

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xxx/ACCESS.2017.DOI

# A Hybrid Genetic Algorithm on Routing and Scheduling for Vehicle-Assisted Multi-Drone Parcel Delivery

KAI PENG<sup>1</sup>, JINGXUAN DU<sup>1</sup>, FANG LU<sup>1</sup>, QIANGUO SUN<sup>1</sup>, YAN DONG<sup>1</sup>, PAN ZHOU<sup>1</sup>, AND MENGLAN HU<sup>1</sup>,

<sup>1</sup>School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Menglan Hu (humenglan@hust.edu.cn).

This work was supported in part by the National Natural Science Foundation of China under Grant 91538203, 61871436, 61872415, 61502193, 61671216, 61702204, 61872416, and 61471408.

**ABSTRACT** In recent years, unmanned aerial vehicles (UAVs) have exhibited significant market potential to greatly reduce the cost and time in the field of logistics. The use of UAVs to provide commercial courier has become an emerging industry, remarkably shifting the energy use of the freight sector. However, due to limited battery capacities, the flight duration of civilian rotorcraft UAVs is still short, hindering them from performing remote jobs. In this case, people customarily utilize ground vehicles to carry and assist UAVs in various applications, including cargo delivery. Most previous studies on vehicle-drone cooperative parcel delivery considered only one UAV, thereby suffering from low efficiency when serving a large number of customers. In this paper, we propose a novel Hybrid Genetic Algorithm, which supports the cooperation of a ground vehicle and multiple UAVs for efficient parcel delivery. Our routing and scheduling algorithm allows multiple UAVs carried by the vehicle to simultaneously deliver multiple parcels to customers residing in different locations. The proposed algorithm consists of a pipeline of several modules: population management, heuristic population initialization, and population education. Performance evaluation results show that the proposed algorithm has significant efficiency over existing algorithms.

**INDEX TERMS** unmanned aerial vehicle, cargo delivery, routing, scheduling.

## I. INTRODUCTION

The enabling Internet-of-Things (IoT) technology has inspired a large number of novel platforms and applications [1], [2], [3], [4], [5], [6]. Among them, one popular IoT platform is the Unmanned Aerial Vehicle (UAV), which owns multi-fold advantages, fast, flexible, lightweight, low-cost, and easy to use. As a result, rotorcraft UAVs have been widely deployed to support a myriad of applications in various civilian and military fields, including sensing and inspection [7], [8], [9], disaster warning [10], [11], edge computing [12], [13], [14], wireless sensor networks [15], [16], [17], vehicle networks [18], [19], [20], cargo delivery [21], [22], [23], and so on.

Such kinds of applications often require UAVs to work across a wide area and visit multiple different locations to execute data collection or cargo distribution. Nevertheless, refrained by limited battery capacities, the hovering time of rotorcraft UAVs is still quite short, hindering them from per-

forming various remote jobs. In this case, people customarily utilize ground vehicles to carry and assist UAVs in various applications, including cargo delivery. This vehicle-drone cooperative cargo delivery mode benefits from both the large capacity of goods transportation of ground vehicles and high mobility of UAVs [24]. However, this working mode brings new challenges in designing smart routing approaches which plan paths and schedule route ground vehicles and UAVs for efficient collaborations.

A number of work on vehicle-drone collaboration has been proposed in the literature [7], [8], [25]. Efficient routing and scheduling strategy can greatly reduce the time consumption and financial expenses incurred in logistics distribution. A few studies investigated leveraging vehicles and UAVs to collaborate in delivering cargos to customers [26]. [21], [27]. Nevertheless, the prior work customarily considers only one UAV in vehicle-drone cooperative parcel delivery. In this paper, we employ multiple UAVs which are carried by

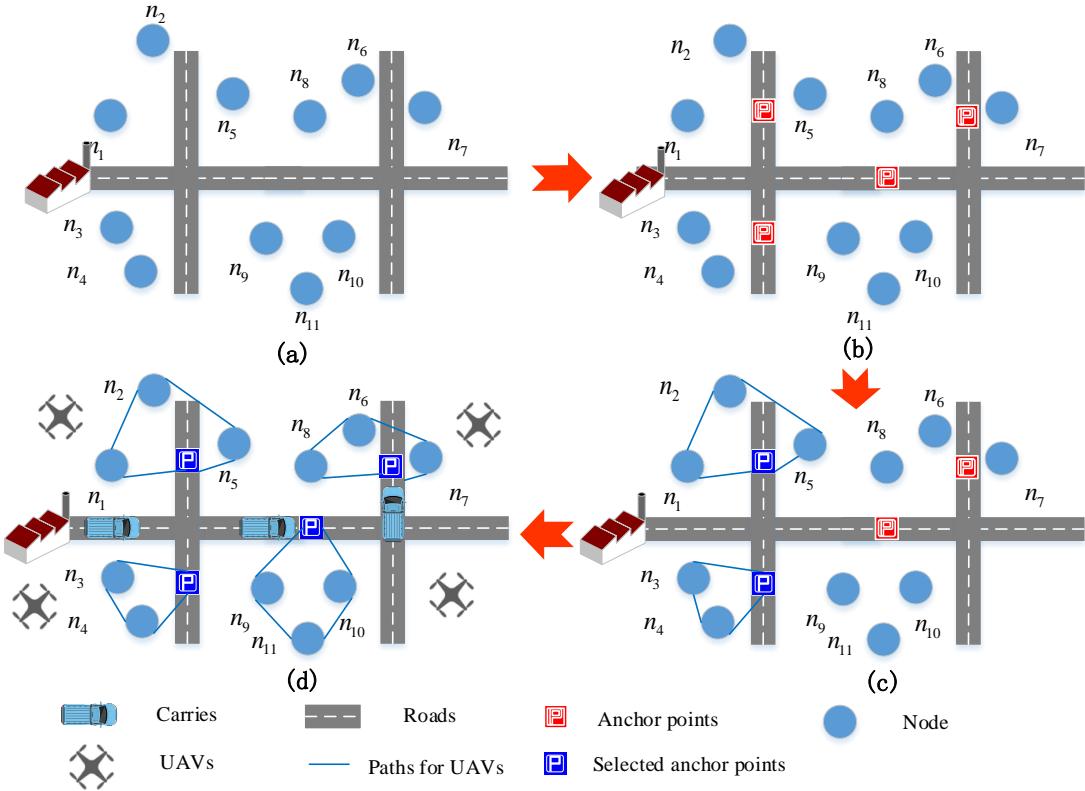


FIGURE 1: Illustration of the scenario where UAVs serve the customers with a carrier.

one vehicle and manage to simultaneously deliver multiple parcels to customers residing in different locations.

Fig.1 depicts a vehicle-drone cooperative cargo delivery scenario. There are roads near the nodes which referred to the customers. Candidate anchor points are at regular intervals on these roads. The selection of proper anchor points is a Facility Location Problem (FLP) [28]. UAVs are released from the vehicle at each selected anchor point and visit customers for the delivery along the planned paths, then return back to the vehicle after the service. The delivery tasks for the nearby customers are abstracted as a Vehicle Routing Problem (VRP) and Travelling Salesman Problem (TSP) [28] with a travel distance constraint in a single UAV path. In this paper, we also need to consider about the route distribution delivered to UAVs which is a Bin Packing Problem (BPP) [29]. The purpose in this paper is to minimize the overall delivery time by path planning for UAVs and the vehicle. This problem is a combinatorial optimization problem, and each sub-problem is NP-Hard. Although the VRP, TSP, FLP, and BPP have been studied in other literatures, we cannot separately apply them to the synthesized problem proposed in this article.

We investigate the novel combinatorial optimization problem and propose a novel Hybrid Genetic Algorithm (HGA). The algorithm consists of population management, population initialization, and population education. HGA can coordinate the complexity and the performance by jump-

ing out the local optimum. Moreover, HGA is applicable to other problems with the minor adjustments. In addition, we propose a Low Visit Cost Crossover algorithm (LVC) that selects genetic fragments based on nodes and anchor points, respectively. LVC ensures a better distribution in the population to avoid premature convergence. Furthermore, we also design a three-hierarchical education algorithm, namely the anchor point based education, path based education, and node based education.

The rest of this paper is organized as follows: Section II discusses the related work, Section III states the problem formulation and notations, Section IV presents the overview of the proposed algorithm, Section V discusses each step of the algorithm in detail, Section VI presents a performance evaluation, and section VII concludes this paper.

## II. RELATED WORK

In recent years, a significant number of work on UAV-based services and delivery has been proposed in the literature. Motlagh et al. [12] provided a survey on low-altitude UAV-based services and focused on UAV-based communications and remote sensing. Yuan et al. [10] exhibits that UAV is a superior platform for sensing and surveillance. Otto et al. [30] surveyed optimization approaches for civil applications of UAVs, with an emphasis in operations planning.

The routing problem for vehicle-drone cooperation can be specified by variations of Vehicle Routing Problems (VRP)

[31], such as 2E-VRP [32]. In a 2E-VRP, a number of primary and secondary vehicles travelling across a two-echelon routing systems to deliver goods to customers. In 2E-VRP, the primary and secondary vehicles operate separately, while in this paper, we a ground vehicle carries multiple drones and thus synchronization is required in their cooperation.

Based on the 2E-VRP model, a few studies on the routing problem for vehicle-drone cooperative sensing has been reported. Halil Savuran et al. [33] proposed a route optimization method for a vehicle-carried UAV based on Genetic Algorithms [34]. Manyam et al. [35] formulated a mixed integer programming problem and designed a branch-and-cut approach to tackle the routing problem. The above studies considered on vehicle-drone cooperation focused on UAV sensing and inspection. Also, they assumed only one UAV in their formulation, which would be inefficient in large-scale IoT applications. In contrast, in this paper we propose to utilize multiple UAVs to simultaneously deliver cargos to customers in parallel, significantly improving efficiency in goods distribution.

Over the last decade, UAV-based parcel delivery has received significant attention in academia. Murray et al. [21] studied a Flying Sidekick Traveling Salesman Problem (F-STSP) for UAV-based cargo delivery. The authors in [21] designed heuristic algorithms for two UAV delivery VRP problems. Another work [36] presented a simulated-annealing-based heuristic algorithm to tackle FSTSP. Ferrandez et al. in [27] proposed to first find the optimal launch locations by clustering and then determine the vehicle route by using a genetic algorithm. Campbell et al. [37] formulated and optimized models for drone delivery in collaboration with a truck. Another work [23] derived a number of worst-case results on VRP with drones, i.e., the maximum savings that can be obtained from using drones. In these previous studies, they customarily considered only one UAV in vehicle-drone cooperative goods distribution, while in this paper we propose to leverage multiple UAVs to simultaneously deliver cargos to customers.

Our proposed algorithm is inspired by [38], which presented an effective hybrid genetic algorithm framework for conquering VRP problems. It may be noted our problem comprises VRP as sub-problems. The idea of [38] lies in leveraging crossover and education (i.e., local search) to produce high-quality solutions. Olivera and Viera [39] proposed a heuristic algorithm to handle a Vehicle Routing Problem with Multiple Trips (VRPMT), which is also a sub-problem of the problem investigated in this paper. They considered a scenario where one carrier is allowed to perform multiple trips while the duration of routes assigned to the same carrier is finite. In another work [40], Cheikh et al. designed a local search heuristic for VRPMT, through which encouraging solutions are obtained. Nevertheless, our proposed method not only employs HGA as its framework, but also leverages a set of local search operations for parking spot selection, route planning, and route assignment.

TABLE 1: Notation and Terminology

Notation	Definition
$N$	a set of nodes to be served
$D$	a set of anchor points to be selected
$U$	a set of UAVs employed
$n_i$	a customer
$u_i$	an UAV
$d_i$	an anchor point
$r_i$	a path for an UAV
$M$	the population which has several initial solutions
$P$	the parent solution which is select from the population
$C$	the child solution after crossover
$T$	a set containing all the UAV paths

### III. PROBLEM FORMULATION

In this paper, we investigate a routing problem for vehicle-drone cooperative parcel delivery. A set of UAVs can visit and serve customers along planned route. However, due to the restrict of the battery capacity, a single UAV only visits customers within a small range during its hanging time. In a word, only drones cannot meet the need to serve all customers in a wide area at once. Accordingly, a vehicle is employed to act as a mobile base station and expand the reachable scope of UAVs. We assume that there is a road network arranged in the target area. Candidate anchor points are selected and sequentially visited by the vehicle with multiple UAVs. At each selected anchor point, assigned UAVs are released to serve the nearby customers along the planned route. When the service task in this anchor point is completed, the vehicle carried with UAVs leaves for the next selected anchor point. Both the vehicle and UAVs repeat the procedure till all customers are visited.

This problem can be decomposed as three subproblems: First, the selection of proper anchor points is a Facility Location Problem (FLP). Second, the delivery tasks are abstracted as a Vehicle Routing Problem (VRP) and Travelling Salesman Problem (TSP) with a travel distance constraint in a single UAV path. Third, the route distribution delivered to UAVs is a Bin Packing Problem (BPP).

We consider a graph  $G = \{V, E\}$  containing  $V$  as the vertex set and  $E$  as the edge set to model this problem. The set  $V$  consists of two subsets,  $N = \{n_1, n_2, n_3, \dots\}$  and  $D = \{d_0, d_1, d_2, d_3, \dots\}$ , representing the set of customers and the set of candidate anchor points.  $d_0$  in  $D$  is the given depot that is the start and end of this problem. A cost matrix  $A = (a_{ij})$  is defined to represent the distance between a node  $n_i$  and a node  $n_j$ . The UAVs are denoted as  $U = \{u_1, u_2, \dots, u_{N_u}\}$ .  $V_{carrier}$  and  $V_{uav}$  are the speed of the carrier and the UAVs.

First, candidate anchor points should be selected and compose the route of the vehicle. There are multiple accessible roads to connect the target region on which all the possible anchor points are distributed. To optimize the overall delivery time, proper anchor points are selected through certain rules from the set  $D$ . Once we select the proper anchor points,

customers are assigned to their nearest anchor point.

Second, the UAVs fly to serve the vicinal customers along the planned route. We apply  $r_i = \{n_1, n_2, \dots, n_m\}$  to represent a single UAV route and  $n_m \in V$  like in Fig.1. A single route is defined as a process that the UAV sets off from the carrier, serve all the distributed customers and finally return to the carrier. At most one node of  $r_i$  belongs to  $D$ .  $R = \{r_1, r_2, \dots, r_{n_r}\}$  is the set of all routes in a solution. For each pair of  $(n_m, r_i)$  where  $n_m \in V$  and  $r_i \in R$ , let  $\alpha(n_m, r_i)$  define whether the route  $r_i$  covers the node  $n_m$  ( $\alpha(n_m, r_i) = 1$ ) or not ( $\alpha(n_m, r_i) = 0$ ). For each pair of  $(r_i, d_k)$  where  $r_i \in R$  and  $d_k \in D$ , let  $\beta(r_i, d_k)$  denote whether the route  $r_i$  is connected to the anchor point  $d_k$  ( $\beta(r_i, d_k) = 1$ ) or not ( $\beta(r_i, d_k) = 0$ ). For each pair of nodes  $(n_{m_1}, n_{m_2}) \in V$  and each route  $r_i \in R$ , let  $\gamma_{r_i}(n_{m_1}, n_{m_2})$  define whether the node  $n_{m_1}$  is right next to the node  $n_{m_2}$  within the route  $r_i$  ( $\gamma_{r_i}(n_{m_1}, n_{m_2}) = 1$ ) or not ( $\gamma_{r_i}(n_{m_1}, n_{m_2}) = 0$ ). The length of a route  $r_i \in R$  is calculated as

$$l(r_i) = \sum_{n_{m_1} \in r_i} \sum_{n_{m_2} \in r_i} \gamma_{r_i}(n_{m_1}, n_{m_2}) a_{m_1 m_2}. \quad (1)$$

If the UAVs are not able to serve all the distributed customers near the anchor point after a single route, some of them can be sent away again to the remainder customers after charging their batteries. This comes down to the scheduling strategy of UAVs and details are described hereinafter. For each pair of  $(r_i, u_j)$  where  $r_i \in R$  and  $u_j \in U$ , we use  $\theta(r_i, u_j)$  to denote whether the route  $r_i$  is assigned to the UAV  $u_j$  ( $\theta(r_i, u_j) = 1$ ) or not ( $\theta(r_i, u_j) = 0$ ). The total distance that the UAV  $u_j$  travels at the anchor point  $d_k$  is defined as:

$$l(u_j, d_k) = \sum_{r_i \in R} \beta(r_i, d_k) \theta(r_i, u_j) l(r_i) \quad (2)$$

When UAVs finish the service tasks for all nearby customers, the carrier drives to the next parking slot and continue to execute the above process. The carrier returns to the given depot when the whole delivery process is over.

Third, the sequence of selected anchor point need to be determined. We apply  $r_0 = \{d_{k_1}, d_{k_2}, \dots, d_{k_{n_d}}\}$  to represent the route for the carrier. Correspondingly,  $\gamma_{r_0}(n_{m_1}, n_{m_2})$  denotes whether the node  $n_{m_1}$  is right next to the node  $n_{m_2}$  within the carrier route  $r_0$  ( $\gamma_{r_0}(n_{m_1}, n_{m_2}) = 1$ ) or not ( $\gamma_{r_0}(n_{m_1}, n_{m_2}) = 0$ ).  $\alpha(n_m, r_0)$  denotes whether the node  $n_m$  is selected as an anchor point ( $\alpha(n_m, r_0) = 1$ ) or not ( $\alpha(n_m, r_0) = 0$ ). The total distance that the carrier travels is described as:

$$l(carrier) = l(r_0) = \sum_{n_{m_1} \in r_0} \sum_{n_{m_2} \in r_0} \gamma_{r_0}(n_{m_1}, n_{m_2}) a_{m_1 m_2}. \quad (3)$$

The total distance cost includes the distance cost of the vehicle, the total distance cost of UAVs and is denoted as

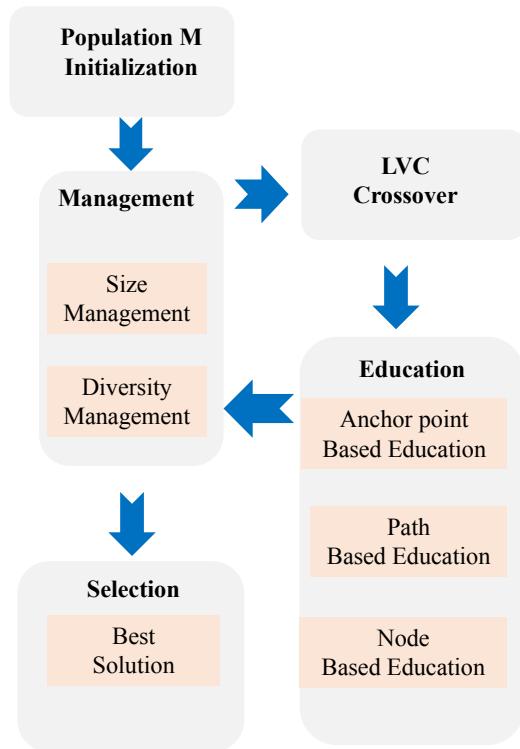


FIGURE 2: Flowchart of HGA.

$C(distance)$ .  $C(time)$  is defined as the total time consumption that the carrier leaves the given depot, finishes the charging task for all customers, and returns to the depot. Our target is to get the minimum value of  $C(time)$ . In this paper, the problem is formulated as an integer programming problem, aiming to minimize the total distance cost and time consumption on the premise of meeting several conditions:

$$C(distance) = l(carrier) + \sum_{u_j \in U} \sum_{d_k \in D} l(u_j, d_k) \quad (4)$$

$$C(time) = \frac{l_{carrier}}{V_{carrier}} + \frac{\sum_{d_k \in D} \alpha(d_k, r_0) Max(l(u_j, d_k))}{V_{uav}} \quad (5)$$

s.t.

$$N_u \geq 1 \quad (6)$$

$$0 \leq l(r_i) \leq Maxd_u \quad (7)$$

$$\sum_{d_k \in D} \alpha(d_k, r_0) = N_d \quad (8)$$

$$\sum_{d_k \in D} \alpha(d_k, r_i) = 1 \quad \forall r_i \in R \quad (9)$$

Constraint (6) indicates that at least one UAV is employed to ensure that a solution is achieved by the combination of UAVs and a vehicle. Constraint (7) guarantees that the length of a single UAV route is not exceed the limit of the maximum single flight distance of the UAV. Constraint (8) announces that all the selected parking slots in a solution must be visited

by the vehicle and each one shall be visited once. Constraint (9) states that a route should be connected to only one anchor point.

#### IV. THE OVERVIEW OF HGA

In this section, we propose the overall design of the novel Hybrid Genetic Algorithm(HGA) to solve the above problem. It simulates the evolutionary process of the "survival of the fittest" rule in nature and helps us gain the optimal solution.

##### A. THE STRUCTURE OF HGA

The HGA initializes a population  $M$  which contains feasible solutions of the problem at first. Then, it iteratively selects two parent solutions  $P_1$  and  $P_2$  from  $M$ , and generates a child solution with smaller cost by the LVC Crossover Algorithm which is described in later section. The local search algorithm is applied to optimize the child solution.  $M$  is updated when a better solution is obtained. The flowchart of HGA is shown in Fig.2 and the description is in Algorithm1:

##### Algorithm 1 Overview of HGA

- 1: Initialize the population  $M$
- 2: **for**  $i = 0 \rightarrow IT_{max}$  **do**
- 3:   (a)Two parent  $P_1$  and  $P_2$  are selected from the population  $M$ .
- 4:   (b)Use the LVCC Algorithm to get an offspring  $C_1$ .
- 5:   (c)Educate the offspring  $C_1$  with the Local Search Algorithm to obtain the optimized solution  $S$
- 6:   (d)Update the population  $M$  with  $S$  by the Population Management procedure
- 7: **end for**
- 8: Return the solution  $S$  with minimum  $C(\text{time})$  or  $C(\text{distance})$  in the population  $M$

##### B. SOLUTION REPRESENTATION

In this part, we stipulate the representation of a feasible solution. According to the problem formulation, a solution  $S$  is composed of two parts: the selected anchor points in the road( $r_0 = \{d_0, d_1, d_2, \dots\}$ ), and the UAV paths around them( $R = \{R_{d_0}, R_{d_1}, R_{d_2}, \dots\}$ ).  $R_{d_k} = \{R_{d_k}^{u_1}, R_{d_k}^{u_2}, \dots, R_{d_k}^{u_j}\}$  is the set of paths assigned to UAVs at the  $d_k$  anchor point. For each UAV, a path is represented as  $r_i = \{d_j, n_k, \dots\}$  where  $d_j \in D$  and  $\{n_k, \dots\} \in N$ .

#### V. THE DESIGN OF HGA

On the basis of the HGA structure, solution construction and optimization are described hereinafter in detail.

##### A. POPULATION INITIALIZATION

In order to generate the population  $M$ , feasible solutions need to be constructed at first. Optional anchor points are determined by the anchor point selection procedure. Then, the vehicle and UAV routes are obtained by a route construction procedure. Last, a scheduling strategy is adopted to

##### Algorithm 2 Solution Construction

- 1:  $M \leftarrow \emptyset$
- 2: **for**  $i$  to  $IT_{max}$  **do**
- 3:    $D_{temp} \leftarrow D$
- 4:    $N_{temp} \leftarrow N$
- 5:   Records the set of anchor points covered by each node in the  $N_{temp}$ , Select those node that contain only one anchor point  $d_{must}$
- 6:    $D_{final} \leftarrow d_{must} \cup D_{final}$
- 7:   select all node contained by  $d_{must}$  as  $N_{num}$
- 8:    $N_{final} \leftarrow N_{num} \cup N_{final}$
- 9:    $N_{temp} \leftarrow N_{temp} \setminus N_{num}$
- 10:    $D_{temp} \leftarrow D_{temp} \setminus d_{must}$
- 11:   Records the nodes set  $T_j$  covered by anchor point  $d_j$ ,the anchor point  $d_j$  is assigned a weight denoted by  $\alpha_i = \frac{|T_j|}{\sum |T_j|}$
- 12:   **while**  $N_{temp} \neq \emptyset$  **do**
- 13:     The anchor point  $d_{temp}$  is randomly selected from the set  $D_{temp}$  according to the weight  $\alpha_i$
- 14:      $D_{temp} \leftarrow D_{temp} \setminus d_{temp}$
- 15:     **if**  $N_{num} \leftarrow \emptyset$  **then**
- 16:       continue // $N_{num}$  denotes node set covered by anchor point  $d_{must}$
- 17:     **end if**
- 18:      $D_{final} \leftarrow d_{temp} \cup D_{final}$
- 19:      $N_{final} \leftarrow N_{num} \cup N_{final}$
- 20:      $D_{temp} \leftarrow D_{temp} \setminus d_{temp}$
- 21:      $N_{temp} \leftarrow N_{temp} \setminus N_{num}$
- 22:   **end while**
- 23:   **while**  $D_{final} \neq \emptyset$  and  $N_{final} \neq \emptyset$  **do**
- 24:     Fetches the first anchor point  $d$  from the set  $D_{final}$  and fetches the first set  $N_{num}$  from the set  $N_{final}$
- 25:      $D_{final} \leftarrow D_{final} \setminus d$
- 26:      $N_{final} \leftarrow N_{final} \setminus N_{num}$
- 27:     The sweep algorithm is used to scan the node set  $N_{num}$  near the anchor point to generate the path set  $T_{temp}$ ,and the cost function  $C_i$  is calculated.
- 28:      $T \leftarrow T_{temp} \cup T$
- 29:      $C \leftarrow C_i \cup C$
- 30:   **end while**
- 31:   If the solution satisfy the reciprocity of solutions,then add solution  $S$  to  $M$
- 32: **end for**

reasonably allocate the UAV routes. The flow of algorithm is shown in Algorithm2.

##### 1) Anchor point selection

The main idea of the procedure is to find the logical anchor points in the target area. If the distance between an anchor point and a customer is less than  $d_u/2$ , we define that the anchor point covers the customer. The heuristic repeatedly selects the anchor points which covers the most customers until all customers are covered by the selected ones. After-

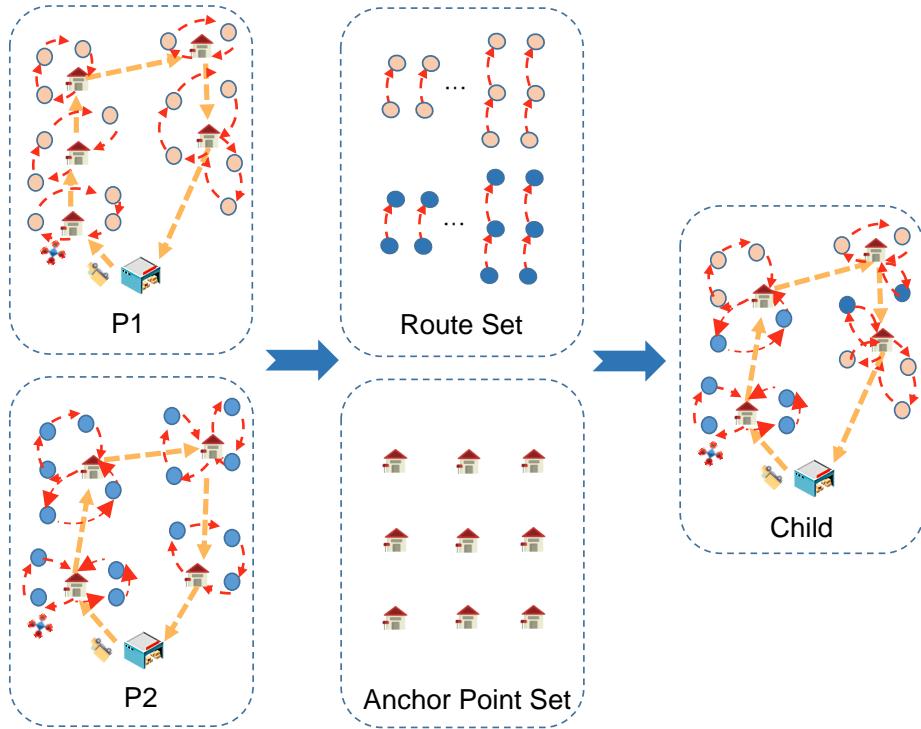


FIGURE 3: Illustration of LVC crossover algorithm.

wards, every customer is assigned to its nearest anchor points.

### 2) Route Construction

For each anchor point, we gain a working set initially containing all customers assigned to it. We first select the furthest customer and build an initial loop route which departs from the anchor point to the selected customer and then return to the anchor point. Then, the route is extends by iteratively inserting the nearest customers to the route. Once a customer is included in a route, it is removed from the working set. When the length of a route is greater than the maximum flying range, the insertion is not admitted and the construction of the current route is terminated. Iteratively in this way, selected anchor points serve all customers assigned to it.

### 3) UAV Scheduling

In UAV scheduling procedure, for each selected anchor point, we first add all the routes that connected to the anchor point  $d_k$  into  $R_{d_k}$  and then iterate to allocate the routes to the UAVs. At every iteration,  $r_i$  is selected with the greatest length among all unassigned routes in  $R_{d_k}$ . The route  $r_i$  is allocated to the UAV  $u_j$  with the minimum tasks.

## B. PARENT SOLUTION SELECTION AND LVCC

We select two parent solutions from the population  $M$  as  $P1$  and  $P2$  to generate a child solution  $C$ . The parent selection procedure is performed by the Binary Tournament Algorithm.

After the parent selection, we propose a novel Low Visit

Cost Crossover(LVCC) algorithm to select the better gene fragment on the chromosome(solution) and compose a child solution with lower cost. The schematic diagram is shown in Fig.3. In this way, better fragments of the parental gene can be passed on to the child, resulting in a better solution.

Algorithm3 is the pseudo code of the LVCC. The crossover algorithm extracts all routes and all anchor points of  $P1$  and  $P2$ , and puts them in  $R_{total}$  and  $D_{total}$  respectively. The route selection procedure in LVCC iteratively selects appropriate routes from  $R_{total}$  with the minimum cost. Notice that a customer shall not be included by multiple routes. The selected customers are deleted from  $R_{total}$ .

The anchor point selection procedure in LVCC is described as follows. First, the route coverage ratio of an anchor point ( $\mu_{d_i}$ ) is defined.  $|Cv_{r_i}|$  represents the number of routes covered by the anchor point  $d_i$ .

$$\mu_{d_i} = \frac{|Cv_{r_i}|}{|R_{final}|} \quad (10)$$

The anchor point with the largest  $\mu_{d_i}$  is selected. After the selection, we assign each route to the nearest selected anchor point to minimize the cost. The route for the vehicle to visit the selected anchor points is a TSP. The dominant gene of  $P1$  and  $P2$  is passed to the child  $C$  by the above steps.

## C. EDUCATION

After the LVCC algorithm, we obtain a better child solution  $C$  which inherits the dominant gene of  $P1$  and  $P2$ . In this section, the Local Search algorithm is adopted to replace the mutation factor of the traditional genetic algorithm and it

---

### Algorithm 3 LVCC

---

```

1: Use the binary tournament algorithm to select two solutions from the population  $M$  as  $P1$  and  $P2$ .
2: Extract all paths of  $P1$  as  $R_{P1}$ .
3: Extract all paths of  $P2$  as  $R_{P2}$ .
4:  $R_{total} \leftarrow R_{P1} \cup R_{P2}$ .
5: Extract all anchor points of  $P1$  as  $D_{P1}$ .
6: Extract all anchor points of  $P2$  as  $D_{P2}$ .
7:  $D_{total} \leftarrow D_{P1} \cup D_{P2}$ .
8:  $R_{final} \leftarrow \emptyset$ 
9:  $D_{final} \leftarrow \emptyset$ 
10: while  $R_{total} \neq \emptyset$  do
11:   select the minimum visit cost path  $r_i$  in  $R_{total}$ .
12:    $R_{final} \leftarrow R_{final} \cup r_i$ .
13:    $R_{total} \leftarrow R_{total} \setminus r_i$ 
14:   All nodes in  $r_i$  will be deleted from  $R_{total}$ .
15: end while
16: while  $R_{final} \neq \emptyset$  do
17:   select  $d_i$  with the maximum  $\mu_{di}$  from  $D_{total}$ .
18:    $D_{final} \leftarrow D_{final} \cup d_i$ .
19:   select all paths  $R_{di}$  covered by  $d_i$  in  $R_{final}$  as  $R_{temp}$ .
20:   for  $r_i$  in  $R_{temp}$  do
21:      $R_{final} \leftarrow R_{final} \setminus r_i$ 
22:   end for
23: end while

```

---

includes three procedure: the anchor point based education, the route based education, and the customer based education.

#### 1) Anchor Point Based Education

The anchor point based education mainly includes two kinds of operation: **delete** and **replace**. Notice that if  $\mu_{di}$  is less than  $MIN$ , the anchor point  $d_i$  is labeled as a cold point. Algorithm4 depicts the pseudo code of the anchor point based education. The cold points need to be deleted and the routes need to reallocated to other neighboring anchor points. Neighboring anchor points of  $d_i$  are defined as:

$$N(d_i) = \{d_x | a_{ix} < d_u\} \quad (11)$$

Then, we attempt to replace an anchor point with its neighborhoods. The operation takes effects if the visit cost gets lower.

#### 2) Route Based Education

The route based education mainly implements the operation **merge**. We merge short routes to decrease the cost of an anchor point. We define a route as a short one with the length of it less than  $Dis$ . As shown in Algorithm4, in the procedure, we merge routes with its neighboring routes that are defined as:

$$N(r_i) = \{r_x | max(a_{ix}) < d_u, n_i \in r_i \text{ and } n_x \in r_x\} \quad (12)$$

The operation is implemented if the merged route satisfies that the length is less than  $d_u$  and the cost of the anchor point

---

### Algorithm 4 Anchor Point Based Education

---

```

1:  $D_{delete} \leftarrow \emptyset$ 
2:  $D_{temp} \leftarrow D_{final}$ 
3: for  $d$  in  $D_{temp}$  do
4:   Select first element  $d$  from  $D_{temp}$ 
5:   if  $d$  contains fewer paths than  $MIN$  then
6:     All paths contained in  $d$  are assigned other anchor points
7:     if distance constraints are satisfied then
8:        $D_{delete} = D_{delete} \cup d$ 
9:        $D_{temp} \leftarrow D_{temp} \setminus d$ 
10:    end if
11:   end if
12: end for

```

---

### Algorithm 5 path based education

---

```

1:  $R_{short} \leftarrow \emptyset$ 
2:  $D_{temp} \leftarrow D_{final}$ 
3: for  $d$  in  $D_{temp}$  do
4:    $D_{temp} \leftarrow D_{temp} \setminus d$ 
5:   Extract all the short paths contained in  $d$  and add to  $R_{short}$ 
6: end for
7: while  $R_{short}$  is not Empty do
8:   Select a pair path  $(r_i, r_j)$  from the  $R_{short}$ 
9:   if distance between  $r_i$  and  $r_j$  less than  $DIS$  then
10:     $r_{new} \leftarrow \text{merge path } r_i \text{ and } r_j$ 
11:    if  $r_{new}$  satifys the UAV distance constraint then
12:       $R_{short} \leftarrow R_{short} \setminus (r_i, r_j)$ 
13:    end if
14:  end if
15: end while

```

---

gets lower.

#### 3) Costumer Based Education

In this procedure, we adjust the order of customers in a route so as to reduce the cost. The customer based education mainly contains two kinds of operation: **exchange** and **replace**. We exchange the visit order of customers in a route. If the cost decreases, the operation is confirmed. Then, we adjust the customers in two neighboring routes to reduce the cost.

### D. POPULATION MANAGEMENT

To add a solution into the population  $M$ , two aspects are under consideration, the Diversity Management and Size Management.

#### 1) Diversity Management

The population  $M$  is initialized to obtain solutions in a random way. Any solutions in  $M$  are unique to avoid useless computational consumption and the premature convergence.

To guarantee the uniqueness speed up the calculation, we propose a strict condition that the difference of the cost

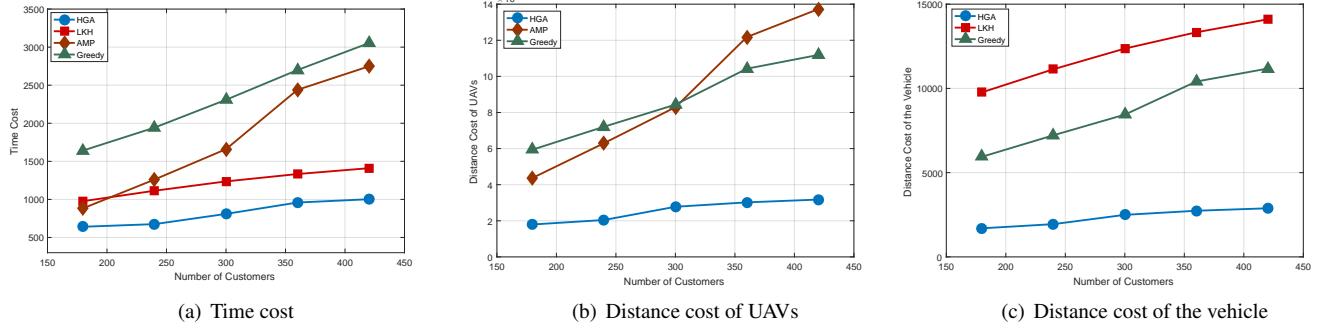


FIGURE 4: Results with varying number of customers and varying customer density.

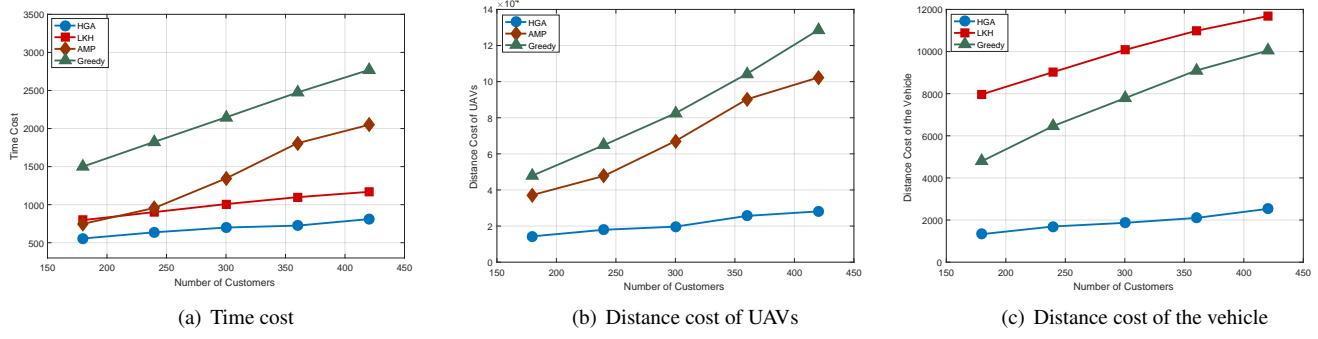


FIGURE 5: Results with varying number of customers and fixed customer density.

functions between each pair of solutions in  $M$  must be more than a threshold  $\Delta > 0$ :

$$\forall P1, P2 \in M, P1 \neq P2, |C(P1) - C(P2)| > \Delta \quad (13)$$

If  $\Delta \neq 0$ , it means  $C(P1) \neq C(P2)$ , so as that  $P1$  and  $P2$  are well spaced. Otherwise, it is necessary to compare the cost function of each anchor point in  $P1$  with that of  $P2$ .

## 2) Size Management

We define a tunable parameter  $M_{max}$  to fix the scale of the population  $M$ . If the size of  $M$  is greater than  $M_{max}$ , we remove  $|M| - M_{max}$  solutions with the greatest  $C(s)$  values from the population  $M$ .

## VI. PERFORMANCE EVALUATION

In order to solve the routing for vehicle-assisted multi-drone parcel delivery problem, we put forward the novel hybrid genetic algorithm. A lot of experiments have been conducted to evaluate the performance of HGA. To intuitively and overall analyze the performance and effect of the proposed algorithm, we utilize three algorithms as the baselines, namely Lin-Kernighan Heuristic (LKH) [41] a greedy algorithm (Greedy), and AMP (a relaxed version of AMP) [39]. LKH is an effective heuristic algorithm for handling classical TSPs. In our experimental implementation, we employ the vehicle to visit every customer from the depot in the order decided by

LKH without considering UAVs. The Greedy heuristic allows the UAVs to serve all the customers by producing multi-trips and select reasonable anchor points for all the trips. AMP is designed for MTVRP and we need to relax the route duration constraint and gain a feasible solution. We presume that the battery capacity of UAVs is unlimited when we implement AMP.

We present the comparison results in the following forms to analyze the effectiveness and application of our proposed method more comprehensively. Two performance metrics draw our special attention. The first one is the total time cost of a solution. It is calculated from the cooperative vehicle-drone system leaving the depot, serving all the customers, to returning to the depot. Obviously, less time consumption corresponds to higher efficiency. The second one is the total distance cost of a solution. We define the distance cost in Section II and it is composed of the carrier travelling distance and the UAV travelling distance. Likewise, less distance cost stands for an efficient system.

In our simulation,  $N_n$  customers within a  $ll$  region is generated to better verify the practicability of our proposed algorithm. A vehicle carried with 5 UAVs is employed to complete the parcel delivery task in the target region. We set the speed of the carrier and s UAV as  $5m/s$  and  $3m/s$ , respectively. The first examination variable is the number of customers. Considering the possible impact of the customer

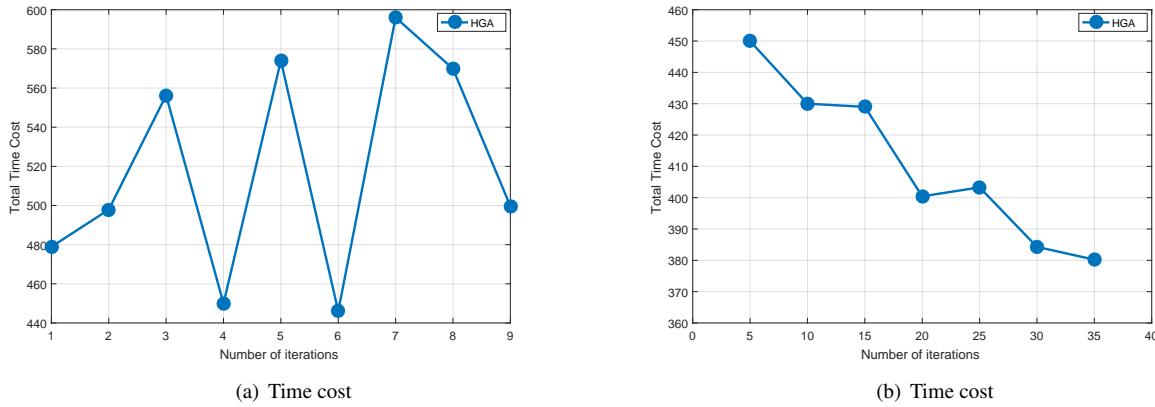


FIGURE 6: Results with varying number of the iteration.

density on performance, we test two scenarios for the overall customer size: changing density and fixed density.

At first, we fix the region as 100100 and vary the number of customers within the range [150, 450]. The customer density is raised in the fixed region as  $N_c$  is increasing. The corresponding results are shown in Figs.4. Second, with  $N_c$  increasing, we vary  $l$  in range[100,500] at the same time so that the customer density is set to be fixed. The corresponding results are depicted in Figs.5. Figs. 4(a) and 5(a) show HGA significantly outperforms other algorithms in terms of total execution time. In addition, from Figs 4(b), 4(c), 5(b), and 5(c), we can observe that HGA also delivers the best performance in terms of distance cost for the vehicle and UAVs, respectively.

Another metric is the iteration run by the algorithm. First, we fix  $N_c = 300$  and  $l = 200$  and vary the iteration from 0 to 9. The results are shown graphically in Fig. 6(a). Fig. 6(a) shows that the algorithm may contain certain randomness but gain the better solution after several times. Last, the iteration changes from 0 to 35 and we choose the best solution in every 5 times to show the overall trend. Fig. 6(b) shows that as the iteration increases, the quality of the solution improves accordingly.

## VII. CONCLUSIONS

In this paper we have proposed a novel Hybrid Genetic Algorithm which solves a multi-drone parcel delivery problem. The algorithm includes three parts: the population initialization, crossover, and education. In the population initialization, anchor points covering more nodes are selected with a higher probability, so as to obtain better solutions. Population management can effectively save computing time. Crossover avoids falling into the local optimal solution. The education, including the anchor point based education, path based education, and node based education, minimizes the total time. The computation results show that the proposed algorithm outperforms the existing algorithms. The propose algorithm can be adopted widely to solve the similar planning

and scheduling problems.

## REFERENCES

- [1] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on wifi sensing: a survey," *IEEE Access*, vol. 6, pp. 13317–13325, 2018.
- [2] H. Zhou, X. Wang, X. Chen, X. Li, and S. Xu, "Data offloading techniques through vehicular ad hoc networks: A survey," *IEEE Access*, vol. 6, pp. 65250–65259, 2018.
- [3] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.
- [4] C. Cai, M. Hu, D. Cao, X. Ma, Q. Li, and J. Liu, "Self-deployable indoor localization with acoustic-enabled iot devices exploiting participatory sensing," *IEEE Internet of Things Journal*, 2019, DOI: 10.1109/JIOT.2019.2900524.
- [5] C. Zhu, V. C. Leung, J. J. Rodrigues, L. Shu, L. Wang, and H. Zhou, "Social sensor cloud: Framework, greenness, issues, and outlook," *IEEE Network*, vol. 32, no. 5, pp. 100–105, 2018.
- [6] C. Cai, M. Hu, X. Ma, K. Peng, and J. Liu, "Accurate ranging on acoustic-enabled iot devices," *IEEE Internet of Things Journal*, 2019, DOI: 10.1109/JIOT.2018.2879371.
- [7] M. Hu, W. Liu, K. Peng, X. Ma, W. Cheng, J. Liu, and B. Li, "Joint routing and scheduling for vehicle-assisted multi-drone surveillance," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2018.2878602, 2018.
- [8] M. Hu, W. Liu, J. Lu, R. Fu, K. Peng, X. Ma, and J. Liu, "On the joint design of routing and scheduling for vehicle-assisted multi-uav inspection," *Future Generation Computer Systems*, vol. 94, pp. 214–223, 2019.
- [9] K. Peng, W. Liu, Q. Sun, X. Ma, M. Hu, D. Wang, and J. Liu, "Wide-area vehicle-drone cooperative sensing: Opportunities and approaches," *IEEE Access*, vol. 7, pp. 1818–1828, 2019.
- [10] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.
- [11] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, "Realtime profiling of fine-grained air quality index distribution using uav sensing," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.
- [12] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, 2016.
- [13] C. Xu, X. Ma, R. Shea, H. Wang, and J. Liu, "Enhancing performance and energy efficiency for hybrid workloads in virtualized cloud environment," *IEEE Transactions on Cloud Computing*, DOI: 10.1109/TCLOUD.2018.2837040, 2018.
- [14] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Transactions on Services Computing*, DOI: 10.1109/TSC.2018.2828426, 2018.

- [15] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Uav-aided projection-based compressive data gathering in wireless sensor networks," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2018.2878834, 2018.
- [16] M. Hu, Z. Chen, K. Peng, X. Ma, P. Zhou, and J. Liu, "Periodic charging for wireless sensor networks with multiple portable chargers," *IEEE Access*, vol. 7, pp. 2612–2623, 2019.
- [17] K. Peng, M. Hu, C. Cai, X. Ma, Y. Yang, and J. Liu, "On simultaneous power replenishment for wireless sensor networks with multiple portable chargers," *IEEE Access*, vol. 6, pp. 63120–63130, 2018.
- [18] C. Huang, Y. Chen, S. Xu, and H. Zhou, "The vehicular social network (vsn)-based sharing of downloaded geo data using the credit-based clustering scheme," *IEEE Access*, vol. 6, pp. 58254–58271, 2018.
- [19] Y. Zhang, M. Chen, N. Guizani, D. Wu, and V. Leung, "Sovcan: Safety-oriented vehicular controller area network," *IEEE Communications*, vol. 55, no. 8, pp. 94–99, 2017.
- [20] M. Chen, Y. Hao, K. Lin, L. Hu, and Z. Yuan, "Label-less learning for traffic control in an edge network," *IEEE Network*, vol. 32, no. 6, pp. 8–14, 2018.
- [21] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.
- [22] F. Wang, F. Wang, X. Ma, and J. Liu, "Demystifying the crowd intelligence in last mile parcel delivery for smart cities," *IEEE Network*, 2019, DOI: 10.1109/MNET.2019.1800228.
- [23] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optimization Letters*, vol. 11, no. 4, pp. 679–697, 2017.
- [24] M. Chen, Y. Tian, G. Fortino, J. Zhang, and I. Humar, "Cognitive internet of vehicles," *Computer Communications*, vol. 120, pp. 58–70, 2018.
- [25] Z. Luo, Z. Liu, and J. Shi, "A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle," *Sensors*, vol. 17, no. 5, pp. 1144–, 2017.
- [26] H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air vehicle deployed on a moving carrier," *Soft Comput.*, vol. 20, no. 7, pp. 2905–2920, 2016.
- [27] S. M. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *Journal of Industrial Engineering and Management*, vol. 9, no. 2, pp. 374–388, 2016.
- [28] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231 – 247, 1992.
- [29] S. Leung, X. Zhou, D. Zhang, and J. Zheng, "Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem," *Computers and Operations Research*, vol. 38, pp. 205–215, 01 2011.
- [30] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.
- [31] S. N. Kumar and R. Panneerselvam, "A survey on the vehicle routing problem and its variants," *Intelligent Information Management*, vol. 4, no. 03, pp. 66–74, 2012.
- [32] R. Cuda, G. Guastaroba, and M. G. Speranza, "A survey on two-echelon routing problems," *Computers & Operations Research*, vol. 55, pp. 185–199, 2015.
- [33] H. Savuran and M. Karakaya, "Route optimization method for unmanned air vehicle launched from a carrier," *Lecture Notes on Software Engineering*, vol. 3, no. 4, pp. 279–284, 2015.
- [34] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994.
- [35] S. G. Manyam, D. W. Casbeer, and K. Sundar, "Path planning for cooperative routing of air-ground vehicles," in *American Control Conference (ACC)*, 2016, pp. 4630–4635, IEEE, 2016.
- [36] A. Ponza, Optimization of drone-assisted parcel delivery. PhD thesis, Universita Degli Studi Di Padova, 2016.
- [37] J. F. Campbell, D. Sweeney, and Z. J. II, "Strategic design for delivery with trucks and drones," in *Technical Report*, 2017.
- [38] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems," *European Journal of Operational Research*, vol. 234, no. 3, pp. 658–673, 2014.
- [39] A. Olivera and O. Viera, "Adaptive memory programming for the vehicle routing problem with multiple trips," *Computers & Operations Research*, vol. 34, no. 1, pp. 28–47, 2007.
- [40] M. Cheikh, M. Ratli, O. Mkaouar, and B. Jarboui, "A variable neighborhood search algorithm for the vehicle routing problem with multiple trips," *Electronic Notes in Discrete Mathematics*, vol. 47, pp. 277–284, 2015.
- [41] K. Helsgaun, "General k-opt submoves for the lin–kernighan tsp heuristic," *Mathematical Programming Computation*, vol. 1, no. 2-3, pp. 119–163, 2009.

•••