocity ($Pl_{ocal}{}^D$, $P_{global}{}^D$, $R_p{}^D$)
//update swarm
      ($R_p{}^T$, $R_p{}^D$) ← Update swarm ($R_p{}^T$, $R_p{}^D$, $v_p{}^T$, $v_p{}^D$)
//evaluate total routes
      ($T_p$) ←Evaluation ($R_p{}^T$, $R_p{}^D$)
//Update global fitness and global position
    if $T_{best} < F_{global}$
      $F_{global}$ ←$T_{best}$
      $P_{global}{}^T$, $P_{global}{}^D$ ←$R_{best}{}^T$, $R_{best}{}^D$
//Update local fitness and global position
      $F_{local}$←$T_{best}$
      $P_{local}{}^T$, $P_{local}{}^D$ ←$R_{best}{}^T$, $R_{best}{}^D$
  Return ($F_{global}$)
```

Appendix C. Pseudo code description of Simulated Annealing (SA)

In the SA, termination condition is tied to the start, end temperature that were set as 0.110, 0.008. The cooling rate is calculated as (C.1), and the temperature is continuously updated throughout iterations. In each iteration, one search operator among random relocation, 2-opt, swap, and insertion, is randomly selected to navigate the solution space.

$$CoolingRate = e^{\log(Temperature^{end}/Temperature^{start})/MaxIteration} \tag{c.1}$$

```
Algorithm 7. Simulated Annealing (SA)
//initialize truck route and evaluate
  $R^T$, $R^D$ ←Generate initial truck and drone routes
  $Temperature^{current} = Temperature^{start} \cdot T$
  While ($Temperature^{current} > Temperature^{end} \cdot T$)
//update current Temperature
    $Temperature^{current} = Temperature^{current} \cdot CoolingRate$
//generate random point I, J (I≠J)
    I, J←random point $I \neq J$
//perform random move between I and J
    $R^T$, $R^D$ ←Random move ($R^T$, $R^D$, I, J)
//evaluate total routes
    (T) ←Evaluation ($R^T$, $R^D$)
//update Accept
    $Accept = e$^(($T_{current}$-T)/$Temperature^{current}$)
//update current solution
    if Random[0,1] < Accept
      $T_{current} = T$
//update best solution
      if T < BestSolution
      BestSolution = T
    iteration++
  Return (BestSolution)
```

Appendix D. Pseudo code description of Nearest Neighborhood (NN

The NN is developed by constructively finding a customer that is at closest distance from the current location. After seeking the customer, it decides which vehicle should it use regarding the travel time.

```
Algorithm 8. Nearest Neighborhood (NN)
//initialize truck route and evaluate
  $R^T$, $R^D$ ← Generate initial truck and drone routes
//construct truck route cumulatively
  i←0
  while ($i < n$, i++) do
//find a customer with the shortest path from current location
    For all $j \in C$ ($i \neq j$) do
      $R_i{}^T$←Find j with min ($\tau_{i,j}{}^t$)
      $R_i{}^D$←Find j with min ($\tau_{i,j}{}^d$)
//if drone is faster, find the next customer with the shortest path for drone
    If $j \in C_d$ ($i \neq j$) && $\tau_{i,j}{}^d < \tau_{i,j}{}^t$
```

```
            Update the customer $j$ as visited
            For all $k \in C$ ($i \neq j \neq$ k) do
                $R_i^D$←Find k with min ($\tau_{j,k}{}^d$)
                Update the customer $k$ as visited
            $i++$
    //if the truck is faster, assign the customer to the truck
        else
            Update the customer $j$ as visited
    //Evaluate
        ($T$) ←Evaluation ($R^T$, $R^D$)
        BestSolution ←$T_{best}$
        Return (BestSolution)
```

Appendix E. A performance comparison of solution approaches with various problem sizes

Table E.1
A performance comparison of solution approaches with various problem sizes

No. of customers	Test set	TPCSA		GA			PSO			SA			NN		
		Obj. value	CPU time	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)
10	1	94.56	6.80	94.56	4.68	0.00	110.05	3.35	16.38	96.46	3.81	2.00	122.17	0.06	29.20
	2	59.00	3.00	61.08	2.68	3.54	71.51	5.43	21.22	63.68	1.70	7.95	80.24	0.06	36.02
	3	77.31	4.20	79.51	1.37	2.85	87.77	2.90	13.54	78.54	1.64	1.60	102.56	0.03	32.67
	4	85.52	2.34	91.41	1.54	6.88	97.83	3.71	14.39	85.52	1.95	0.00	106.36	0.03	24.37
	5	61.83	1.97	63.42	1.97	2.57	68.48	4.26	10.76	68.34	1.95	10.53	69.14	0.02	11.82
	6	71.49	2.39	74.14	2.03	3.70	77.19	4.51	7.97	76.95	1.97	7.63	79.27	0.03	10.88
	7	77.07	2.92	77.07	1.25	0.00	81.06	4.07	5.18	80.16	1.76	4.02	87.76	0.03	13.88
	8	76.89	1.92	76.89	2.61	0.00	77.90	3.09	1.31	78.62	1.98	2.25	96.76	0.02	25.84
	9	94.80	2.11	94.80	2.81	0.00	111.64	3.07	17.77	97.09	2.07	2.42	115.07	0.03	21.39
	10	74.89	2.18	74.89	2.11	0.00	78.47	3.98	4.79	74.99	2.15	0.14	83.68	0.03	11.74
	Avg	77.34	2.98	78.78	2.30	1.95	86.19	3.84	11.33	80.04	2.10	3.85	94.30	0.03	21.78
	Std	11.34	1.43	11.22	0.94	2.24	14.56	0.74	6.10	10.17	0.59	3.43	16.25	0.02	8.84
20	1	111.10	4.85	111.75	11.37	0.58	134.59	6.57	21.14	112.97	3.37	1.69	148.26	0.05	33.45
	2	110.92	7.24	111.85	20.00	0.84	113.82	15.60	2.62	129.09	3.43	16.39	138.99	0.03	25.31
	3	111.43	9.78	111.58	8.77	0.13	115.92	19.20	4.02	109.11	3.35	−2.08	120.98	0.03	8.57
	4	100.82	3.82	127.24	11.73	26.20	146.81	11.56	45.61	113.69	4.82	12.77	141.51	0.05	40.36
	5	115.30	3.82	120.64	8.63	4.63	139.41	6.04	20.91	118.61	3.51	2.87	155.16	0.05	34.57
	6	120.79	5.26	122.75	8.50	1.63	128.88	10.28	6.70	124.21	3.53	2.83	132.65	0.05	9.82
	7	96.72	3.03	106.59	10.56	10.21	128.47	7.46	32.83	110.46	3.07	14.21	111.59	0.03	15.37
	8	115.03	4.62	118.79	6.80	3.27	146.31	12.71	27.20	111.55	3.42	−3.02	142.65	0.05	24.01
	9	110.74	7.58	117.67	5.38	6.25	123.27	5.97	11.31	118.18	4.98	6.72	140.17	0.03	26.57
	10	105.17	4.02	107.07	5.57	1.80	121.19	8.13	15.23	103.61	3.03	−1.48	129.47	0.03	23.10
	Avg	109.80	5.40	115.59	9.73	5.56	129.87	10.35	18.76	115.15	3.65	5.09	136.14	0.04	24.11
	Std	6.77	2.02	6.52	4.02	7.49	11.16	4.22	13.01	7.13	0.64	6.74	12.22	0.01	9.96
30	1	141.24	6.57	168.20	11.31	19.08	180.73	10.00	27.96	151.95	4.90	7.58	177.38	0.05	25.59
	2	153.02	6.04	154.46	8.72	0.94	156.00	12.28	1.95	170.13	6.60	11.18	200.96	0.06	31.32
	3	152.38	5.19	163.04	23.09	6.99	188.56	7.60	23.74	141.66	4.40	−7.03	203.41	0.06	33.49
	4	153.12	5.99	172.47	22.90	12.64	185.59	18.52	21.21	175.47	4.51	14.60	175.70	0.09	14.75
	5	127.59	5.82	144.30	15.94	13.10	154.07	10.17	20.75	160.27	3.79	25.62	160.83	0.05	26.05
	6	154.54	5.76	165.18	12.40	6.88	177.21	9.39	14.67	150.95	4.96	−2.32	186.17	0.08	20.47
	7	129.69	5.63	134.35	18.41	3.59	137.79	21.23	6.24	136.85	4.96	5.52	154.31	0.06	18.98
	8	140.47	8.11	148.44	22.54	5.67	149.88	12.34	6.70	160.85	4.82	14.51	161.65	0.05	15.08
	9	152.25	6.04	178.84	24.18	17.46	189.87	15.48	24.71	168.22	4.71	10.49	202.14	0.05	32.77
	10	162.28	4.85	159.39	12.68	−1.78	170.31	16.35	4.95	153.62	5.23	−5.34	234.10	0.08	44.26
	Avg	146.66	6.00	158.87	17.22	8.46	169.00	13.33	15.29	157.00	4.89	7.48	185.67	0.06	26.27
	Std	10.82	0.84	12.92	5.45	6.56	17.41	4.18	9.09	11.75	0.68	9.63	23.48	0.02	8.88
40	1	153.04	6.05	151.65	10.87	−0.91	156.39	15.19	2.19	183.35	5.37	19.81	184.32	0.11	20.44
	2	169.71	5.85	174.55	30.95	2.85	180.34	9.86	6.26	209.85	5.74	23.66	210.21	0.08	23.87
	3	163.22	6.54	176.99	18.11	8.44	199.67	24.80	22.33	175.23	5.09	7.35	193.64	0.08	18.64
	4	139.59	6.38	148.01	18.41	6.03	151.96	16.80	8.86	205.30	5.16	47.07	206.27	0.05	47.77
	5	184.94	6.79	203.33	11.58	9.94	228.96	10.67	23.80	211.10	5.40	14.14	213.85	0.08	15.63
	6	185.35	6.21	207.20	12.01	11.79	213.33	29.69	15.10	172.79	5.99	−6.78	221.95	0.08	19.75
	7	161.11	7.05	186.00	15.15	15.45	188.38	10.14	16.93	155.78	5.74	−3.31	186.40	0.06	15.70
	8	181.31	6.05	178.98	16.29	−1.28	198.61	44.71	9.54	190.25	5.43	4.93	220.79	0.06	21.77
	9	167.61	7.05	179.49	31.23	7.09	184.59	23.40	10.14	174.49	5.10	4.11	197.64	0.06	17.92
	10	178.91	7.08	192.29	17.80	7.48	196.63	18.91	9.90	225.50	5.35	26.04	231.80	0.08	29.56
	Avg	168.48	6.51	179.85	18.24	6.69	189.89	20.42	12.50	190.36	5.44	13.70	206.69	0.07	23.10
	Std	14.09	0.44	18.27	6.94	5.04	22.31	10.29	6.56	20.74	0.29	15.23	15.13	0.02	9.10
50	1	182.26	8.02	179.38	16.15	−1.58	187.14	13.01	2.68	217.84	6.74	19.53	219.41	0.09	20.38
	2	203.47	10.17	198.15	14.46	−2.61	204.80	16.75	0.66	222.59	7.19	9.40	235.34	0.09	15.66
	3	201.98	7.36	207.08	12.53	2.53	217.84	20.98	7.85	212.74	6.13	5.33	243.44	0.08	20.52
	4	210.56	7.47	219.23	33.82	4.12	228.99	14.23	8.75	226.71	6.80	7.67	255.40	0.09	21.29
	5	205.59	8.69	206.86	14.45	0.62	211.23	14.66	2.75	216.75	6.26	5.43	235.12	0.11	14.37
	6	184.48	7.89	196.86	22.00	6.71	200.61	14.20	8.75	223.22	6.65	21.00	225.65	0.11	22.32

(continued on next page)

Table E.1 (*continued*)

No. of customers	Test set	TPCSA		GA			PSO			SA			NN		
		Obj. value	CPU time	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)	Obj. value	CPU time	Gap (%)
	7	191.15	9.22	230.04	29.39	20.35	257.56	12.07	34.75	184.73	5.94	−3.36	240.71	0.09	25.93
	8	200.82	13.98	222.13	31.06	10.61	224.34	38.84	11.71	223.25	6.85	11.17	256.41	0.08	27.68
	9	184.64	9.86	192.14	19.81	4.06	197.51	15.94	6.97	211.23	7.27	14.40	245.57	0.19	33.00
	10	178.16	10.03	208.99	19.00	17.31	212.56	25.04	19.31	221.34	6.66	24.24	221.72	0.12	24.45
	Avg	194.31	9.27	206.09	21.27	6.21	214.26	18.57	10.42	216.04	6.65	11.48	237.88	0.11	22.56
	Std	10.87	1.86	14.39	7.23	7.31	18.74	7.72	9.54	11.43	0.41	8.00	12.31	0.03	5.25

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transport. Sci. 1–17 2018.

Banker, S., 2013. Forbes Magazine. Amazon and Drones. Here Is Why it Will Work. Available from: www.forbes.com/sites/stevebanker/2013/12/19/amazon-drones-here-is-why-it-will-work/#4024059a5e7d.

Bryan, V., 2014. Reuters. Drone delivery: DHL 'parcelcopter' flies to German isle. Available from: www.reuters.com/article/us-deutsche-post-drones/drone-delivery-dhl-parcelcopter-flies-to-german-isle-idUSKCN0HJ1ED20140924/.

Canny, J., Reif, J., 1987. New lower bound techniques for robot motion planning problems. In: Foundations of Computer Science, 1987., 28th Annual Symposium on. IEEE, pp. 49–60.

Carlsson, J.G., Song, S., 2017. Coordinated logistics with a truck and a drone. Manag. Sci. 1–31 2017.

Chao, I-Ming, 2002. A tabu search method for the truck and trailer routing problem. Comput. Oper. Res. 29 (1), 33–51.

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S.G., Bian, L., 2017. Drones for disaster response and relief operations: a continuous approximation model. Int. J. Prod. Econ. 188, 167–184.

de Freitas, J.C., Penna, P.H.V., 2018. A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. Electron. Notes Discrete Math. 66, 95–102.

Derigs, U., Pullmann, M., Vogel, U., 2013. Truck and trailer routing—problems, heuristics and computational experience. Comput. Oper. Res. 40 (2), 536–546.

Di Franco, C., Buttazzo, G., 2015. Energy-aware coverage path planning of UAVs. In: Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference on. IEEE, pp. 111–117.

Dillow, C., 2015. Fortune. Meet Matternet, the Drone Delivery Startup That's Actually Delivering. Available from: http://fortune.com/2015/05/01/matternet-drone-delivery.

Dorling, K., Heinrichs, J., Messier, G.G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. IEEE Trans. Syst. Man. Cybernet.: Systems 47 (1), 70–85.

Drexl, M., 2011. Branch-and-price and heuristic column generation for the generalized truckand-trailer routing problem. Revista de Métodos Cuantitativos para la Economía y la Empresa 12.

Drexl, M., 2014. Branch-and-cut algorithms for the vehicle routing problem with trailers and transshipments. Networks 63 (1), 119–133.

FAA, 2017. Fly for Work/Business. Available from: www.faa.gov/uas/getting_started/fly_for_work_business.

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. Ind. Eng. Manag. 9 (2), 374.

Garone, E., Determe, J.F., Naldi, R., 2014. Generalized traveling salesman problem for carrier-vehicle systems. J. Guid. Contr. Dynam. 37 (3), 766–774.

Garone, E., Naldi, R., Casavola, A., 2011. Traveling-salesman problem for a class of carrier-vehicle systems. J. Guid. Contr. Dynam. 34 (4), 1272.

Garone, E., Naldi, R., Casavola, A., Frazzoli, E., 2008. Cooperative path planning for a class of carrier-vehicle systems. In: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on. IEEE, pp. 2456–2462.

Garone, E., Naldi, R., Casavola, A., Frazzoli, E., 2010. Cooperative mission planning for a class of carrier-vehicle systems. In: Decision and Control (CDC), 2010 49th IEEE Conference on. IEEE, pp. 1354–1359.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning.

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

HorseFly, 2014. HorseFly 'Octocopter' primed to fly the future to your front door. Available from: www.uc.edu/news/nr.aspx?id=19929/.

International Business Machines Corporation. CPLEX Optimizer. [On- line]. Available from:(https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer).

Korte, B., Vygen, J., 2012. Combinatorial Optimization, vol. 2 Springer.

Lee, S., 2012. The role of centrality in ambulance dispatching. Decis. Support Syst. 54 (1), 282–291.

Lee, S., 2013. Centrality-based ambulance dispatching for demanding emergency situations. J. Oper. Res. Soc. 64 (4), 611–618.

Lee, S., 2014. Role of parallelism in ambulance dispatching. IEEE Trans. Syst. Man. Cybernet.: Systems 44 (8), 1113–1122.

Lin, S.W., Vincent, F.Y., Chou, S.Y., 2009. Solving the truck and trailer routing problem based on a simulated annealing heuristic. Comput. Oper. Res. 36 (5), 1683–1692.

Liu, J., Reeves, C.R., 2001. Constructive and composite heuristic solutions to the P//Σ Ci scheduling problem. Eur. J. Oper. Res. 132 (2), 439–452.

Lozano-Pérez, T., Wesley, M.A., 1979. An algorithm for planning collision-free paths among polyhedral obstacles. Commun. ACM 22 (10), 560–570.

Mathew, N., Smith, S.L., Waslander, S.L., 2015. Planning paths for package delivery in heterogeneous multirobot teams. IEEE Trans. Autom. Sci. Eng. 12 (4), 1298–1308.

MikroKopter, 2017. Technical specifications. . Available from: www.mikrokopter.de/en/products/nmk8stden/nmk8techdaten.

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

Oliver, I.M., Smith, D., Holland, J.R., 1987. Study of permutation crossover operators on the traveling salesman problem. In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology. L. Erlhaum Associates, Cambridge, MA. Hillsdale, NJ 1987.

Perlow, B., 2016. ABC News. Amazon completes 1st drone delivery. Available from: https://abcnews.go.com/US/amazon-completes-drone-delivery/story?id=44185981.

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. Master's thesis. Università Degli Studi Di Padova, Padova, Italy.

Raja, P., Pugazhenthi, S., 2012. Optimal path planning of mobile robots: a review. Int. J. Phys. Sci. 7 (9), 1314–1320.

San, K.T., Lee, E.Y., Chang, Y.S., 2016. The delivery assignment solution for swarms of UAVs dealing with multi-dimensional chromosome representation of genetic algorithm. In: Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual. IEEE, pp. 1–7.

Scheuerer, S., 2006. A tabu search heuristic for the truck and trailer routing problem. Comput. Oper. Res. 33 (4), 894–909.

Ting, C.K., Liao, X.L., 2013. The selective pickup and delivery problem: formulation and a memetic algorithm. Int. J. Prod. Econ. 141 (1), 199–211.

Tokekar, P., Vander Hook, J., Mulla, D., Isler, V., 2016. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. IEEE Trans. Robot. 32 (6), 1498–1511.

UPS, 2016. Launching lifesaving deliveries by drone in Rwanda. Available from: https://compass.ups.com/drone-medicine-delivery.

Wang, D., 2016. Flexport. The economics of drone delivery. Available from: www.flexport.com/blog/drone-delivery-economics.

Wang, K.P., Huang, L., Zhou, C.G., Pang, W., 2003. Particle swarm optimization for traveling salesman problem. In: Machine Learning and Cybernetics, 2003 International Conference on, vol. 3 IEEE.

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

Workhorse, 2016. First test delivery of the workhorse group HorseFly drone. Available from: http://workhorse.com/newsroom/2016/01/first-test-delivery-of-the-workhorse-group-horsefly-drone/.

Xiao, Y., Zhao, Q., Kaku, I., Xu, Y., 2012. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Comput. Oper. Res. 39 (7), 1419–1431.

Zhang, J., Zhao, Y., Xue, W., Li, J., 2015. Vehicle routing problem with fuel consumption and carbon emission. Int. J. Prod. Econ. 170, 234–242.

Zhuoning, D., Rulin, Z., Zongji, C., Rui, Z., 2010. Study on UAV path planning approach based on fuzzy virtual force. Chin. J. Aeronaut. 23 (3), 341–350.