Center for Advanced Multimodal Mobility

Solutions and Education

Project ID: 2019 Project 07

ASSESSMENT OF PARCEL DELIVERY SYSTEMS USING UNMANNED AERIAL VEHICLES

Final Report

by

Stephen D. Boyles, Ph.D. (ORCID: https://orcid.org/0000-0002-5414-5438)
Associate Professor, Department of Civil, Architectural & Environmental Engineering
The University of Texas at Austin
301 E. Dean Keeton St. Stop C1761, Austin, TX, 78751
Phone: 1-512-471-3548; Email: sboyles@austin.utexas.edu

Tengkuo Zhu (ORCID: https://orcid.org/0000-0001-8365-4349)
Graduate Research Assistant, Department of Civil, Architectural & Environmental Engineering
The University of Texas at Austin
301 E. Dean Keeton St. Stop C1761, Austin, TX, 78751
Email: zhutengkuo@utexas.edu

for

Center for Advanced Multimodal Mobility Solutions and Education
(CAMMSE @ UNC Charlotte)
The University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223

August 2020

ACKNOWLEDGEMENTS

This project was funded by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE @ UNC Charlotte), one of the Tier I University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT), under the FAST Act. Partial support for this work was also provided by the National Science Foundation under Grants 1826230, 1526109/1562291, 1562109, 1826337, 1636154, and 1254921.

DISCLAIMER

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

The idea of deploying electric vehicles and unmanned aerial vehicles (UAVs), also known as drones, to perform "last-mile" delivery in logistics operations has attracted increasing attention in the past few years. In this paper, an electric vehicle travelling salesman problem with drone (EVTSP-D) is formulated as a mixed-integer/linear program to aid logistics organizations with a new method of delivering parcels which can extend the driving range of both vehicles, exploit their advantages and reduce the operation cost. An iterative heuristic algorithm with different search strategies is also developed, which can solve an instance with 25 customers. Results of numerical experiments show that the heuristic is much more efficient than ILOG CPLEX solver and incorporating UAVs into EV-based routing was found to reduce average delivery times by up to 40% for the instances tested. A real-world case study on the Austin network along with the sensitivity analysis of different parameters is also conducted and presented and the results indicate that UAV speed has a greater effect on delivery time compared to UAV operation limit and EV driving range.

Electric Vehicle Travelling Salesman Problem with Drone

The idea of deploying electric vehicles and unmanned aerial vehicles (UAVs), also known as drones, to perform "last-mile" delivery in logistics operations has attracted increasing attention in the past few years. In this paper, an electric vehicle travelling salesman problem with drone (EVTSP-D) is formulated as a mixed-integer/linear program to aid logistics organizations with a new method of delivering parcels which can extend the driving range of both vehicles, exploit their advantages and reduce the operation cost. An iterative heuristic algorithm with different search strategies is also developed, which can solve an instance with 25 customers. Results of numerical experiments show that the heuristic is much more efficient than ILOG CPLEX solver and incorporating UAVs into EV-based routing was found to reduce average delivery times by up to 40% for the instances tested. A real-world case study on the Austin network along with the sensitivity analysis of different parameters is also conducted and presented and the results indicate that UAV speed has a greater effect on delivery time compared to UAV operation limit and EV driving range.

***Key words*: Traveling Salesman, Electric vehicle, Unmanned aerial vehicle, Transportation logistics**

I. Introduction

In the United States, the transportation sector generates 28.9% of the national greenhouse gas emissions (EPA, 2018). Many local governments and corporate policies aim to promote transportation modes with lower emissions of pollution and greenhouse gases. Electric vehicles (EVs) are an emerging alternative to internal combustion engines, and several companies have started to use EVs in their operations. For example, in 2018, FedEx announced a fleet expansion and added 1,000 electric delivery vehicles to operate commercial and residential pick-up and delivery services in the United States (FedEx, 2018). Switching to electric fleets not only has long-term effects on mitigating the impact of climate change but may also have immediate financial benefits, as fuel cost accounts for 39% to 60% of operating costs in the trucking sector (Sahin et al., 2009). Compared to conventional internal combustion engines and petroleum-fuel powered vehicles, EVs are much more energy-efficient and require less maintenance, which indicates potential savings to freight and logistics companies (Howey et al., 2011, Ma et al., 2012).

Another new trend in recent years is the integration of UAVs into the operation of e-commerce and on-demand item delivery. The use of UAVs for "last-mile" parcel delivery promises to change the landscape of the logistics industry. Amazon, Google, DHL all announced plans to use UAVs to deliver small packages, and Google has conducted thousands of test flights in Australia. The past few years have witnessed a dramatic increase in UAV applications (DroneZon, 2019). According to Teal Group's prediction, commercial use of UAVs will grow eightfold over the decade to reach US $7.3 billion in 2027 (Teal Group, 2018).

However, EVs and UAVs have limited range and require access to recharging stations. A commercial UAV has a range of about only 10 miles. Because it has a relatively small service area, distribution centers are needed to aid UAV delivery (Chauhan et al., 2019, Hong et al., 2018, Hoareau et al., 2017). For the most common EVs used in service operations, the minimum charging time is 0.5 h, and the battery capacity is around 22 kWh, which indicates a nominal driving range of 142 km, approximately one-fourth of a petroleum-powered vehicle (Pelletier et al., 2017). Moreover, the driving range is also affected by road slope, driving speed, loading capacity, and the use of peripherals (De Cauwer et al., 2015). Although this problem could be alleviated in the future by carrying higher-capacity batteries, at present, an EV may need to visit charging stations to charge its battery and extend its driving range.

In this paper, we investigate a new problem called the electric vehicle traveling salesman problem with drone

(EVTSP-D), where an electric truck performs deliveries with a UAV in a cooperative way. In this problem, the EV and the UAV could perform delivery tasks simultaneously. The EV serves as the UAV hub, where the UAV can refresh its battery and be loaded with new parcels. Due to driving range limits, EV may need to visit multiple charging stations between customer visits during its daily operation. Note that some charging stations may be visited multiple times, while others may not be visited at all. There are two key differences between this research and recent work on the flying sidekick traveling salesman problem (Murray and Chu, 2015). We model EV and associated battery range constraints and recharging whereas Murray and Chu (2015) model a regular truck. Another key assumption in this problem, the other source of distinction from Murray and Chu (2015), is that the EV and the UAV share their electricity, that is, there is a battery capacity for both EV and UAV, and when the UAV is launched from the EV, the remaining electricity of EV also decreases. The main contributions of the paper are:

- The EVTSP-D is introduced and formulated;
- An efficient heuristic algorithm is proposed to solve EVTSP-D;
- Computational experiments demonstrate the improvement of delivery time by utilizing a UAV;
- The numerical analyses indicate the proposed heuristic can obtain good solutions in a much shorter time than a commercial solver;
- The real-world case study illustrates that the proposed heuristic is capable of solving EVTSP-D of practical size within minutes.

The rest of the paper is organized as follows. A literature review of EVRP and UAV is given in Section 2, focusing on current approaches of using EV and UAV to perform delivery tasks. The problem description of EVTSP-D and its mixed-integer linear programming (MILP) formulation are shown in Section 3. An efficient iterative heuristic algorithm is proposed in Section 4. Section 5 presents computational experiments on random instances, performance comparisons with a commercial solver and a real-world case study. The conclusion and future research are presented in Section 6.

II. Literature Review

The vehicle routing problem (VRP) and the traveling salesman problem (TSP), are among the most well-studied optimization problems in operations research. This problem was first proposed by Dantzig and Ramser (1959). Since then, many variants have been considered, incorporating service time windows, capacities, maximum route lengths, distinguishing pickups and deliveries, fleet inhomogeneities, and so forth. Various exact and heuristic methods have been proposed to solve the problem (Baldacci et al., 2012, Laporte, 2009, Desaulniers et al., 2010, Osman, 1993, Gendreau et al., 1992). Braekers et al. (2016), Montoya-Torres et al. (2015), Pillac et al. (2013), Toth and Vigo (2014), Eksioglu et al. (2009), Golden et al. (2008), Toth and Vigo (2002) provide a thorough literature review of VRP variants and solution algorithm families.

Erdoğan and Miller-Hooks (2012) introduced the green vehicle routing problem (G-VRP), where the goal is to route a fleet of Alternative Fueled Vehicles (AFV) to serve a set of customer's within a time limit while respecting the driving range of the vehicles. The AFVs are allowed to extend their driving range by visiting refueling stations more than once. Conrad and Figliozzi (2011) developed the recharging vehicle routing problem where vehicles can recharge at particular customer locations. Conrad and Figliozzi (2011) also considered customer time windows and fleet capacity constraints. Schneider et al. (2014) also modeled customer time windows and fleet capacity constraints in Electric VRP (E-VRP) problem while making the recharging times dependent on the remaining charge levels. Montoya et al. (2017) extended the previous E-VRP models to consider nonlinear charging functions by using piecewise linear approximations. The solution methods for E-VRP variants are diverse and range from exact methods such as branch and bound, branch and cut (Koç and Karaoglan, 2016), and branch and price (Schneider et al., 2014, Hiermann et al., 2016); heuristic methods such as modified savings method of Clarke and Wright with density-based clustering (Erdoğan and Miller-Hooks, 2012), local improvement based on neighborhood swap (Schneider et al., 2014, Masmoudi et al., 2018), and metaheuristics such as simulated annealing and tabu search (Keskin and Çatay, 2016, Goeke and Schneider, 2015, Felipe et al., 2014). Pelletier et al. (2017) and Erdelić and Carić (2019) provide a comprehensive survey of the different variants of the electric vehicle routing problem and associated solution algorithms. *Unlike the above-mentioned research, this paper focuses on the traveling salesman variant*. Doppstadt et al. (2016) formulated the traveling salesman problem for hybrid electric vehicles considering four modes of operation - combustion, electric, charging, and boost. An iterated tabu search with local search operators which switches route structure, as well as operating modes, was used to solve real-world instances. Doppstadt et al. (2019) extend Doppstadt et al. (2016)'s model by considering customer time windows, and proposed a new variable neighborhood search based solution method. Liao et al. (2016) provided an efficient dynamic programming based polynomial-time algorithm for the electric vehicle shortest travel time path problem and approximation algorithms for the EV touring problems. The algorithms incorporated battery capacity constraints and battery swaps. Roberti and Wen (2016) provided a mixed-integer linear programming formulation for the electric vehicle traveling salesman problem with time windows for both full and partial recharge policies. A three-phase heuristic employing variable neighborhood descent to reach time window feasibility and minimize cost tour and a dynamic programming algorithm to achieve feasibility concerning battery capacities is developed. *While there has been a significant amount of work on E-VRP and TSP, none of them have considered an integrated delivery system with drones.*

Meanwhile, an increasing number of studies investigate the efficiency of delivery systems that deploy UAVs. Otto et al. (2018) provide a detailed review of the various civil applications of drones in domains such as agriculture, monitoring, transport, security, etc. Murray and Chu (2015) introduced the flying sidekick traveling salesman problem (FSTSP) which assumes that a truck can launch its UAV at the depot or customer node and remains on its route, while the UAV delivers one small parcel to another customer before meeting again at a rendezvous location (another customer node on the truck's route). Murray and Chu (2015) proposed a two-stage route and reassign heuristic wherein the first stage a truck TSP tour which visits all customers is determined. In the second stage, select customers are reassigned to UAV based on cost savings. Murray and Chu (2015) also introduced the parallel drone scheduling Traveling Salesman Problem (PDSTSP), where multiple drones and a truck originating from a depot serve a set of customers. In the PDSTSP heuristic, customers are partitioned into those that can be served by UAVs, and the remaining customers are assumed to be served by truck. A parallel machine scheduling problem is solved to determine customer assignments to drones. A swap-based heuristic is used to exchange customers from UAV and truck partitions to improve the solution. Mbiadou Saleu et al. (2018) developed an iterative two-stage heuristic involve customer partitioning and routing optimization for the PDSTSP. Agatz et al. (2018) developed an integer programming formulation for a variant of FSTSP called Traveling Salesman Problem with Drones (TSPD) and a "route-first, cluster-second" heuristic which constructs a TSP with drone tour from a TSP tour. A subtle difference between FSTSP and TSPD is that in FSTSP, the drone departs from a truck at a node and joins the truck at a different node whereas in TSPD, the truck can wait at a node and the drone can rejoin the truck at the same node it departed from. Note that several authors have used TSPD while referring to FSTSP. Ha et al. (2018) studied a variant of Murray and Chu (2015) with the objective of minimizing operating and waiting time costs rather than completion time. The authors propose two heuristics - a modification of Murray and Chu (2015)'s heuristic to minimizing costs and Greedy Randomized Adaptive Search Procedure (GRASP). Es Yurek and Ozmutlu (2018) developed an iterative two-stage algorithm to solve Murray and Chu (2015)'s FSTSP, which was referred to as TSPD. In the first stage, the truck route is determined, whereas in the second stage, the drone tours are determined. Bouman et al. (2018) modify the Bellman-Held-Karp dynamic programming algorithm for the TSP to develop an exact solution approach for TSPD, whereas Poikonen et al. (2019) use a branch and bound method. De Freitas and Penna (2020, 2018) developed a randomized variable neighborhood descent heuristic, which modified an initial TSP solution obtained from Concorde solver to solve the FSTSP. Boysen et al. (2018) focus on scheduling single and multiple drone deliveries launched from a truck with a fixed route. Jeong et al. (2019) modify Murray and Chu (2015)'s model to include the impact of payload on energy consumption and no-fly zones and propose a two-stage construction and search heuristic. Dayarian et al. (2020) focused on a new variant where drones are used to resupply a truck making deliveries. Kim and Moon (2018) study a variant termed Traveling Salesman Problem with Drone Station (TSPDS) where a truck is used to resupply a drone station, which is different from a depot. Multiple drones then make deliveries to customers from drone stations. The truck will also make deliveries to customers after supplying the drone station. A two-phase solution algorithm is developed, which involves determining optimal TSP for customers who can be served by truck only and parallel machine scheduling problem to determine customer assignment to drones. *While there has been a significant body of work on integrating drones into existing routing frameworks since 2015, none of them consider EVs and their associated range constraints.*

Other researchers have used continuous approximation techniques to analyze UAV routing problems. Carlsson and Song (2018) used a continuous approximation approach to study improvements in efficiency with using a drone with a traveling salesman problem framework. Based on asymptotic as well as computational analysis, the improvements in efficiency were found to be proportional to the square root of the ratio of the speeds of the truck and the UAV. Ferrandez et al. (2016) determined that using multiple drones per truck led to an increase in savings in energy and time and developed continuous approximation formulas to estimate the savings. Figliozzi (2017) compared lifecycle CO_2 emissions of drones relative to other delivery mechanisms such as diesel vans and electric trucks. UAVs were found to have lower lifecycle CO_2 emissions per distance compared to typical diesel vans.

Several researchers have focused on VRP variants involving drones. Dorling et al. (2016) formulated drone delivery problems as a multi-trip VRP. Key contributions were a linear approximation of energy consumption as a function of payload and a simulated annealing based solution algorithm. Wang et al. (2017) derive worst-case bounds on the maximum savings obtained by integrating drones into traditional truck deliveries. Ham (2018) adopt a constraint programming approach where multiple drones and trucks depart from a single depot. Drones can deliver as well as pick up while considering customer time windows. Ulmer and Thomas (2018) model deliveries using a heterogeneous fleet of trucks and drones from a single depot and found that spatial partitioning of the delivery zones into those delivered by trucks and those delivered by drones are more effective. Wang and Sheu (2019) studied a VRP with a drone variant where drones can be launched from a truck, serve multiple customers, and then return to docking hub from which they can be picked up by the same or different trucks. A branch-and-price formulation is developed to solve the mixed-integer linear program. Sacramento et al. (2019) formulated the VRP variant of FSTSP where multiple truck and UAV combinations are used to serve customers and solved the model using an adaptive large neighborhood search metaheuristic.

In this paper, a new delivery method that deploys an electric vehicle and UAV is investigated. This new method combines the advantages of EVs' with those of UAVs', and forms an integrated route. From a modeling perspective, we are studying a variant of FSTSP with EV and a drone making deliveries in tandem. Unlike the FSTSP which uses regular trucks, we model EV and therefore the battery capacity of the truck is taken into consideration. The EV can also extend its range by charging at charging stations. The battery on the EV can be used by the drone for charging. Therefore, in our model, the truck and the drone share electricity.

III. Problem Description and Mathematical Formulation

The EVTSP-D is defined on an undirected, complete graph $G = (V, E)$, with a vertex set V consisting of the customer set $I = \{v_1, v_2, ..., v_c\}$, the depot v_0, and a set of charging stations (CS) $S = v_{c+1}, v_{c+2}, ..., v_{c+s}$. The vertex set is thus $V = \{v_0\} \cup I \cup S$ and $|V| = c + s + 1$. It is assumed that all charging stations have unlimited capacities. The edge set $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ contains the edges connecting vertices of V. Each edge (v_i, v_j) is associated with two non-negative travel time τ_{ij} and d_{ij}, which corresponds to the travel time needed for the EV and UAV to travel from node i to node j, respectively. In addition, no limit is set on the number of stops that can be made for recharging.

During recharging, we assume that the battery is charged to its full capacity every time an EV reaches a CS node. This represents, for example, instances when depleted batteries are swapped for fully-charged ones at stations. We further assume that this battery swap process happens instantly. The latter assumption can be relaxed into "full-charge-fixed charging-time" by adding a constant term to the CS node departure time in the MIP formulation we introduce later. This “full-charge"”policy assumption is commonly adopted in the literature (Erdoğan and Miller-Hooks, 2012, Conrad and Figliozzi, 2011), not only for simplicity of the formulation, but also because battery swapping (in few minutes) is more efficient than fast-charging (usually less than 30 minutes) and avoid the adverse impact of random charging on power grid operation (Ahmad et al., 2020).

The goal of EVTSP-D is to find a coordinated tour that starts and ends at the depot and visits a subset of vertices (including charging stations when necessary) such that the total delivery time is minimized. All customers must be served once, either by the electric vehicle or by the UAV. At the start of the tour, all parcels are loaded into the electric vehicle. During the operational process, the electric vehicle can launch the UAV at a vertex and retrieve the UAV later at another vertex. The electric vehicle and the UAV operate independently after the UAV is launched. Note that both the launch vertex and retrieve vertex should be visited by the EV.

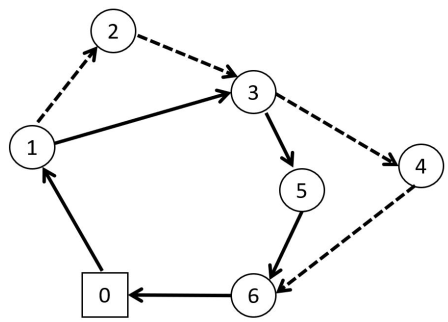

Figure 1 A simple representation of the EV-UAV coordinated route

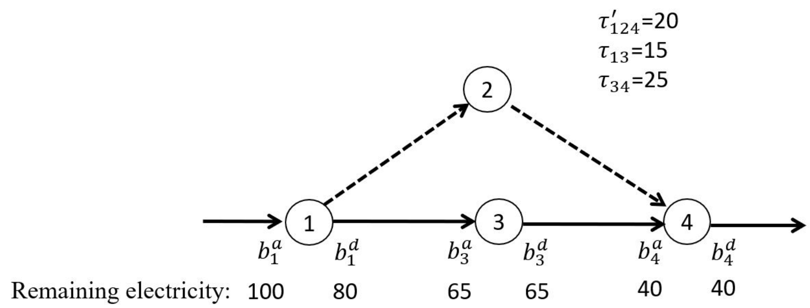

Figure 2 A simple representation of the electricity assumption

In this work, we assume that there is only one EV and one UAV in operation at any given time. There may be multiple UAVs on the EV (which may be charging while another is in flight), but launching and retrieving several UAVs simultaneously introduces significant complications, both algorithmically and practically (coordinating the retrieval of multiple UAVs within their range limit becomes difficult if any of them is delayed). A similar setting can be seen in Wang and Sheu (2019), where special UAV hubs are constructed.

As a result, the final feasible solution consists of the EV route and several non-overlapping UAV routes which start and end on the EV route. A simple example is shown in Figure 1, where the EV route is $\{0-1-3-5-6-0\}$ while the UAV route is $\{1-2-3, 3-4-6\}$.

We assume that the electric vehicle and the UAV share their electricity, that is, there is a limited capacity battery on the EV which can be used by both the EV and the UAV. For simplicity, we further assume that the UAV could be charged to full capacity instantly. This assumption represents, for instance, that multiple UAVs are loaded on the EV (although only one is in flight at any time) and can be charged, so that a fresh UAV can launch immediately after another UAV is retrieved. This assumption could be relaxed by adding an additional constraint separating launch nodes in the formulation appearing below.

Since both vehicles share their electricity, if the UAV is launched from the EV at a customer node, the electricity required for the UAV route is deducted from the remaining battery level of the EV. To explain this, denote b_i^a as the remaining electricity of EV upon arrival of node i and b_i^d as the remaining electricity of EV upon departure of node i. Using the network in Figure 2, assume the electric vehicle launches the UAV at node 1, travels to node 3 and retrieves the UAV at node 4 while the UAV serves the customer 2. If $b_1^a = 100$, which indicates that EV's remaining battery level upon arrival at node 1 is 100, then EV's battery level upon departure of node 1 could be calculated as $b_1^d = 100 - d_{124} = 100 - 20 = 80$, where d_{124} represents the required electricity of the UAV to be launched at node 1, serves customer 2 and returns to node 4. The battery level of the other nodes are also shown in Figure 2. Since the battery level of EV should be non-negative along its route, the EV may need to visit charging stations to refresh its battery when necessary. It is worth noting that in EVTSP-D, the final solution might contain more than one visit to some specific charging station nodes, while some other charging station nodes might never be visited at all. To permit multiple (and possibly zero) visits to the charging station nodes, while requiring exactly one visit to the customer nodes, graph G is augmented to create $G_0 = (V_0, E_0)$ with a set of s' dummy vertices, $\Phi = \{v_{c+s+1}, v_{c+s+2}, ..., v_{c+s+s'}\}$, one for each potential visit to a charging station node. Denote V' as the augmented vertices set and $V' = V \cup \Phi$. The number of dummy vertices associated with each charging station, n_s, is set to the number of times the associated $v_s \in S$ can be visited. n_s should be set as small as possible so as to reduce the augmented network size, but large enough to not restrict multiple beneficial visits.

Additional notation used in formulating the EVTSP-D is defined next.

A. Sets

I	:	Set of all customers in the problem, $I = \{v_1, v_2, ..., v_c\}$ and $\lvert I\rvert = c$
I'	:	Subset of customers that are available to UAV delivery service, $I' \subset I$
S	:	Set of all charging stations in the original network, $S = \{v_{c+1}, v_{c+2}, ..., v_{c+s}\}$ and $\lvert S\rvert = s$
S'	:	Augmented set of all charging stations, including the m copies of set S. $S' = \{v_{c+1}, .., v_{c+s}, v_{c+s+1}, ..., v_{c+ms}\}$ and $\lvert S'\rvert = ms$. Note that in the augmented network each node can be visited at most once
N	:	Set of all nodes in the augmented network, $N = S' \cup C \cup \{0, c+ms+1\}$, where 0 and $c+ms+1$ both represent the depot and $\lvert N\rvert = c+ms+2$
N_0	:	Set of nodes from which a vehicle may depart in the augmented network. $N_0 = \{v_0, v_1, ..., v_c\} \cup S'$
N_+	:	Set of nodes to which a vehicle may arrive in the augmented network. $N_+ = \{v_1, v_2, ...v_c\} \cup S' \cup \{v_{c+ms+1}\}$
N'	:	Set of all customer nodes and charging station nodes in the augmented network. $N' = I \cup S'$
D	:	Set of tuples of the UAV's feasible route $\langle i, j, k\rangle$, where the UAV is launched from node i, travels to node j and returns to node k. $D = \{\langle i, j, k\rangle : i \in N_0, j \in C', k \in N_+, i \neq j, j \neq k, i \neq k, d_{ij} + d_{jk} \leq Q_d\}$, where Q_d represents the operational time limit of the UAV and d_{ij} represents the UAV's travel time cost from node i to node j.

B. Parameters

τ_{ij}	:	Travel time cost for the EV to travel from node i to node j
d_{ij}	:	Travel time cost for the UAV to travel from node i to node j
d_{ijk}	:	Travel time cost for the UAV to launch from node i, serves node j and return to node k
Q_d	:	Operational time limit of the UAV, which is measured in time units
Q	:	Driving time limit of the EV, which is measured in time units
S_L	:	Time needed to launch the UAV
S_R	:	Time needed to retrieve the UAV
M	:	A positive large number which is an upper bound on total travel time

C. Decision variables:

$x_{ij} \in \{0, 1\}$	:	Equals one if the EV travels from node i to node j and zero otherwise, where $i \neq j$ and $i \in N_0, j \in N_+$
$y_{ijk} \in \{0, 1\}$	:	Equals one if the UAV is launched from node i, travels to node j and returns to the EV at node k and zero otherwise
$p_{ij} \in \{0, 1\}$	:	Equals one if customer node i is visited before customer j in the EV's path and zero otherwise

u_i	:	Position of node i in the EV's path
b_i^a	:	Remaining battery charge of the EV upon arrival at node i, which is measured in time units
b_i^d	:	Remaining battery charge of the EV upon departure from node i, which is measured in time units
t_j	:	Time when the EV arrives at node j
$t_j^{'}$	:	Time when the UAV arrives at node j

D. Mathematical Formulation

Objective :

$$\min \quad t_{c+ms+1} \tag{1}$$

The objective is to minimize the time when both EV and UAV return to the depot after serving all the customers. There are a large number of constraints in the problem, which are introduced below and interspersed with descriptions.

Routing Constraints:

$$\sum_{\substack{i\in N_0\\ i\neq j}} x_{ij} + \sum_{\substack{i\in N_0\\ i\neq j}} \sum_{\substack{k\in N_+\\ \langle i,j,k\rangle\in D}} y_{ijk} = 1 \qquad \forall j \in I \tag{2}$$

$$\sum_{j\in N_+} x_{0j} = 1 \tag{3}$$

$$\sum_{i\in N_0} x_{i,c+ms+1} = 1 \tag{4}$$

$$u_i - u_j + 1 \leq (c+ms+2)(1-x_{ij}) \qquad \forall i \in N^{'}, j \in N_+, j \neq i \tag{5}$$

$$\sum_{\substack{i\in N_0\\ i\neq j}} x_{ij} = \sum_{\substack{k\in N_+\\ k\neq j}} x_{jk} \qquad \forall j \in N^{'} \tag{6}$$

$$\sum_{j\in C^{'}} \sum_{\substack{k\in N_+\\ \langle i,j,k\rangle\in D}} y_{ijk} \leq 1 \qquad \forall i \in N_0 \tag{7}$$

$$\sum_{i\in N_0} \sum_{\substack{j\in C^{'}\\ \langle i,j,k\rangle\in D}} y_{ijk} \leq 1 \qquad \forall k \in N_+ \tag{8}$$

$$2y_{ijk} \leq \sum_{\substack{h\in N_0\\ h\neq i}} x_{hi} + \sum_{\substack{l\in N_0\\ l\neq k}} x_{lk} \qquad \forall i \in N_0, j \in C^{'}, k \in N_+, \langle i,j,k\rangle \in D \tag{9}$$

$$y_{0jk} \leq \sum_{\substack{h\in N_0\\ h\neq k}} x_{hk} \qquad \forall j \in C^{'}, k \in N_+, \langle 0,j,k\rangle \in D \tag{10}$$

$$1-(c+ms+2)\left(1-\sum_{\substack{j\in C^{'}\\ \langle i,j,k\rangle\in D}} y_{ijk}\right) \leq u_k - u_i \qquad \forall i \in N_0, k \in N_+, k \neq i \tag{11}$$

Constraints (2)–(11) are associated with the routing of the two vehicles. In particular, constraint (2) guarantees that each customer node is visited once by either the EV or UAV. Constraints (3) and (4) state that the EV must start from and return to the depot. Constraint (5) is a sub-tour elimination constraint for the EV. Constraint (6) indicates that if the EV visits node j then it must also depart from node j. Constraints (7) and (8) state that each node can launch or retrieve the UAV at most once. Constraint (9) ensures that if there exists a UAV route $\langle i, j, k\rangle$, then EV should travel between i and k. Constraint (10) states that if the UAV is launched from the depot and returned to node k, then node k should be visited by the EV. Constraint (11) is a sub-tour elimination constraint for the UAV.

Battery Constraints:

$$b_j^a \le b_i^d - \tau_{ij} x_{ij} + M(1 - x_{ij}) \qquad \forall i \in N_0, j \in N_+, i \neq j \tag{12}$$

$$b_0^a = Q \tag{13}$$

$$b_i^d = Q - \sum_{\substack{j \in C' \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i,j,k\rangle \in D}} y_{ijk} d_{ijk} \qquad \forall i \in S' \cup \{0, c + ms + 1\} \tag{14}$$

$$b_i^d = b_i^a - \sum_{\substack{j \in C' \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i,j,k\rangle \in D}} y_{ijk} d_{ijk} \qquad \forall i \in I \tag{15}$$

$$b_i^a \ge 0 \qquad \forall i \in N \tag{16}$$

$$b_i^d \ge 0 \qquad \forall i \in N \tag{17}$$

Constraints (12)–(17) are associated with the battery electricity level. In particular, constraint (12) states that if the EV travels from node i to node j, then the electricity level before arriving at node j is τ_{ij} less than the electricity level after leaving node i, regardless whether node i, j are customer nodes or charging stations. Constraint (13) ensures that when EV departs from the depot it is fully charged. Constraint (14) states that if the EV departs from a charging station node i and there is a UAV route that starts at node i, then when EV departs from node i it is no longer fully charged and the UAV route electricity consumption should be deducted from full-charged battery. Constraint (15) states the same situation as constraint (14) except when node i is a customer. Constraint (16) and (17) ensures that the remaining battery charge should be non-negative.

Coordination Constraints:

$$t_i^{'} \geq t_i - M\left(1 - \sum_{\substack{j \in C' \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall i \in N_0 \tag{18}$$

$$t_i^{'} \leq t_i + M\left(1 - \sum_{\substack{j \in C' \\ j \neq i}} \sum_{\substack{k \in N_+ \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall i \in N_0 \tag{19}$$

$$t_k^{'} \geq t_k + M\left(1 - \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C' \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall k \in N_+ \tag{20}$$

$$t_k^{'} \leq t_k - M\left(1 - \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C' \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall k \in N_+ \tag{21}$$

$$t_k \geq t_h + \tau_{hk} + S_L \sum_{\substack{l \in C' \\ l \neq k}} \sum_{\substack{m \in N_+ \\ \langle k,l,m \rangle \in D}} y_{klm} + S_R \sum_{\substack{i \in N_0 \\ i \neq k}} \sum_{\substack{j \in C' \\ \langle i,j,k \rangle \in D}} y_{ijk} - M(1 - x_{hk}) \qquad \forall h \in N_0, k \neq h \tag{22}$$

$$t_j^{'} \geq t_i^{'} + d_{ij} - M\left(1 - \sum_{\substack{k \in N_+ \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall j \in C^{'}, i \in N_0, i \neq j \tag{23}$$

$$t_k^{'} \geq t_j^{'} + d_{ij} + S_R - M\left(1 - \sum_{\substack{i \in N_0 \\ \langle i,j,k \rangle \in D}} y_{ijk}\right) \qquad \forall j \in C^{'}, k \in N_+, k \neq j \tag{24}$$

Constraints (18)–(24) are associated with travel time of the two vehicles. In particular, constraints (18)–(21) ensure that the travel time and UAV range limit are correctly handled. Constraint (22) indicates that if the EV travels from node h to node k where $h \in N_0, k \in N_+$, its arrival time at node k must incorporate the its arrival time at node h, travel time from node h to node k, the UAV's launch time at node h and retrieve time at node k. This constraint is not binding if the electric vehicle does not travel from node h to node k. Constraints (23) and (24) are associated with the UAV's arrival time. Suppose there is a UAV route of $\langle i, j, k \rangle$, then the UAV's arrival time of node j and node k should be related to the UAV's travel time between i and j, j and k and the UAV's retrieve time at node k.

Ordering Constraints:

$$t'_k - t'_j + d_{ij} \le Q_d + M(1 - y_{ijk}) \qquad \forall k \in N_+, j \in C', i \in N_0, \langle i,j,k \rangle \in D \tag{25}$$

$$u_i - u_j \le -1 + (c + ms + 2)(1 - p_{ij}) \qquad \forall i, j \in N', i \ne j \tag{26}$$

$$p_{ij} + p_{ji} = 1 \qquad \forall i, j \in N', i \ne j \tag{27}$$

$$t'_l \ge t'_k - M\left(3 - \sum_{\substack{j \in C' \\ \langle i,j,k \rangle \in D \\ j \ne l}} y_{ijk} - \sum_{\substack{m \in C' \\ m \ne i \\ m \ne k \\ m \ne l}} \sum_{\substack{n \in N_+ \\ \langle l,m,n \rangle \in D \\ n \ne i \\ n \ne k}} y_{lmn} - p_{il}\right) \qquad \forall i, l \in N_0, k \in N_+, i \ne k \ne l, \langle i,j,k \rangle, \langle l,m,n \rangle \in D \tag{28}$$

Constraints (25)–(28) are associated with ordering the two vehicles. Constraint (25) ensures that the UAV route should be within the UAV's flight range. Constraint (26) is a sub-tour elimination constraint and constraint (27) ensures the correct ordering of two different nodes. Constraint (28) indicates that if there exists two UAV route deliveries $\langle i, j, k \rangle$ and $\langle l, m, n \rangle$ and node i is visited before node l by the EV, then node l must be visited after node k.

Domain Constraints:

$$t_0 = 0 \tag{29}$$

$$p_{0j} = 1 \qquad \forall j \in N_+ \tag{30}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in N_0 \tag{31}$$

$$y_{ijk} \in \{0, 1\} \qquad \forall i \in N_0 \tag{32}$$

$$1 \le u_i \le c + ms + 2 \qquad \forall i \in N_+ \tag{33}$$

$$t_i \ge 0 \qquad \forall i \in N \tag{34}$$

$$t'_i \ge 0 \qquad \forall i \in N \tag{35}$$

$$p_{ij} \in \{0, 1\} \qquad \forall i \in N_0, j \in N_+, j \ne i \tag{36}$$

Constraints (29)–(36) specify the domain of all the decision variables.

IV. Solution methods

There are several complications in EVTSP-D: the electric vehicle driving range limitations, the existence of charging stations that can be visited multiple times or not at all and planning the UAV's route alongside that of the EV. The EVTSP-D computational time via a commercial solver is usually prohibitive even for small instances. In previous research of FSTSP, which did not include electricity constraints, a solution time of two hours was reported to solve a network with ten nodes (Es Yurek and Ozmutlu, 2018). Thus, heuristics designed for EVTSP-D are necessary to solve a problem with practical size. In this section, an iterative three-step decomposition heuristic algorithm is presented to solve the proposed EVTSP-D. The proposed algorithm is a modified version of the solution method presented in Es Yurek and Ozmutlu (2018), which reduces the number of combinations that need to be explored by optimizing the search direction.

The heuristic algorithm works by partitioning the customer node set into two subsets - one contains all the customer nodes served by the EV (also called "EV nodes") and the other one contains the remaining customer nodes (also called "UAV nodes"), The EVTSP-D can be solved by constructing an EV route that serves all the EV nodes and optimizing the schedule of UAV delivery based on the constructed EV route afterwards. If the final integrated route satisfies all the electricity constraints, then a feasible solution is found. Denote the number of EV nodes and UAV nodes as n_E and n_U, respectively, with $n_E + n_U = c$, where c is the number of customers in the network. In this way, for each possible value of n_E and n_U, the original problem is decomposed into three subproblems: an electric vehicle path construction problem, which is an instance of the electric vehicle traveling salesmen problem (EVTSP), a UAV node insertion problem, and final solution feasibility check problem. The final route cost is the route cost of EVTSP solution plus the additional waiting time incurred by adding UAV route into the EV's route.

If all three sub-problems are solved exactly, and all possible EV-UAV nodes partitions are explored, this method can solve EVTSP-D exactly. However, there are two major issues associated with this method: first, the number of different combinations of EV nodes and UAV nodes grows exponentially with the size of the network. Second, both the EVTSP and UAV node insertion problems are NP-hard. To address the second issue, a heuristic algorithm is implemented to solve the EVTSP, so that the computational time of the first sub-problem is not prohibitive as the network grows in size. However, using a heuristic for EVTSP also indicates that even if all the possible combinations are explored, the best-found solution may still be sub-optimal. For the first issue, two different search strategies are explained below.

For small instances, we can enumerate all the different combinations of EV nodes and UAV nodes. Note that the final solution may still be sub-optimal since we use a heuristic to solve the EVTSP subproblem. Since in general the UAV has a higher travel speed than the EV, we want the UAV to serve as many customers as possible. In theory, the UAV could serve all the customers as the EV travels between the charging stations, launching and retrieving the UAV. In this sense, the algorithm starts with searching the scenario of $n_E = 0, n_U = c$ and decreasing n_U gradually. In this paper, the algorithm adopting this searching strategy is named "full iterative search algorithm" (FIS), whose implementation is shown in Algorithm 1. In this algorithm, I is the set of customers in the network and $|I| = c$.

However, in FIS it is not always necessary to explore the whole space of (n_E, n_U).

Theorem 1. *In EVTSP-D, consider two scalar $a, b \in N^+$ and $a < b < c$ where c is the total number of customers in the network. Denote n_E as the number of EV nodes in the integrated route, then the minimal cost of EVTSP among all the possible combinations when $n_E = a$ is not greater than that when $n_E = b$.*

Proof. In EVTSP-D, denote p^*_a as the minimal cost of EVTSP among all the possible combinations when $n_E = a$. Let $comb_a$ denote the set containing all the combinations when $n_E = a$ and $|comb_a| = m = \binom{c}{a}$, $comb^i_a$ represents the ith combination of the set $comb_a$, and f as an EVTSP solution algorithm function that maps from a set of nodes to a real number. So $p^*_a = \min f(comb_a) = \min\{f(comb^1_a), f(comb^2_a), ..., f(comb^m_a)\}$. Similarly, p^*_b is the minimal cost of EVTSP among all the possible combinations when $n_E = b$ and $p^*_b = \min f(comb_b) = \min\{f(comb^1_b), f(comb^2_b), ..., f(comb^n_b)\}$ and $n = \binom{c}{b}$.

For every $comb^i_a \in comb_a$, there always exist a set of combinations $\{comb^1_b, comb^2_b, ...\} \subseteq comb_b$, such that every node in $comb^i_a$ is also included in any combination that is included in this set. Conversely, for every $comb^i_b \in comb_b$, eliminating any $b - a$ nodes in the combination would result in another combination that is included in set $comb_a$. So, $comb^i_a$ always has $b - a$ nodes less than its corresponding combinations in $comb_b$ and no element in $comb_a$ can be "ruled out" by this correspondence. Since every combination in $comb_b$ always have $b - a$ more node than combination in $comb_a$, every EVTSP cost of $comb^i_a$ is not greater than the cost of its corresponding combinations in $comb_b$. As a result, the minimal cost of EVTSP under the algorithm f of all the combinations in $comb_a$ is not greater than that of $comb_b$, regardless of what f is chosen as long as the same f is implemented across all the combinations. □

Based on Theorem 1, if at some point in the search, for a certain value of $n_E = a$, the minimal cost of EVTSP of all combinations with this $n_E = a$ is greater than the best-found solution, then all the remaining combinations with $n_E \geq a$ would have a greater EVTSP cost and thus a greater final delivery cost than the best-found solution (since the optimal waiting time of the second subproblem is non-negative and the final delivery cost is the cost of EVTSP plus the additional waiting time). In this case, the search could terminate immediately with a best solution. For example, for a 11 customer case, suppose currently the algorithm's best found solution has an objective value of 100 and we are searching the scenario of $n_E = 5, n_U = 11 - 5 = 6$. Denote the combination set under this scenarios as $comb_5$. If the minimal traveling time of EVTSP within the set $comb_5$ is no less than the best-known final delivery time 100, then the search could terminate immediately. In the Algorithm 1, this process is implemented in lines 22–26.

Algorithm 1 Heuristic with FIS

Input: Network information
Output: EVTSP-D feasible solution

1: $cost \leftarrow \infty$
2: $route \leftarrow 0$
3: $n_U \leftarrow c$, the number of customers in the network
4: **while** $n_U \neq 0$ **do**
5: Create two empty sets, B and T
6: List all customer nodes combinations with n_U UAV nodes and $c - n_U$ EV nodes, add them to set B
7: **for** each combination $comb \in B$ **do**
8:   Denote the EV nodes set as $comb^i_E$, the set of customers served by the EV
9:   Denote the UAV nodes set as $comb^i_U = I \backslash comb^i_E$, the set of customers served by the UAV
10:   $route^E,\ cost^E \leftarrow$ `ConstructEVPath`$(comb^i_E)$
11:   Add $cost^E$ into set T
12:   **if** $cost^E < cost$ **then**
13:    $t_w,\ route^{new} \leftarrow$ `InsertUAVNodes`$(route^E, comb^i_U)$
14:    $cost^{new} \leftarrow cost^E + t_w$
15:    **if** $cost^{new} < cost$ and `CheckFeasibility`$(route^{new})$ == True **then**
16:     $cost \leftarrow cost^{new}$
17:     $route \leftarrow route^{new}$
18:    **end if**
19:   **end if**
20: **end for**
21: **if** $\min\{T\} \geq cost$ **then**
22:   $n_U \leftarrow 0$
23: **else**
24:   $n_U \leftarrow n_U - 1$
25: **end if**
26: **end while**
27: **return** $cost$, $route$

For large instances where it is computationally prohibitive to search the whole (n_E, n_U) space, we instead search some specific combinations that is likely to provide a "good" feasible in a partial (n_E, n_U) space area. So, the algorithm starts searching with $n_E = \alpha c$ where α is the fraction of total nodes where using αc nodes as EV nodes would "likely" to obtain "good" feasible solutions, based on experience of previous test results. Then, to search different combinations in the specific (n_E, n_U) space, a saving-based method is exploited, where the time-saving of each customer node is calculated as the time saving if the node is eliminated from the EVTSP route that contains all the customer nodes and we can only search a limited number of combinations that yield best savings (line 10 in Algorithm 2). In this paper, this approach is named as the "partial iterative search algorithm" (PIS), whose pseudocode is shown in Algorithm 2.

In the rest of the section, a detailed explanation of the three sub-problems are presented.

1. Solve EVTSP (ConstructEVPath)

This sub-problem corresponds to the *ConstructEVPath* function in the algorithm. In this subproblem, given a EV node set N_E and charging station node set S, the function aims to construct an EV route that visits all the EV nodes in N_E, satisfies the EV driving range constraint while minimizing the total travel time. In this paper, a modified Clarke-Wright savings algorithm (MCWS) is implemented to solve the problem. MCWS is a greedy-based maximum saving algorithm proposed in Erdoğan and Miller-Hooks (2012). To solve EVTSP, an augmented network should be

Algorithm 2 Heuristic with PIS

Input: Network information, time limit T
Output: EVTSP-D feasible solution

$cost \leftarrow \infty$
$route \leftarrow 0$
$route^E, cost^E \leftarrow$ `ConstructEVPath(I)`
Calculate the potential time-saving for every customer node based on $route^E$
Sort node of its time-saving from high to low and stored in a new set $savingNode$
Sort integer set $indexSet = \{i, 0 \leq i \leq c\}$ of their distance to αc from low to high
i = 0
while $cost = \infty$ and time limit is not reached **do**
    $n_U \leftarrow i$th element of $indexSet$
    $B \leftarrow$ a fixed length set that contains nodes combinations with greatest savings with n_U UAV nodes
    for each combination $comb^i \in B$ **do**
        Denote the EV nodes set as $comb^i_E$, the set of customers served by EV
        Denote the UAV nodes set as $comb^i_U = I \backslash comb^i_E$, the set of customers served by UAV
        $route^E,\ cost^E \leftarrow$ `ConstructEVPath`($comb^i_E$)
        $t_w,\ route^{NEW} \leftarrow$ `InsertUAVNodes`($route^E, comb^i_U$)
        if `CheckFeasibility`($route^{NEW}$) == True **then**
            $cost \leftarrow cost^E + t_w$
            $route \leftarrow route^{NEW}$
        end if
    end for
    $i \leftarrow i + 1$
end while
return $cost, route$

constructed to ensure a single charging station could be visited multiple times. The details of MCWS are shown below:

- **Step 1**: Create back-and-forth vehicle tours (v_0, v_i, v_0) for every EV node $v_i \in N_E$. Add each created tour to the *tours* list.
- **Step 2**: Calculate the tour duration for all tours in the *tours* list and check their feasibility. Place all feasible tours in the *feasible tours list* and the remainder in the *infeasible tour list*.
- **Step 3**: For each tour in the *infeasible tour list*, calculate the cost of an charging station insertion between customer vertices v_i and the depot v_0, $c(vi, v0) = \tau(v_i, v_f) + \tau(v_f, v_0) - \tau(v_i, v_0)$ for every charging station $v_f \in S$. For every such tour, insert an charging station with the least insertion cost. If driving range constraints are met after the insertion of an charging station, add the resulting tour to the *feasible tours list*. Otherwise, discard the tour.
- **Step 4**: Compute the savings associated with merging each pair of tours in the *feasible tours list* by identify all vertices that are adjacent to the depot in a tour and create a *savings pair list* (SPL) that includes all possible pairs of these vertices (v_i, v_j) such that v_i and v_j belong to different tours. Compute the savings associated with each pair of vertices in the SPL, $s(v_i, v_j) = \tau(v_0, v_i) + \tau(v_0, v_j) - \tau(v_i, v_j)$. Rank the pairs in the SPL in decreasing order of savings $s(v_i, v_j)$.
- **Step 5**: While SPL is not empty, select and remove the topmost pair of vertices (v_i, v_j) in the SPL and merge their associated tours. For the merged tour, check driving range constraint. If it is satisfied, add the resulting tour to the *feasible tours list*. Otherwise, compute the insertion cost $c(v_i, v_j) = \tau(v_i, v_f) + \tau(v_f, v_j) - \tau(v_i, v_0) - \tau(v_j, v_0)$ for savings pair (v_i, v_j) for every charging station $v_f \in S$. Insert the charging station between v_i and v_j with the least insertion cost to make the resulting tour feasible. If the final tour contains more than one charging station, consider whether it is possible to remove one or more of the charging stations from the tour and remove any redundant charging stations. Add the resulting tour to the *feasible tours list* and return to Step 4. If no tour has been added to the *feasible tours list*, stop.

2. UAV node insertion (InsertUAVNodes)

This sub-problem corresponds to the *InsertUAVNodes* function in the algorithm. This sub-problem aims to insert the UAV node into the EV route that is constructed in the previous step while minimizing the waiting time incurred. The MILP formulation of this subproblem is proposed in Es Yurek and Ozmutlu (2018) and the CPLEX solver is used to solve the sub-problem in this paper. The details of the formulation are shown in the appendix.

3. Final solution feasibility check (CheckFeasibility)

This sub-problem corresponds to the *CheckFeasibility* function in the algorithm and aims to check if the integrated route satisfies all the shared electricity constraints. Although the EV's route and UAV's route both satisfy all the constraints separately, the final solution that incorporates both routes might be infeasible because of the shared electricity assumption. More specifically, the electricity level constraint might be violated when the UAV's routes are incorporated into EV's route and the electricity consumption of UAV's route $\langle i, j, k \rangle$ is deducted from the b_i^d. If the final coordinated route still satisfies the electricity constraint, the route is then stored for future comparison until the time limit is reached. This algorithm is shown in Algorithm 3, where a solution s is represented as an EV route r_E and UAV route r_U.

Algorithm 3 Solution feasibility check

Input: A solution $s = \{r_E, r_U\}$
Output: *True* if solution s is feasible and *False* otherwise

for each node i in r_E **do**
  $j \leftarrow$ next(i) in r_E
  if i is the depot and the first node in r_E **then**
    $b_i^d \leftarrow Q$
  else
    $b_j^a \leftarrow b_i^d - \tau_{ij}$
    if j is the charging station node **then**
      if j is a launch node of sortie <j,m,n> **then**
        $b_j^d \leftarrow Q - d_{jmn}$
      else
        $b_j^d \leftarrow Q$
      end if
    else
      if j is a launch node of sortie <j,m,n> **then**
        $b_j^d \leftarrow b_j^a - d_{jmn}$
      else
        $b_j^d \leftarrow b_j^a$
      end if
    end if
  end if
end for
if minimal value of b_j^a, b_j^d of all nodes in r_E is non-negative **then**
  return *True*
else
  return *False*
end if

V. Numerical Analysis

In numerical analysis, the proposed MILP formulation and iterative search algorithm are compared with FSTSP formulation and neighbourhood search heuristic. Then the proposed algorithm is tested on small randomly generated instances, followed by a real-world case study. In this section, all the cost and compute time are measured in seconds and all experiments are run on a 3.6 GHz Intel Core i7 desktop with 32 GB RAM.

A. MILP formulation comparison

In this section, the MILP formulation proposed in section 3 is compared to the FSTSP formulation proposed in Murray and Chu (2015). EVTSPD is a generalization of FSTSP when the EV has unlimited driving range and there are no charging stations in the network. In this case the only difference between the proposed EVTSPD formulation with that in Murray and Chu (2015) is that the former have extra loose battery constraints and battery decision variables. To analyze the performance of these two formulations, tests on randomly generated instances with different sizes are conducted. For all the generated instances, the single depot is located at $(0, 0)$, and the coordinates of the customer nodes are uniformly distributed between -10 km and 10 km. The truck travels at the speed of 40 km/h while the UAV's speed is 60 km/h. The flight range of the UAV is 25 minutes. The results are shown in Table 1. In the table, $|C|$ indicates the number of customers while the computational time is the average result of at least ten independent runs. Both MILP models are implemented in Pyomo and solved with ILOG's CPLEX Concert Technology solver (version 12.6.3).

Table 1 MILP formulation performance comparison

	Computational Time	
$\|C\|$	FSTSP	EVTSPD
5	1.1	0.9
6	3.6	4.6
7	4.8	12.8
8	33.1	136.1
9	492.3	2057.8
10	>7200	>7200

As can be seen from Table 1, the computation time on instance that contains less than 6 customers of both formulations are very close. However, the gap appears as the instance size increases. For instances with 7, 8 and 9 customers the EVTSPD formulation's computational time is about three times of FSTSP formulation. Based on the CPLEX modelling report, the EVTSPD has about 10% more constraints and decision variables than the FSTSP formulation. For instances containing more than 9 customers both formulation are unable to find optimal result in two hours.

B. Comparison between iterative search method and neighborhood search heuristic

To evaluate and analyze the performance of the iterative search method proposed in this paper, it is compared with previously published variable neighborhood search (VNS) method proposed in De Freitas and Penna (2018) on randomly generated flying sidekick traveling salesman problem instances. The VNS method first creates an initial solution using the optimal TSP solution and saving-based heuristics proposed in Murray and Chu (2015). Then a local search algorithm is performed to obtain the local optimum. If the obtained solution is better than the incumbent one, it is assigned to be the current solution and the search continues. VNS stops when no better neighbours exists for the current solution. Seven different neighbours are defined including reinsertion, two opt, etc.

To compare the iterative search method and VNS on FSTSP instances, randomly generated instances are tested, with the same setting described in the previous subsection. Besides, we set the EV's driving range as unlimited and the number of charging stations as zero. Some of the unnecessary functions are also eliminated from the iterative search algorithm such as the function that checks truck route's feasibility. An TSP solver replaces the MCWS algorithm to get the initial truck's route. Note that by using a TSP solver, all three subproblems are solved optimally and this indicates that the results found by IS is also optimal. Both the iterative search and VNS methods are coded in Python. The test results are shown in Table 2, where the results are the average value of at least 20 independent runs and the column *Std* presents the standard deviation of the cost.

Table 2 IS and VNS performance comparison

	VNS			IS		
$\lvert C\rvert$	Cost	Std	Compute Time (s)	Cost	Std	Compute Time (s)
10	4764.0	822.3	<1	4296.5	734.6	<1
15	6244.0	693.0	<1	5744.0	695.8	4.1
20	6852.0	668.2	<1	6180.5	711.2	22.1
25	7603.5	540.2	<1	7422.5	525.5	30.0
30	8497.5	706.7	<1	8491.5	896.8	53.2

As can be seen from Table 2, the average final route cost of both methods are very close in general, with iterative search method has slightly better results. This is reasonable considering the IS is able to find optimal solution while for VNS optimality is not guaranteed. In terms of computational efficiency the VNS dominates IS in all instances, while for IS a exponential growth of computational time is observed. This comparison illustrates that when the instance size is small, the iterative search method should be preferred over VNS heuristic. Compared to the VNS proposed in De Freitas and Penna (2018), the IS method can easily explore solutions with different drone nodes, while it is relatively difficult for VNS to do this as most of the defined neighbours focus more on the permutations of truck nodes.

C. EVTSPD experiments on small instances

This section compares the performance of the proposed heuristic algorithm on small instances with solving the problem via commercial solver, and illustrates the delivery time improvement of deploying UAV to deliver parcels. The experimental setting is described in the first subsection, while the second one presents the computational results.

1. Experimental Setting

Since both the related problems of EVRP and TSP-D are introduced in recent years, benchmark instances are limited, and most of them are not available publicly. Because of the complexity of the TSP-D and the extra complexity incurred by adding the EV driving range limit constraint, the feasible instance that could be solved by a commercial solver is of relatively small size. According to Es Yurek and Ozmutlu (2018), commercial solvers can only obtain optimal solutions for instances containing up to 10 customers in 2 hours. So in this paper, we conduct the numerical analysis on randomly generated instances. Most of the experimental settings are adopted from previous research Murray and Chu (2015), Es Yurek and Ozmutlu (2018). For all the generated instances, the single depot is located at $(0, 0)$, and the coordinates of the customer nodes and charging station nodes are uniformly distributed between -10 km and 10 km. The EV travels at the speed of 40 km/h while the UAV's speed is 60 km/h. Besides, to test the effect of the EV driving range constraint on the route cost, two parameter setting are tested for the 5-customer case. In the first setting the EV has a driving time limit of 4000 seconds while the UAV has a driving time limit of 1200 seconds, while in the second setting EV's driving range is 5400 seconds and the UAV has a flight range of 1500 seconds. For the 10-customer case, we only test the second setting. The launch/retrieve time and charging time are set to zero in these instances. The distance matrix for the electric vehicle is calculated using the Manhattan metric, while the Euclidean metric is used for the UAV to reflect its greater mobility options. In all the numerical experiments, the charging station set S is replicated twice in the network.

The proposed heuristic algorithm is coded in Python while the MILP formulation is implemented in Pyomo.

2. *Computational Results*

The computational results for random instances are shown in Table 3 and Table 4. In the tables, for the case name of "5C2S2R01", "5C" represents that this instance includes five different customers, "2S" represents that this instance contains two different charging stations, "2R" represents that the augmented network contains two copies of all the charging stations and "01" is the index for a specific type of instance. As a result, in the augmented network of case "5C2S2R01", the actual number of nodes is 11 including the depot. Besides, denote the optimal solution and FIS solution as $cost^{opt}$ and $cost^{FIS}$, respectively, the optimality gap shown in the table is calculated as:

$$\frac{cost^{FIS} - cost^{opt}}{cost^{opt}}.$$

Note that in Table 3 the first 10 cases adopts the first setting (EV range = 4000s, UAV range = 1200s) while the latter ten cases adopts the second one (EV range = 5400s, UAV range = 1500s). In Table 4 all the 10 customer cases have EV range = 5400s, UAV range = 1500s.

As can be seen from Table 3, for 5-customer instances, the average computational time of CPLEX is about 2400 seconds while that of the FIS algorithm is less than one second. On average, the optimality gap of FIS lies between 0.10 and 0.15. As a comparison, Murray and Chu (2015) reports an optimality gap between 0.1 and 0.37 for the saving-based and nearest neighbor heuristics for the instances of FSTSP. Also, note that the proposed algorithm adopts MCWS heuristic to solve the EVTSP subproblem, which has an optimality gap of 0.05 to 0.10 based on the numerical experiments reported in Erdoğan and Miller-Hooks (2012). So, our results indicate a higher optimality gap when a UAV, which shares the electricity with the EV, is added into the integrated route. Besides, compare the performance of FIS under two different settings, it is obvious that when the EV range constraint is loose such that MCWS can find near optimal EV route, the FIS can obtain near optimal result, as in cases 5C2S2R12 and 5C2S2R14 to 5C2S2R17.

Note that even with five customers, the problem is difficult to solve. The EVTSP subproblem is NP-hard; and even in small instances, complete enumeration is difficult because an arbitrary number of charging station visits can be inserted between customer visits, and because charging stations can be visited multiple times between different customers. Therefore, we use a heuristic to solve this subproblem, and optimality cannot be guaranteed for EVTSP for these instances.

For 10-customer instances, CPLEX fails to find the optimal solution within two hours so only the results of iterative search method are reported here. Full iterative search takes about 80 seconds on average, which is greater than the 7 seconds for partial iterative search.On average the obtained route cost of PIS is 15% greater than FIS. It is also noticeable that in some cases PIS might fail to find feasible solution of the problem.

Additional tests are conducted to evaluate the improvement of the final solution with the assistance of UAV delivery. The final solution of EVTSP-D is compared with solving the same instance as the EVTSP. Denote $cost^{EVTSP}$ as the solution cost of EVTSP, the results are shown in Table 2, in which the difference gap is calculated as:

$$\frac{cost^{EVTSP} - cost^{FIS}}{cost^{FIS}}$$

As illustrated, the utilization of UAV delivery would greatly reduce the delivery time. On average, the final delivery time of EVTSP-D is 39.17% lower than EVTSP solution for 5-customer instances and 28.55% lower for 10-customer instances. This improvement indicates that the logistics companies might have a great benefit by utilizing UAV deliveries by reducing delivery times.

Table 3 Computational results for 5-customer cases

Case Name	Computational Time (s) CPLEX	FIS	$cost^{opt}$	$cost^{FIS}$	Optimality gap
5C2S2R01	240	<1	4390	4950	0.127
5C2S2R02	486	<1	4500	4890	0.087
5C2S2R03	2103	<1	4600	5000	0.087
5C2S2R04	2329	<1	2720	2730	0.004
5C2S2R05	5912	<1	4140	4960	0.198
5C2S2R06	259	<1	6460	6650	0.029
5C2S2R07	2775	<1	2930	3760	0.284
5C2S2R08	771	<1	4780	6350	0.328
5C2S2R09	3038	<1	3970	4430	0.116
5C2S2R10	6451	<1	2780	3560	0.280
Average	**2436**	**<1**	**4127**	**4728**	**0.153**
5C2S2R11	1002	<1	3480	5350	0.537
5C2S2R12	826	<1	2620	2620	0
5C2S2R13	5210	<1	2390	2560	0.071
5C2S2R14	3177	<1	2540	2540	0
5C2S2R15	1672	<1	2330	2330	0
5C2S2R16	3235	<1	2450	2450	0
5C2S2R17	3450	<1	2850	2850	0
5C2S2R18	1280	<1	3120	3940	0.263
5C2S2R19	1616	<1	2540	2780	0.094
5C2S2R20	1823	<1	3770	4180	0.109
Average	**2329**	**<1**	**2809**	**3160**	**0.107**

Table 4 Computational results for 10-customer cases

	FIS		PIS		
Case Name	cost	time	cost	time	Gap
10C4S2R01	5010	84	5370	7	0.072
10C4S2R02	2650	133	2830	8	0.068
10C4S2R03	3940	158	5150	15	0.307
10C4S2R04	4550	63	5490	4	0.206
10C4S2R05	4570	75	Inf	2	N/A
10C4S2R06	5500	25	5500	3	0
10C4S2R07	4830	33	6750	2	0.397
10C4S2R08	4940	37	5540	3	0.121
10C4S2R09	4190	53	4340	7	0.036
10C4S2R10	4870	156	5880	19	0.207
Average	**4505**	**81.7**	**5205**	**7**	**0.156**

Table 5 Computational results for final delivery time comparison

Case Name	$cost^{FIS}$	$cost^{EVTSP}$	Difference gap	n_U
5C4S2R21	5320	5680	6.77%	3
5C4S2R22	4510	5930	31.49%	2
5C4S2R23	5000	8000	60.00%	2
5C4S2R24	2030	3690	81.77%	2
5C4S2R25	3560	Inf	N/A	4
5C4S2R26	4520	6010	32.96%	1
5C4S2R27	3590	4790	33.43%	4
5C4S2R28	4360	6900	58.26%	2
5C4S2R29	4130	4800	16.22%	3
5C4S2R30	3100	4080	31.61%	4
Average	**4012**	**5542**	**39.17%**	**2.7**
10C5S2R11	5320	7920	48.87%	3
10C5S2R12	6400	7240	13.13%	7
10C5S2R13	6490	8150	25.58%	2
10C5S2R14	5730	7060	23.21%	7
10C5S2R15	5690	6990	22.85%	3
10C5S2R16	7250	9100	25.52%	4
10C5S2R17	5270	6350	20.49%	3
10C5S2R18	5440	7330	34.74%	3
10C5S2R19	5410	8020	48.24%	7
10C5S2R20	5860	7200	22.87%	7
Average	**5886**	**7536**	**28.55%**	**4.6**

D. Real-world case study

A real-world case study is conducted to illustrate the effectiveness of the proposed algorithm in solving EVTSP-D with practical size. This study also analyzes the effects of several key parameters on the final delivery time. In this case study, the downtown Austin network is analyzed, which contains one depot, 25 customer nodes, and 10 charging station nodes. The travel time of the EV from one node to another node is estimated using the Google map Python API for the peak hour of a typical Monday. Thus, the resulting travel time matrix is asymmetric. In the default setting, the travel time of the UAV from one node to another is assumed to be half of that of the EV. We assume that all customers could be visited by either the EV or the UAV. The UAV has a maximum operation time of 30 minutes, and the EV has a driving limit of 2 hours without charging. The EV adopts a full-charge policy at each charging station, and its battery could be refreshed immediately (by replaced with a fully-charged battery). All the charging stations could be visited multiple times if necessary. There is no time window constraint associated with each customer node, so the service time is also ignored. The charging station set S is replicated twice in the network and the value of α is chosen as 0.4, based on the observation in the numerical experiment that most of the heuristic solutions has $0.4c$ nodes as EV nodes. The PIS is adopted to solve the problem and the computational time limit is 2 minutes considering the problem size.

The final integrated route of the case study is shown in Figure 3. The completion delivery time is 12745 seconds (approximately 3.54 hours), and the final route consists of 15 customers being served by the UAV and ten customers being served by the EV.

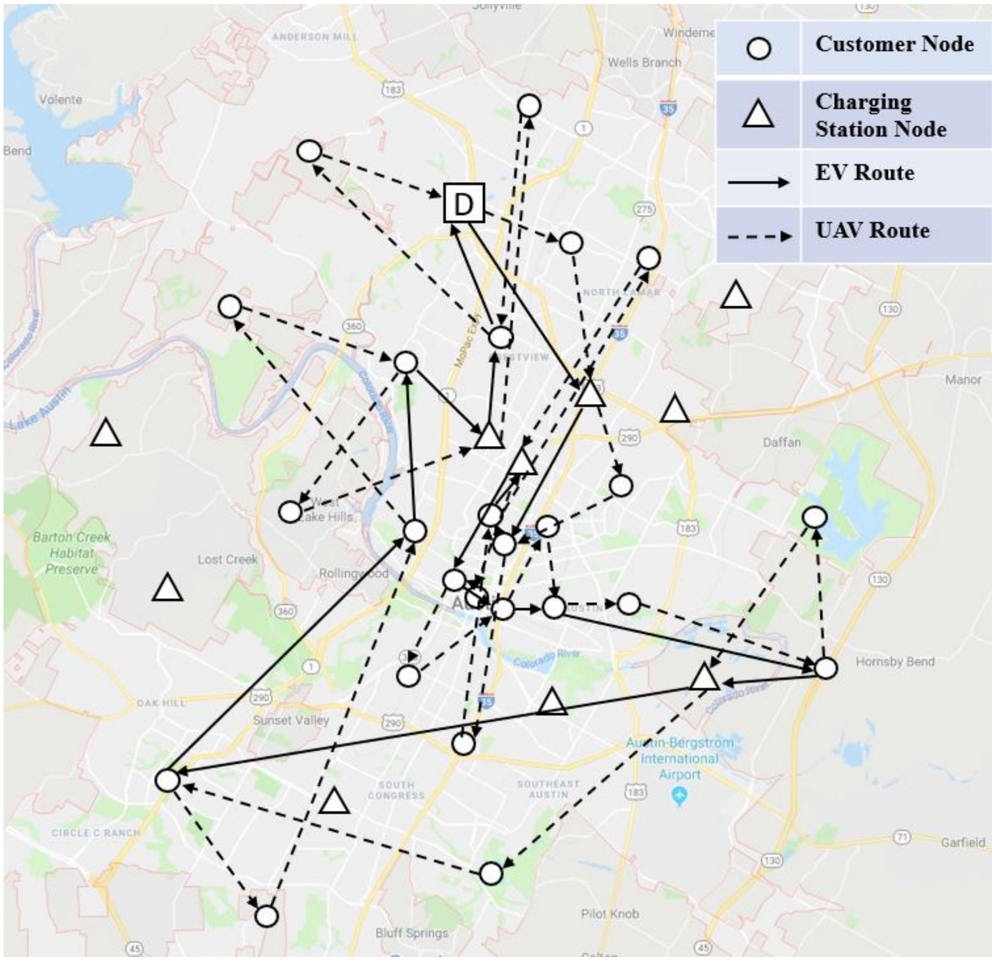

Figure 3 A heuristic solution route for the real-world case study

In the rest of this section, we study the impact of characteristics of different parameters on the performance of this EV-UAV delivery system, from which we will gain insights into this new mode of transportation. All the results are the best-found solution based on PIS with computational time of one hour. Throughout this section, denote $cost$ as the final delivery time, n_U as the number of customers served by the UAV, $cost^{EV}$ as the travel time of electric vehicle measured in seconds and t_w as the total waiting time during the delivery process.

1. UAV speed

The UAV is typically faster than the EV as it can take shortcuts without being affected by ground traffic congestion. The X-Wing project initiated by Google declares their delivery UAVs can reach the speed of 120 km/h when tested in a suburban area in Australia. This section explores the effect of the relative speed ratio of UAV to EV on the final delivery cost. Denote the relative travel speed of UAV and EV as β, Table 6 provides the results for five different relative speeds of UAV from β = 1 to β = 3. As expected, the delivery completion time decreases as the relative speed of UAV increases. When the UAV has the same speed as EV (β = 1) or The UAV has slightly higher speed than the EV (β = 1.5), no feasible route is found within one hour. As the UAV speed increases, the total delivery time, the EV travel time and the total waiting time reduces gradually. In particular, when the relative speed increases from β = 1 to β = 3, the resulting waiting time decreases by 86% (from 1295 seconds to 180 seconds). This result indicates that increasing the UAV speed might be an effective approach to reduce the risk and operational cost incurred by the waiting period during the delivery.

Table 6 Effect of UAV speed on EVTSP-D solution

	$cost$	n_U	$cost^{EV}$	t_w
$\beta = 1$	Infeasible	Infeasible	Infeasible	Infeasible
$\beta = 1.5$	Infeasible	Infeasible	Infeasible	Infeasible
$\beta = 2$	10865	15	9570	1295
$\beta = 2.5$	9420	15	8640	780
$\beta = 3$	8740	15	8560	180

2. UAV's flight range

A UAV's flight range is another factor that might influence the delivery completion time as a wider flying range enables delivering parcels to customers that are located further and thus increases the number of customers that could be served by the UAV. Research on how the UAV's flight range affects the TSPD solution and time-saving in a real-world case study is limited. This section fills this gap by exploring the effect of the UAV flight range on the final feasible solution in the Austin network. Denote the UAV's flight range as Q_d which is measured in minutes; Table 7 presents the characteristics of the final solution with different Q_d value.

As shown in Table 7, contrary to the expectation, the delivery completion time of the feasible solution found within one hour increases with the UAV flight range. For example, the final delivery time found within one hour when the UAV flight range is 15 minutes is 10745 seconds while that for the UAV with 60 minutes operation limit is 11075 seconds. We think that this is because as the UAV flight range increases, the number of feasible branches that need to be examined increases, since greater UAV operation limit indicates more customer nodes could be served by the UAV. In a certain amount of time, the number of combinations that is explored in the algorithm decreases, and we might end up with a "bad" solution.

Theoretically, given enough computational time, the best found solution with a greater UAV flight range would not be worse than that with low flight range. Additional tests are conducted to demonstrate this statement. Since the feasible solution found when $Q_d = c$ is also feasible when when $Q_d > c$, one can start the searching procedure with the previously found feasible solution and check if the algorithm could find a better solution given a "good" starting point. The results are shown in Table 5. In the table, the solution when $Q_d = 30$ uses the solution of $Q_d = 15$ as the starting point. Similarly, the solution when $Q_d = 45$ uses the solution of $Q_d = 30$ as the starting point. The computational time is one hour. Based on the results, when Q_d increases from 15 minutes to 30 minutes a better solution is found. However, the algorithm seems to stuck in a minimal point when Q_d increases from 30 minutes to 60 minutes. The result illustrates that given a initial solution which has tighter bounds and relaxed constraint (in this case, UAV's flight range), the algorithm might obtain a better (or at least no-worse) solution.

Table 7 Effect of UAV flight range on EVTSP-D solution

	$cost(s)$	n_U	$cost^{EV}(s)$	$t_w(s)$
$Q_d = 15$	10745	15	8760	1985
$Q_d = 30$	10865	15	9570	1295
$Q_d = 45$	10980	15	9570	1410
$Q_d = 60$	11075	15	9570	1505

Table 8 Final results when using previously found solution as initial point

	$cost(s)$	n_U	$cost^{EV}(s)$	$t_w(s)$
$Q_d = 15$	10745	15	8760	1985
$Q_d = 30$	9420	15	8640	780
$Q_d = 45$	9420	15	8640	780
$Q_d = 60$	9420	15	8640	780

3. EV's driving range

The driving range of EV depends on the size of the battery it carries. However, as battery size increases, the operational and maintenance cost also grows, so logistics companies deploying electric vehicles must make a trade-off between these two factors. In EVTSP-D, the driving range of the EV affects the final delivery completion time in that low driving range indicates EV has to visit charging stations frequently. Tighter electricity constraints also makes it difficult for the heuristic algorithm to find a "good" feasible solution. Denote the EV's driving range as Q which is measured in hours, this section presents the result of how EV's driving range affects the solution route cost, as illustrated in Table 9. As presented, the final delivery time decreases with the EV driving range increases. When the EV driving range is only 1.5 hours, it has to visit the charging station frequently to refresh the battery which results in a long waiting time in the route. Note that in one hour, only one feasible route is found by the heuristic algorithm when $Q = 1.5$ hour, which supports the intuition that when electricity constraint is tight, it is difficult to even find a feasible solution.

As in previous section, additional test are conducted to check if the algorithm could find a better solution given a better initial searching point. The results are shown in Table 10. As presented, the final delivery time decreases as we relax the EV driving range constraint and use a better initial point, which is different from the effect of UAV flight range where the solution is stuck at a local point.

Table 9 Effect of EV Driving Range on EVTSP-D solution

	$cost(s)$	n_U	$cost^{EV}(s)$	$t_w(s)$
$Q = 1.5$	12255	15	8570	3685
$Q = 2$	10865	15	9570	1295
$Q = 3$	10610	15	9530	1080

Table 10 Final results when using previously found solution as initial point

	$cost(s)$	n_U	$cost^{EV}(s)$	$t_w(s)$
$Q = 1.5$	12255	15	8570	3685
$Q = 2$	9420	15	8640	780
$Q = 3$	9140	15	8400	740

VI. Conclusion

In this paper, the mixed integer-linear programming formulation of EVTSP-D is presented, and an efficient iterative search heuristic is proposed. EVTSPD aims to find a coordinated EV-UAV route that minimizes total travel time and serves a set of customers while incorporating stops at charging stations in route plans to ensure sufficient charge. The proposed formulation and solution algorithm are compared with TSPD to evaluate their performance. The results shows that the formulation and iterative search method are efficient when instance size is small. The case study on randomly generated instances indicates that the proposed algorithm performs well compared to exact solution methods, with significantly less computational time. besides, the real-world case study demonstrates that the proposed heuristic can solve the problem with practical size within a reasonable computational time. It also conducts several sensitivity analyses to illustrate how some key parameters affect the final integrated route. The test results demonstrate that UAV speed has a major influence on the final delivery time, compared to EV driving range and UAV flight range. The EVTSP-D formulation, along with these solution techniques, will aid organizations with EV fleets in overcoming difficulties that exist as a result of limited recharging infrastructure. The new delivery concept of using UAV and EV to perform last-mile delivery would result in financial and environmental benefits when considering the reduced operation cost of fueling and switching to UAV, which does not require a costly human pilot. Besides, the research in the newly-merged approach will provide intuition for the future development of a more sophisticated implementation of delivery service.

There remain several practical challenges for UAV delivery, regarding payload capacity, safety, and public acceptance(Watkins et al., 2020). It also remains to be seen whether battery-swapping stations (as the type we assume) or fast-charging stations ultimately make more economic sense for logistics fleets. As the number of customers increases, it will become important to consider a multi-vehicle version of the current problem, perhaps with heterogeneous EV and UAV range and capacity, or with alternative (non full-charge) policies. All of these should be addressed in future research.

VII. Acknowledgements

This research is based on work supported by the National Science Foundation under Grant No. 1826230, 1562109/1562291, 1562109, 1826337, 1636154 and 1254921. This work is also supported by the Center for Advanced Multimodal Mobility Solutions and Education (CAMMSE).

A. MIP formulation of UAV node insertion

This section introduces the mixed integer programming model of the UAV node insertion function in the iterative search algorithm. This method takes the a fixed EV's route and a set of selected nodes as the input and seeks to serve these nodes by UAV and insert these sorties into the EV's route. This formulation is originally proposed in Es Yurek and Ozmutlu (2018). The definition of the indexes, sets, decision variables and parameters are described below:

A. Indexes

i, j	:	Node
k	:	Position
p	:	UAV tour

B. Sets

C	:	Set of all customer nodes
D	:	Set of customers that would be served by the UAV
S	:	Set of all potential UAV sorties

C. Parameters

d_s	:	Duration of sortie s
f_{is}	:	Binary parameter which equals 1 if sortie s starts from node i, and 0 otherwise
a_{is}	:	Binary parameter which equals 1 if sortie s serves node i, and 0 otherwise
l_{is}	:	Binary parameter which equals 1 if sortie s ends at node i, and 0 otherwise
N	:	Number of customer nodes in the instance
t_i	:	Arrival time of the truck at node i
m_k	:	Node assigned to position k in the truck's route
r_t	:	Current truck's route
$\{0, C+1\}$	:	The depot set

D. Decision variables

x_s	:	Binary parameter which equals 1 if sortie s is chosen in the final solution, and 0 otherwise
w_i	:	Waiting time of the truck at node i

Objective :

$$\min \sum_{j \in C \setminus D \cup \{C+1\}} \Big(w_i \tag{1}$$

The objective is to minimize the total waiting time that is incurred by adding UAV sorties to the current truck's route.

s.t.

$$\sum_{s\in S}\{d_s l_{is} x_s - (t_i l_{is} x_s - \sum_{j\in C\setminus D\cup\{0\}} t_j f_{js} x_s)\} \leq w_i \qquad \forall i \in C \setminus D \cup \{C+1\} \tag{2}$$

$$\sum_{s\in S} a_{is} x_s = 1 \qquad \forall i \in D \tag{3}$$

$$\sum_{s\in S} f_{is} x_s \leq 1 \qquad \forall i \in C \setminus D \cup \{0\} \tag{4}$$

$$\sum_{s\in S} l_{is} x_s \leq 1 \qquad \forall i \in C \setminus D \cup \{C+1\} \tag{5}$$

$$\sum_{s\in S} f_{m_k s} x_s + \sum_{s\in S} l_{m_k s} x_s - 2\left(1 - \sum_{s\in S} f_{m_i s} l_{m_j s} x_s\right) \leq 0 \qquad \forall i = 0, 1, ..n-1, j = i+2, ..., C+1, i \leq k \leq j, i \neq j \tag{6}$$

$$x_s \in \{0, 1\} \qquad s \in S \tag{7}$$

$$w_i \geq 0 \qquad i \in C \setminus D \cup \{C+1\} \tag{8}$$

Constraint (2) specifies that when the truck waits for the UAV then the waiting time is incurred. Constraint (3), (4) and (5) indicate that every node can only be launch node, UAV node and retrieve node for at most one sortie, respectively. Constraint (6) indicates that the UAV cannot be re-launched before it is retrieved. Constraint (7) and (8) specifies the domain of the decision variables.

References

United States Environmental Protection Agency (EPA), "Sources of Greenhouse Gas Emissions," , 2018. URL `https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions`.

FedEx, "FedEx Acquires 1,000 Chanje Electric Vehicles," , 2018. URL `https://about.van.fedex.com/newsroom/fedex-acquires-1000-chanje-electric-vehicles/`.

Sahin, B., Yilmaz, H., Ust, Y., Guneri, A. F., and Gulsun, B., "An approach for analysing transportation costs and a case study," *European Journal of Operational Research*, 2009. doi:10.1016/j.ejor.2007.10.030.

Howey, D. A., Martinez-Botas, R., Cussons, B., and Lytton, L., "Comparative measurements of the energy consumption of 51 electric, hybrid and internal combustion engine vehicles," *Transportation Research Part D: Transport and Environment*, Vol. 16, No. 6, 2011, pp. 459–464.

Ma, H., Balthasar, F., Tait, N., Riera-Palou, X., and Harrison, A., "A new comparison between the life cycle greenhouse gas emissions of battery electric vehicles and internal combustion vehicles," *Energy policy*, Vol. 44, 2012, pp. 160–173.

DroneZon, "Drones For Deliveries From Medicine To Post, Packages And Pizza," , 2019. URL `https://www.dronezon.com/drones-for-good/drone-parcel-pizza-delivery-service/`.

Teal Group, "Teal Group Predicts Worldwide Civil Drone Production Will Soar Over the Next Decade," , 2018. URL `https://www.tealgroup.com/index.php/pages/press-releases/54-teal-group-predicts-worldwide-civil-UAV-production-will-soar-over-the-next-decade`.

Chauhan, D., Unnikrishnan, A., and Figliozzi, M., "Maximum coverage capacitated facility location problem with range constrained drones," *Transportation Research Part C: Emerging Technologies*, Vol. 99, 2019, pp. 1–18.

Hong, I., Kuby, M., and Murray, A. T., "A range-restricted recharging station coverage model for drone delivery service planning," *Transportation Research Part C: Emerging Technologies*, Vol. 90, 2018, pp. 198–212.

Hoareau, G., Liebenberg, J. J., Musial, J. G., and Whitman, T. R., "Distributed, unmanned aerial vehicle package transport network," , Sep. 12 2017. US Patent 9,760,087.

Pelletier, S., Jabali, O., Laporte, G., and Veneroni, M., "Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models," *Transportation Research Part B: Methodological*, 2017. doi:10.1016/j.trb.2017.01.020.

De Cauwer, C., Van Mierlo, J., and Coosemans, T., "Energy consumption prediction for electric vehicles based on real-world data," *Energies*, 2015. doi:10.3390/en8088573.

Murray, C. C., and Chu, A. G., "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, 2015. doi:10.1016/j.trc.2015.03.005.

Dantzig, G. B., and Ramser, J. H., "The truck dispatching problem," *Management Science*, 1959. doi:10.1287/mnsc.6.1.80.

Baldacci, R., Mingozzi, A., and Roberti, R., "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, Vol. 218, No. 1, 2012, pp. 1–6.

Laporte, G., "Fifty years of vehicle routing," *Transportation science*, Vol. 43, No. 4, 2009, pp. 408–416.

Desaulniers, G., Desrosiers, J., and Spoorendonk, S., "The Vehicle Routing Problem with Time Windows: State-of-the-Art Exact Solution Methods," *Wiley encyclopedia of operations research and management science*, 2010.

Osman, I. H., "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research*, 1993. doi:10.1007/BF02023004.

Gendreau, M., Hertz, A., and Laporte, G., "New insertion and post-optimization procedures for the traveling salesman problem," *Operations Research*, 1992. doi:10.1287/opre.40.6.1086.

Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I., "The vehicle routing problem: State of the art classification and review," *Computers & Industrial Engineering*, Vol. 99, 2016, pp. 300–313.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N., "A literature review on the vehicle routing problem with multiple depots," *Computers & Industrial Engineering*, Vol. 79, 2015, pp. 115–129.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L., "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, Vol. 225, No. 1, 2013, pp. 1–11.

Toth, P., and Vigo, D., *Vehicle routing: problems, methods, and applications*, SIAM, 2014.

Eksioglu, B., Vural, A. V., and Reisman, A., "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, Vol. 57, No. 4, 2009, pp. 1472–1483.

Golden, B. L., Raghavan, S., and Wasil, E. A., *The vehicle routing problem: latest advances and new challenges*, Vol. 43, Springer Science & Business Media, 2008.

Toth, P., and Vigo, D., *The vehicle routing problem*, SIAM, 2002.

Erdoğan, S., and Miller-Hooks, E., "A green vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, 2012. doi:10.1016/j.tre.2011.08.001.

Conrad, R. G., and Figliozzi, M. A., "The recharging vehicle routing problem," *Proceedings of the 2011 industrial engineering research conference*, IISE Norcross, GA, 2011, p. 8.

Schneider, M., Stenger, A., and Goeke, D., "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, Vol. 48, No. 4, 2014, pp. 500–520.

Montoya, A., Guéret, C., Mendoza, J. E., and Villegas, J. G., "The electric vehicle routing problem with nonlinear charging function," *Transportation Research Part B: Methodological*, 2017. doi:10.1016/j.trb.2017.02.004.

Koç, Ç., and Karaoglan, I., "The green vehicle routing problem: A heuristic based exact solution approach," *Applied Soft Computing*, Vol. 39, 2016, pp. 154–164.

Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. F., "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research*, Vol. 252, No. 3, 2016, pp. 995–1018.

Masmoudi, M. A., Hosny, M., Demir, E., Genikomsakis, K. N., and Cheikhrouhou, N., "The dial-a-ride problem with electric vehicles and battery swapping stations," *Transportation research part E: logistics and transportation review*, Vol. 118, 2018, pp. 392–420.

Keskin, M., and Çatay, B., "Partial recharge strategies for the electric vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, Vol. 65, 2016, pp. 111–127.

Goeke, D., and Schneider, M., "Routing a mixed fleet of electric and conventional vehicles," *European Journal of Operational Research*, Vol. 245, No. 1, 2015, pp. 81–99.

Felipe, Á., Ortuño, M. T., Righini, G., and Tirado, G., "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 71, 2014, pp. 111–128.

Erdelić, T., and Carić, T., "A survey on the electric vehicle routing problem: variants and solution approaches," *Journal of Advanced Transportation*, Vol. 2019, 2019.

Doppstadt, C., Koberstein, A., and Vigo, D., "The hybrid electric vehicle–traveling salesman problem," *European Journal of Operational Research*, Vol. 253, No. 3, 2016, pp. 825–842.

Doppstadt, C., Koberstein, A., and Vigo, D., "The Hybrid Electric Vehicle-Traveling Salesman Problem with Time Windows," *European Journal of Operational Research*, 2019.

Liao, C.-S., Lu, S.-H., and Shen, Z.-J. M., "The electric vehicle touring problem," *Transportation Research Part B: Methodological*, Vol. 86, 2016, pp. 163–180.

Roberti, R., and Wen, M., "The electric traveling salesman problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 89, 2016, pp. 32–52.

Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E., "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, Vol. 72, No. 4, 2018, pp. 411–458.

Mbiadou Saleu, R. G., Deroussi, L., Feillet, D., Grangeon, N., and Quilliot, A., "An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem," *Networks*, Vol. 72, No. 4, 2018, pp. 459–474.

Agatz, N., Bouman, P., and Schmidt, M., "Optimization approaches for the traveling salesman problem with drone," *Transportation Science*, 2018. doi:10.1287/trsc.2017.0791.

Ha, Q. M., Deville, Y., Pham, Q. D., and Hà, M. H., "On the min-cost traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, 2018. doi:10.1016/j.trc.2017.11.015.

Es Yurek, E., and Ozmutlu, H. C., "A decomposition-based iterative optimization algorithm for traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, 2018. doi:10.1016/j.trc.2018.04.009.

Bouman, P., Agatz, N., and Schmidt, M., "Dynamic programming approaches for the traveling salesman problem with drone," *Networks*, Vol. 72, No. 4, 2018, pp. 528–542.

Poikonen, S., Golden, B., and Wasil, E. A., "A branch-and-bound approach to the traveling salesman problem with a drone," *Informs Journal on Computing*, Vol. 31, No. 2, 2019, pp. 335–346.

De Freitas, J. C., and Penna, P. H. V., "A variable neighborhood search for flying sidekick traveling salesman problem," *International Transactions in Operational Research*, Vol. 27, No. 1, 2020, pp. 267–290.

De Freitas, J. C., and Penna, P. H. V., "A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem," *Electronic Notes in Discrete Mathematics*, 2018. doi:10.1016/j.endm.2018.03.013.

Boysen, N., Briskorn, D., Fedtke, S., and Schwerdfeger, S., "Drone delivery from trucks: Drone scheduling for given truck routes," *Networks*, Vol. 72, No. 4, 2018, pp. 506–527.

Jeong, H. Y., Song, B. D., and Lee, S., "Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones," *International Journal of Production Economics*, Vol. 214, 2019, pp. 220–233.

Dayarian, I., Savelsbergh, M., and Clarke, J.-P., "Same-day delivery with drone resupply," *Transportation Science*, 2020.

Kim, S., and Moon, I., "Traveling salesman problem with a drone station," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49, No. 1, 2018, pp. 42–52.

Carlsson, J. G., and Song, S., "Coordinated logistics with a truck and a drone," *Management Science*, Vol. 64, No. 9, 2018, pp. 4052–4069.

Ferrandez, S. M., Harbison, T., Weber, T., Sturges, R., and Rich, R., "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *Journal of Industrial Engineering and Management (JIEM)*, Vol. 9, No. 2, 2016, pp. 374–388.

Figliozzi, M. A., “Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO2e emissions,” *Transportation Research Part D: Transport and Environment*, Vol. 57, 2017, pp. 251–261.

Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S., “Vehicle routing problems for drone delivery,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 1, 2016, pp. 70–85.

Wang, X., Poikonen, S., and Golden, B., “The vehicle routing problem with drones: several worst-case results,” *Optimization Letters*, 2017. doi:10.1007/s11590-016-1035-3.

Ham, A. M., “Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming,” *Transportation Research Part C: Emerging Technologies*, Vol. 91, 2018, pp. 1–14.

Ulmer, M. W., and Thomas, B. W., “Same-day delivery with heterogeneous fleets of drones and vehicles,” *Networks*, Vol. 72, No. 4, 2018, pp. 475–505.

Wang, Z., and Sheu, J.-B., “Vehicle routing problem with drones,” *Transportation research part B: methodological*, Vol. 122, 2019, pp. 350–364.

Sacramento, D., Pisinger, D., and Ropke, S., “An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones,” *Transportation Research Part C: Emerging Technologies*, Vol. 102, 2019, pp. 289–315.

Ahmad, F., Alam, M. S., Alsaidan, I. S., and Shariff, S. M., “Battery swapping station for electric vehicles: opportunities and challenges,” *IET Smart Grid*, 2020.

Watkins, S., Burry, J., Mohamed, A., Marino, M., Prudden, S., Fisher, A., Kloet, N., Jakobi, T., and Clothier, R., “Ten questions concerning the use of drones in urban environments,” *Building and Environment*, Vol. 167, 2020, p. 106458.