Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9778188, 21 pages
https://doi.org/10.1155/2022/9778188

Research Article

Path Planning for Multi-Vehicle-Assisted Multi-UAVs in Mobile Crowdsensing

Jiancheng Song, Liang Liu, Yulei Liu, Jie Xi, and Wenbin Zhai

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Correspondence should be addressed to Liang Liu; liangliu@nuaa.edu.cn

Received 11 November 2021; Revised 16 December 2021; Accepted 31 December 2021; Published 20 January 2022

Academic Editor: Guangwei Yang

Due to the capability of fast deployment and controllable mobility, unmanned aerial vehicles (UAVs) play an important role in mobile crowdsensing (MCS). However, constrained by limited battery capacity, UAVs cannot serve a wide area. In response to this problem, the ground vehicle is introduced and used to transport, release, and recycle UAVs. However, existing works only consider a special scenario: one ground vehicle with multiple UAVs. In this paper, we consider a more general scenario: multiple ground vehicles with multiple UAVs. We formalize the multi-vehicle-assisted multi-UAV path planning problem, which is a joint route planning and task assignment problem (RPTSP). To solve RPTSP, an efficient multi-vehicle-assisted multi-UAV path planning algorithm (MVP) is proposed. In MVP, we first allocate the detecting points to proper parking spots and then propose an efficient heuristic allocation algorithm EHA to plan the paths of ground vehicles. Besides, a genetic algorithm and reinforcement learning are utilized to plan the paths of UAVs. MVP maximizes the profits of an MCS carrier with a response time constraint and minimizes the number of employed vehicles. Finally, performance evaluation demonstrates that MVP outperforms the baseline algorithm.

1. Introduction

In recent years, due to the massive increase in sensor-rich mobile devices, mobile crowdsensing (MCS) [1] has emerged as a new way of sensing, which relies on a crowd of personal mobile phones, tablet computers, and other smart gadgets to perform large-scale tasks. While traditional sensing technologies incur large overheads due to the deployment of lots of sensors, MCS just needs to pay some incentive rewards to attract individuals to perform sensing tasks, which is obviously more cost-effective. Therefore, MCS has been used in a lot of valuable applications recently, such as detecting air quality and collecting traffic information [2, 3].

In addition, tremendous progress in the research of microelectromechanical systems has enabled UAVs to enter the civilian market. Since UAVs are economical, flexible, and easy to operate, they have been widely used in agriculture, geological exploration, military, and other fields [4–6]. Due to the high mobility and fast deployment, UAVs can collect various data anywhere and anytime when equipped with rich sensors. They could also be used to collect data in regions where ground vehicles are difficult to reach, e.g., flood hazard areas. With the increasing popularity of UAVs, more and more researchers began to introduce UAVs into the MCS to get better performance.

Despite the mentioned benefits, the hovering time of UAVs is quite constrained by limited battery capacity, which prevents them from serving a wide area. To solve the problem, in practice, ground vehicles are utilized to transport UAVs to collect data. In addition, UAVs will fly back to a ground vehicle to charge themselves after completing sensing tasks. The so-called vehicle-assisted UAV sensing benefits from both the long driving distance of the vehicle and high flexibility of UAVs [7].

After introducing the ground vehicle, efficient path planning and scheduling of drones and ground vehicle become a key issue. There are lots of researches that are dedicated to optimizing the routing and scheduling of vehicle-assisted UAVs for the transporting of parcels [8–10], wherein vehicles can also visit the customers to deliver parcels. These works are inappropriate for the vehicle-drone cooperative sensing problem studied in this paper, wherein a vehicle is only used to transport UAVs. In the meantime, only few researches studied the routing of the vehicle-drone cooperative sensing system [11, 12]. [11] assumes that the vehicle has only one drone, and [12] considers one vehicle with multiple drones and proposes algorithms to collect sensing data in multiple target points simultaneously.

However, all the existing researches suppose that there is only one ground vehicle. In real world, we need to employ multiple vehicles with multiple drones to perform sensing tasks simultaneously. Compared with relying on only one ground vehicle, using multiple vehicles can effectively improve the efficiency of the MCS system.

When involving multiple vehicles, path planning of vehicles and drones becomes a more complex problem, which cannot be converted into a classic optimization problem. In this paper, we formalize the joint route planning and task assignment problem (RPTSP). To simplify RPTSP, we divide it into two subproblems: task assignment problem and path planning problem. In our scenario, the second subproblem includes multivehicle path planning and multi-UAV path planning. We propose an efficient heuristic allocation algorithm EHA to determine the paths of multiple vehicles, which actually solves the task assignment problem and multivehicle path planning problem. EHA utilizes an iterative process to assign tasks for each ground vehicle based on the expenses incurred and the time consumed. Besides, to solve the multi-UAV path planning problem, we transformed it into a Multiple Traveling Salesmen Problem (MTSP). After that, the genetic algorithm and reinforcement learning are utilized to solve MTSP. In general, our goal is to maximize the profits of the MCS carrier with a time budget by globally optimizing the assignment of tasks and the route of vehicles and UAVs. Besides, we hope that the number of employed ground vehicles can be minimized.

To the best of our knowledge, we are the first to consider the task assignment and the routing problem for multi-vehicle-assisted multi-UAVs in MCS. The contributions of this work are summarized as follows:

(i) We introduce multiple vehicles to the vehicle-drone cooperative sensing system and formalize the joint route planning and task assignment problem (RPTSP).

(ii) To solve RPTSP, we propose a multi-vehicle-assisted multi-UAV path planning algorithm (MVP), which maximizes the profits of the MCS carrier with a response time constraint

(iii) Extensive experiments are conducted, and the results show that MVP outperforms the baseline algorithm

The remainder of this paper is organized as follows. Section 2 gives an overview of the existing work related to the problem that we are addressing. Section 3 describes the system model and problem formulation. Section 4 illustrates the details of MVP. Section 5 introduces the simulation experiment. Section 6 concludes this paper.

2. Related Work

Recently, a lot of researches have been conducted on MCS. These works take different methods to perform sensing tasks in different application scenarios: (1) the MCS system relies on individuals' smart devices to perform sensing tasks, (2) it utilizes one vehicle and one drone to perform sensing tasks, (3) it utilizes one vehicle and multiple drones to perform sensing tasks. In the following, we will describe these works.

2.1. MCS Utilizing the Smart Devices. For traditional MCS that relies on the smart gadgets possessed by individuals, there have been many studies on assigning sensing tasks to participants.

These studies have different optimization objectives; e.g., He et al. [13] devised an efficient local ratio-based algorithm to maximize the benefits of the MCS carrier under a time budget constraint; Xiong et al. [14] introduced an incentive allocation framework to minimize total incentive payment while ensuring predefined spatial-temporal coverage; and Shi et al. [15] designed a crowdsensing task assignment mechanism to maximize the task completion rate under a predefined incentive budget.

2.2. MCS Utilizing One Vehicle and One Drone. When evolving towards MCS architectures consisting of UAVs and vehicles, route planning should be well designed to minimize the consumed time or rewards in performing sensing tasks.

Chen et al. [16] designed a trajectory segment selection scheme to remove data redundancy and improve the coverage quality. Luo et al. [11] proposed two heuristic algorithms to solve the two-echelon cooperated routing problem for the ground vehicle and its carried drone. Savuran and Karakaya [17] proposed a path optimization method that allows the vehicle to keep moving when the drone is performing tasks. However, the above works all consider the case of only one drone, which makes it impossible for vehicle-assisted UAVs to perform multiple tasks in parallel.

2.3. MCS Utilizing One Vehicle and Multiple Drones. As for the case of using multiple UAVs, Hu et al. [18] proposed a vehicle-assisted multi-UAV routing and scheduling algorithm (VURA). It works by iteratively deriving solutions based on UAV routes picked from the memory that contains flight paths of drones. Through continuous joint optimization of parking spot selection, path planning, and tour assignment, VURA can produce a final appropriate solution. In [12], Hu et al. proposed a novel algorithm (VAMU) based on VURA, which schedules the multiple drones to be launched and recycled in different places. It avoids the time wastage when the vehicle waits for drones to return and thus reduces the time required to complete all tasks.

However, the works mentioned above did not take multiple vehicles into account. In reality, the MCS carrier needs to employ multiple vehicles with multiple drones to perform sensing tasks simultaneously, which significantly improves the efficiency of the MCS system. Therefore, this motivates us to consider a more general scenario, in which multiple vehicles and multiple drones are used to perform sensing tasks allocated by the MCS system.

3. System Model and Problem Formulation

3.1. System Model. In our model, there are several ground vehicles with different numbers of UAVs. A set of detecting points that UAVs need to visit and collect data are distributed in a target region. Every detecting point needs to be visited once only by a UAV. And a UAV can visit multiple detecting points (i.e., perform sensing tasks) consecutively in a single flight. Due to the limited battery capacity, UAVs cannot serve a wide area. To solve the problem and maximize the efficiency in the meantime, this paper employs multiple ground vehicles to assist UAVs.

As shown in Figure 1, with the aid of ground vehicles, the UAVs can visit detecting points distributed in a very large region. From the starting point, a ground vehicle transports drones to the preselected parking spots sequentially. Once a ground vehicle arrives at a parking spot, the UAVs carried by it are released to perform sensing tasks in nearby detecting points. When the UAVs finish their missions of one single trip, they return to the corresponding ground vehicle to be charged. Notice that it is possible for a UAV to be launched and recycled multiple times in each parking spot. When all detection points near the parking spot have been visited by UAVs, the ground vehicle with multiple UAVs leaves to the next selected parking spot. Once all detecting points are visited, the whole sensing task in the target region is finished.

The consumed time and the generated incentive rewards must be considered when the MCS system assigns sensing tasks and plans the route of vehicles as well as UAVs.

For simplicity, we make the following hypothesis:

(i) The MCS system knows the coordinates of each detecting point

(ii) Each detecting point should be accessed once

(iii) The time consumed by the UAV to perform the sensing task in a detecting point is constant

(iv) Both the ground vehicle and UAVs travel at a constant speed [12, 18]

3.2. Problem Formulation. Let $G = \{V, E\}$ be an undirected graph where V is the set of vertex and E is the set of edge. And V is divided into $V_d = \{d_1, d_2, \cdots, d_{N_d}\}$ and $V_p = \{p_1, p_2, \cdots, p_{N_p}\}$, which represent the set of detecting points and the set of candidate parking spots, respectively. N_d represents the number of detecting points, and N_p represents the number of parking spots. The distance of point n_1 and n_2 is denoted by $\mathrm{dis}(n_1, n_2)$, where the points n_1 and n_2 represent the detecting point or parking spot.

Ground vehicles we employed are expressed as $\mathrm{GV} = \{\mathrm{gv}_1, \mathrm{gv}_2, \cdots, \mathrm{gv}_{N_{\mathrm{gv}}}\}$, where N_{gv} represents the total number of ground vehicles. And $U_{\mathrm{gv}_i} = \{u_1, u_2, \cdots, u_{m_i}\}(\mathrm{gv}_i \in \mathrm{GV})$ denotes the set of UAVs that the vehicle gv_i possesses, where m_i represents the number of drones owned by vehicle gv_i. $V_{\mathrm{gv}_i} = \{d_{k_1}, d_{k_2}, \cdots, d_{k_i}, \cdots, d_{k_n}\}$, where $d_{k_n} \in V_d$ denotes the detecting points that the UAVs possessed by gv_i need to access. Besides, we use V_{vehicle} and V_{uav} to represent the speed of the vehicles and the UAVs, respectively. D_{u_i} represents the maximum flight distance of UAV u_i. In this paper, the main objective is to jointly optimize the sensing task assignment and the path planning of ground vehicles and UAVs such that the incentive rewards produced are minimized with a time budget constraint that is denoted as $\mathrm{Time}_{\mathrm{budget}}$.

We need to address the following issues: first, we should think out how to select the proper parking points from the roads lying on the target region. There are infinite points that can be selected as candidate parking spots in the road network. What we need to do is to select some points as parking spots from this infinite number of points and assign detecting points to these parking spots. The detecting points that are assigned to parking spot p_j are represented as $V_{p_j} = \{d_{k_1}, d_{k_2}, \cdots, d_{k_i}, \cdots, d_{k_n}\}(p_j \in V_p, d_{k_i} \in V_d)$. It should be guaranteed that when the vehicle is parked at these points, the drones carried on the vehicle can access all the detecting points.

Then, we need to plan the flight paths of UAVs when a ground vehicle arrives at parking spot p_j. In our scenario, each drone performs a trip by visiting the detecting points along its route sequentially. Since we employ N_u UAVs to perform tasks in parallel, the flight paths of UAVs possessed by ground vehicle gv_i can be denoted as $\mathrm{FP}_{p_j}^{\mathrm{gv}_i} = \{R_{p_j}^{u_1}, R_{p_j}^{u_2}, \cdots, R_{p_j}^{u_{m_i}}\}$ when the ground vehicle is parking at the parking spot p_j. In detail, the flight path of $u_k (1 \le k \le m_i)$ at parking spot p_j is expressed as $R_{p_j}^{u_k} = \{a_1, a_2, \cdots, a_z\}(a_z \in V_{p_j})$.

We use $C(R_{p_j}^{u_k}, a_z)$ to denote whether the route $R_{p_j}^{u_k}$ contains the detecting point a_z, where $R_{p_j}^{u_k} \in \mathrm{FP}_{p_j}$ and $a_z \in V_{p_j}$. In detail, $C(R_{p_j}^{u_k}, a_z) = 1$ means that the detecting point a_z is included in the route $R_{p_j}^{u_k}$. Furthermore, $\mathrm{Adj}_{R_{p_j}^{u_k}}(d_a, d_b)$ $(d_a, d_b \in V_{p_j})$ is used to express whether the points d_a and d_b are adjacent in the route $R_{p_j}^{u_k}$. For example, $\mathrm{Adj}_{R_{p_j}^{u_k}}(d_a, d_b) = 1$ indicates that d_a and d_b are adjacent in the route $R_{p_j}^{u_k}$. Then, the total flight distance of the UAV u_k at the parking spot p_j can be defined as follows:

$$l\left(R_{p_j}^{u_k}\right) = \sum_{d_a \in R_{p_j}^{u_k}} \sum_{d_b \in R_{p_j}^{u_k}} \mathrm{dis}(d_a, d_b) \mathrm{Adj}_{R_{p_j}^{u_k}}(d_a, d_b). \qquad (1)$$

Based on the above equation, the total distance that all UAVs of the ground vehicle gv_i travel at the parking spot p_j is given as follows:

$$l\left(\mathrm{gv}_i, p_j\right) = \sum_{r_z \in \mathrm{FP}_{p_j}^{\mathrm{gv}_i}} l(r_z). \qquad (2)$$

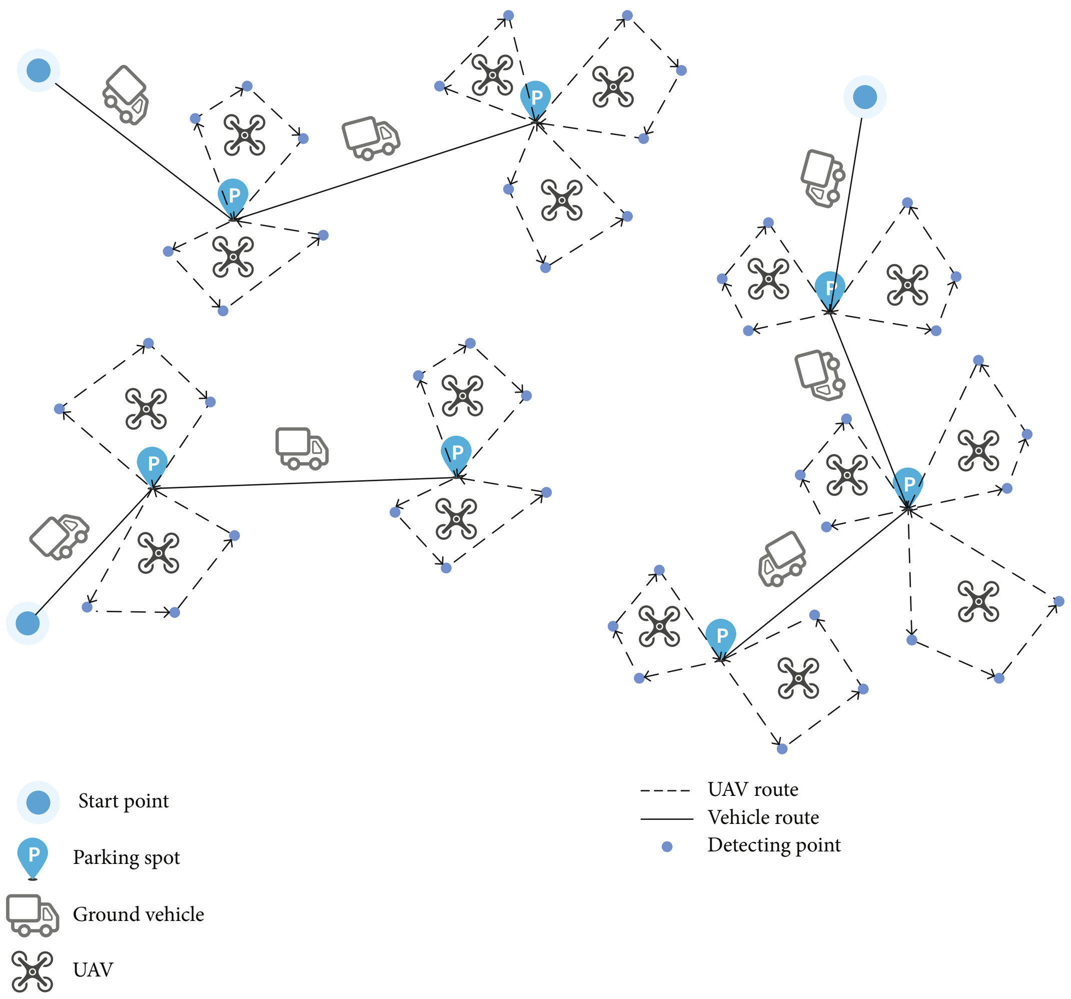


Figure 1: System model.

Besides, a sorted set of parking spots needs to be determined for each vehicle to minimize the generated incentive rewards with a time budget constraint. We use $R_{\mathrm{gv}_i} = \{p_{k_i}, p_{k_2}, \cdots, p_{k_i}, \cdots, p_{k_n}\}$ where $p_{k_i} \in V_p$ to represent the route for the ground vehicle gv_i. We use $\mathrm{Adj}_{R_{\mathrm{gv}_i}}(p_a, p_b)$ to denote whether the points p_a and p_b are adjacent in the vehicle route R_{gv_i} $(\mathrm{Adj}_{R_{\mathrm{gv}_i}}(p_a, p_b) = 1)$ or not $(\mathrm{Adj}_{R_{\mathrm{gv}_i}}(p_a, p_b)=0)$. Then, the travel distance of the ground vehicle gv_i is expressed as follows:

$$l(gv_i) = \sum_{p_a \in R_{\mathrm{gv}_i}} \sum_{p_b \in R_{\mathrm{gv}_i}} \mathrm{dis}(p_a, p_b)\mathrm{Adj}_{R_{\mathrm{gv}_i}}(p_a, p_b). \tag{3}$$

Since the distance of UAVs and the vehicle that carries them is calculated, we can easily get the time $\mathrm{Time}(\mathrm{gv}_i)$ consumed by vehicle gv_i with its UAVs in the whole sensing task:

$$\mathrm{Time}(\mathrm{gv}_i) = \frac{l(\mathrm{gv}_i)}{V_{\mathrm{vehicle}}} + \sum_{p_j \in R_{\mathrm{gv}_i}} \frac{l\left(\mathrm{gv}_i, p_j\right)}{V_{\mathrm{uav}}}. \tag{4}$$

Thus, the total time $\mathrm{Time}_{\mathrm{final}}$ cost by all vehicles and UAVs is defined as follows:

$$\mathrm{Time}_{\mathrm{final}} = \max\left(\mathrm{Time}(\mathrm{gv}_1), \mathrm{Time}(\mathrm{gv}_2), \cdots, \mathrm{Time}(\mathrm{gv}_i)\right). \tag{5}$$

Finally, the calculation of the incentive rewards generated throughout the execution of all sensing tasks should also be confirmed. In this paper, for a ground vehicle, its revenue depends on the total flight distance of its UAVs and the travel distance of its ground vehicle. Therefore, the final incentive rewards earned by a ground vehicle are calculated as follows:

$$\mathrm{IR}(\mathrm{gv}_i) = \mathrm{IR}_{\mathrm{base}} + \alpha l(\mathrm{gv}_i) + \beta \sum_{p_j \in R_{\mathrm{gv}_i}} l\left(\mathrm{gv}_i, p_j\right), \tag{6}$$

where $\mathrm{IR}_{\mathrm{base}}$ represents the base incentive rewards that a ground vehicle can get if it has accepted the sensing tasks and completed assigned tasks. α and β are the price coefficients for the UAV and the vehicle, respectively, which are used to transform distance to the price.

Table 1: Notation and terminology.

Notation	Definition
V_p	The set of parking spots
V_d	The set of detecting points
N_p	The number of parking spots
N_d	The number of detecting points
$\mathrm{dis}(n_1, n_2)$	The distance between two different points
GV	The set of ground vehicles
U_{gv_i}	The set of UAVs that the vehicle gv_i possesses
N_{gv}	The number of ground vehicles
V_{gv_i}	The set of detecting points assigned to the vehicle gv_i
V_{p_j}	The set of detecting points assigned to the parking spot p_j
V_{vehicle}	The speed of ground vehicles
V_{uav}	The speed of UAVs
D_u	The max flight distance of UAVs
$\mathrm{Time}_{\mathrm{budget}}$	The budget of time
m_i	The number of drones of the vehicle gv_i
$R_{p_j}^{u_k}$	The flight path of u_k at the parking spot p_j
$\mathrm{FP}_{p_j}^{\mathrm{gv}_i}$	The flight paths of drones possessed by vehicle gv_i at the parking spot p_j
$C\left(R_{p_j}^{u_k}, a_z\right)$	Binary indicating whether the route $R_{p_j}^{u_k}$ contains the detecting point a_z
$\mathrm{Adj}_{R_{p_j}^{u_k}}(d_a, d_b)$	Binary indicating whether d_a and d_b are adjacent in the route $R_{p_j}^{u_k}$
$l\left(\mathrm{gv}_i, p_j\right)$	The total distance that all drones of the vehicle gv_i travels at the parking spot p_j
R_{gv_i}	The route of the vehicle gv_i
$l(t_i)$	The travel distance of the vehicle gv_i
$\mathrm{Time}(t_i)$	The time consumed by the vehicle gv_i with its drones in the whole sensing mission
$\mathrm{IR}(\mathrm{gv}_i)$	The total incentive rewards earned by the vehicle gv_i with its drones

In our scenario, for the MCS carrier, the income is proportional to the number of detection points. Besides, for an actual sensing task, the income is a fixed value, which means that maximizing profits equals minimizing expenses. Then, our objective can be transformed to minimize the generated incentive rewards. The total incentive reward cost, expressed as $\mathrm{Cost}(s)$, includes all the revenues of each ground vehicle in the solution. The algorithm proposed in this paper is aimed at minimizing the total incentive reward consumption.

$$\min \mathrm{Cost}(s) = \sum_{\mathrm{gv}_i \in \mathrm{GV}} \mathrm{IR}(\mathrm{gv}_i), \tag{7}$$

$$N_t \geq 1, \tag{8}$$

$$m_i \geq 1, \forall i \in [1, N_t], \tag{9}$$

$$0 \leq l\left(R_{p_j}^{u_k}\right) \leq D_u, \tag{10}$$

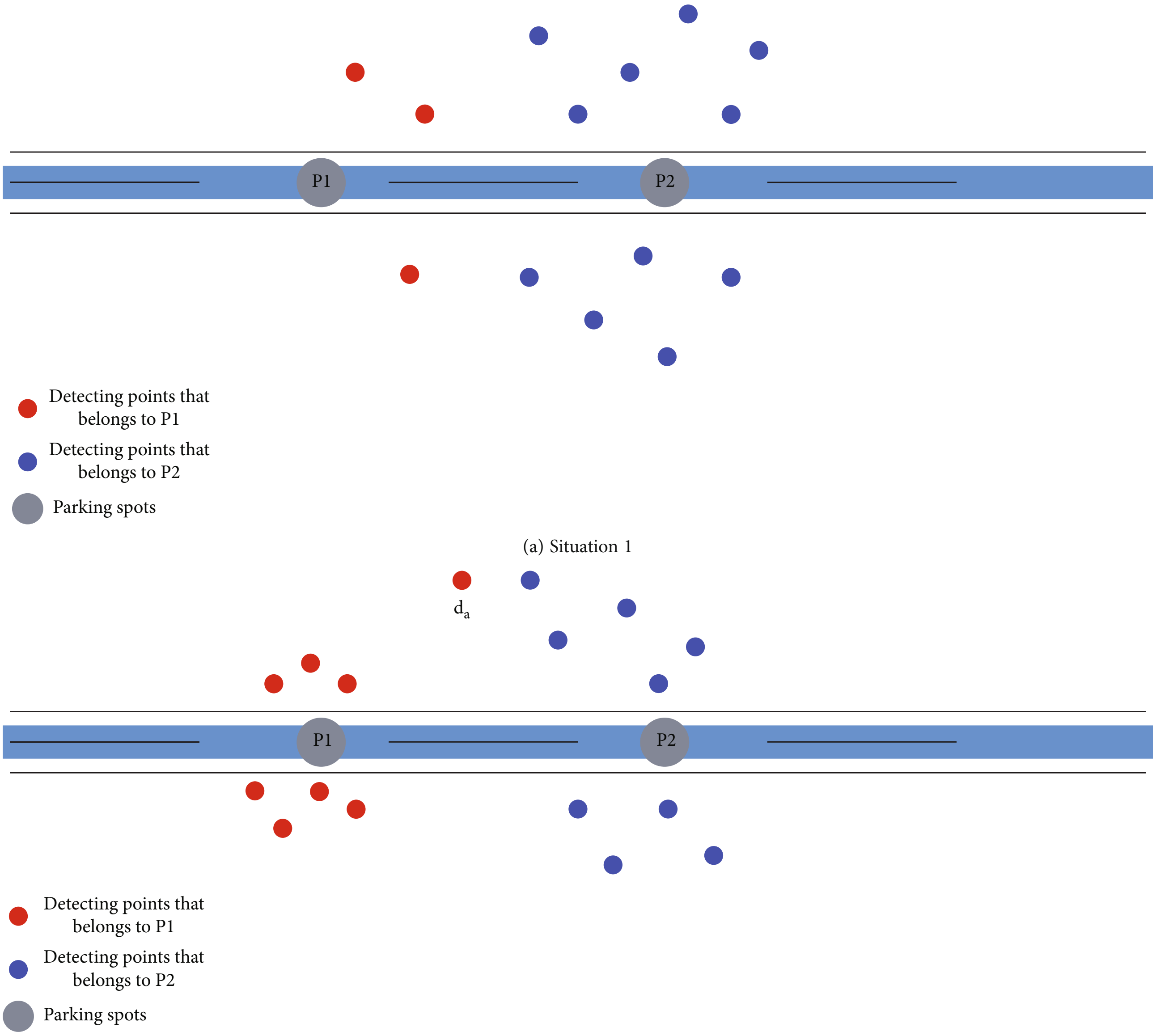
$$0 \leq \mathrm{Time}_{\mathrm{final}} \leq \mathrm{Time}_{\mathrm{budget}}, \tag{11}$$

$$\sum_{R \in \mathrm{FP}_{p_j}^{\mathrm{gv}_i}} \sum_{a_z \in V_{p_j}} C(R, a_z) = \left| V_{p_j} \right|, \forall p_j \in V_p. \tag{12}$$

Constraint (8) confirms that at least one ground vehicle is employed. Constraint (9) emphasizes the requirement that every ground vehicle has at least one drone. Constraint (10) means that the flight distance of a UAV should be less than the maximum flight distance. Constraint (11) indicates that the time consumption in a solution should not exceed the time budget. Constraint (12) ensures that all detecting points of each parking spot are visited. The notation and terminology used throughout the paper are shown in Table 1.

4. Algorithm Design

In this section, we first introduce some challenges encountered in designing the efficient solution. Then, we propose


P1
P2
Detecting points that belongs to P1
Detecting points that belongs to P2
Parking spots
d_a
P1
P2
Detecting points that belongs to P1
Detecting points that belongs to P2
Parking spots

(a) Situation 1

(b) Situation 2

Figure 2: The situations of allocating detecting points.

MVP to solve these problems. After that, we provide a detailed description of each part of MVP.

4.1. The Challenges. In order to complete the whole sensing mission effectively, we need to solve the following problems:

(1) How to allocate each detecting point to a proper parking spot? In our scenario, the detecting points are randomly distributed in a large target region, and the parking spots are sampled from the points of the road network. To reduce the distance between the detection point and the parking point, for each detecting point, a naive method is to allocate it to the closest candidate parking spot, which is selected every other distance in the roads. However, the shortest distance between parking spots and detecting points does not mean the smallest time cost for the UAVs to visit those detecting points. It is illustrated by the following examples

First, as shown in Figure 2(a), some parking spots possess few detecting points. When ground vehicles traverse these points, too few detecting points will cause some drones to sit idle. This will incur unnecessary time wastage. Besides, as shown in Figure 2(b), the detecting point d_a is assigned to the parking spot p_1 due to the shortest distance between them. However, other detecting points assigned to the parking spot p_1 are far from d_a. Thus, it will incur a relatively large time cost when a UAV traverses $p_{1'}s$ detecting points.

(2) How to allocate the parking spots to ground vehicles? Unlike previous studies, we employ multiple vehicles in the meantime. We transform the problem of assigning sensing tasks into the parking spot

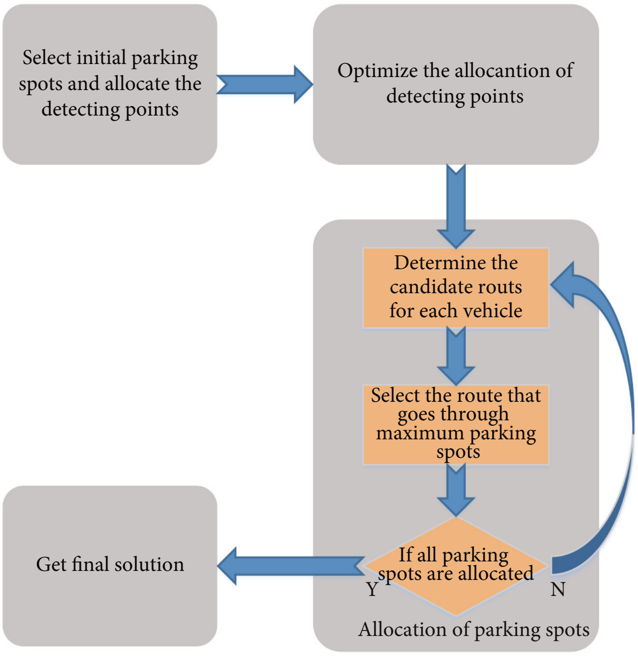

Select initial parking spots and allocate the detecting points
Optimize the allocantion of detecting points
Determine the candidate routs for each vehicle
Select the route that goes through maximum parking spots
If all parking spots are allocated
Y
N
Get final solution
Allocation of parking spots

FIGURE 3: The overall architecture of MVP.

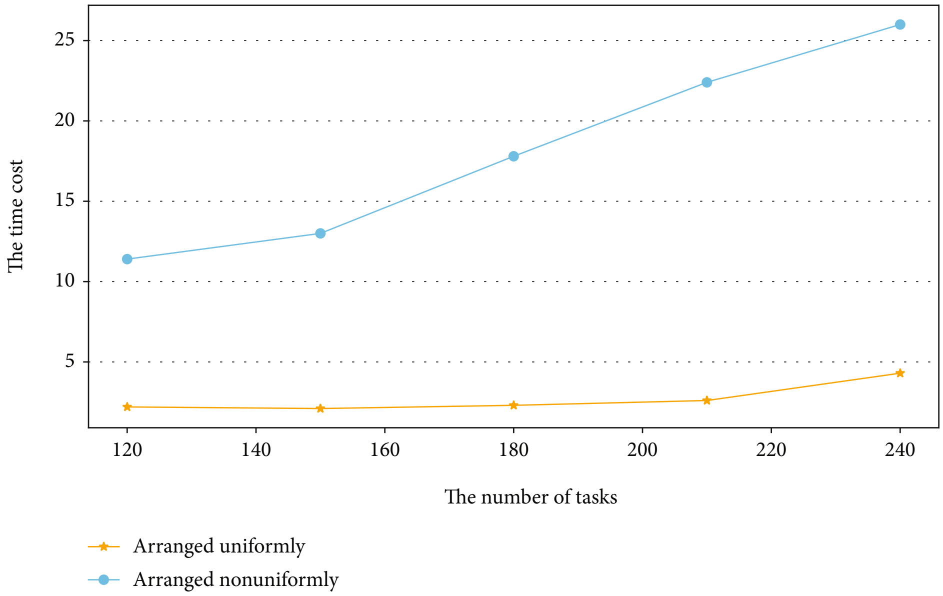


FIGURE 4: The time cost in nonuniform environment.

allocation problem. This problem can be defined as follows: given a set of parking spots, a set of detecting points, and the corresponding relation between them, we need to determine the route for each employed vehicle to minimize the produced incentive rewards with a time budget constraint. Notice that each parking spot can be visited exactly once by only one ground vehicle, and the number of employed ground vehicles is not determined until all parking spots are allocated

(3) How to plan the paths of UAVs when a ground vehicle arrives at a parking spot? After the allocation of the detecting points, we know exactly the corresponding detecting points of each parking spot. When a ground vehicle arrives at a parking spot, we should plan the paths of UAVs. Since every vehicle possesses multiple drones, the problem can be transformed into the multiple travelling salesman problem (MTSP). The MTSP in our scenario is illustrated as follows: given a parking spot and a set of

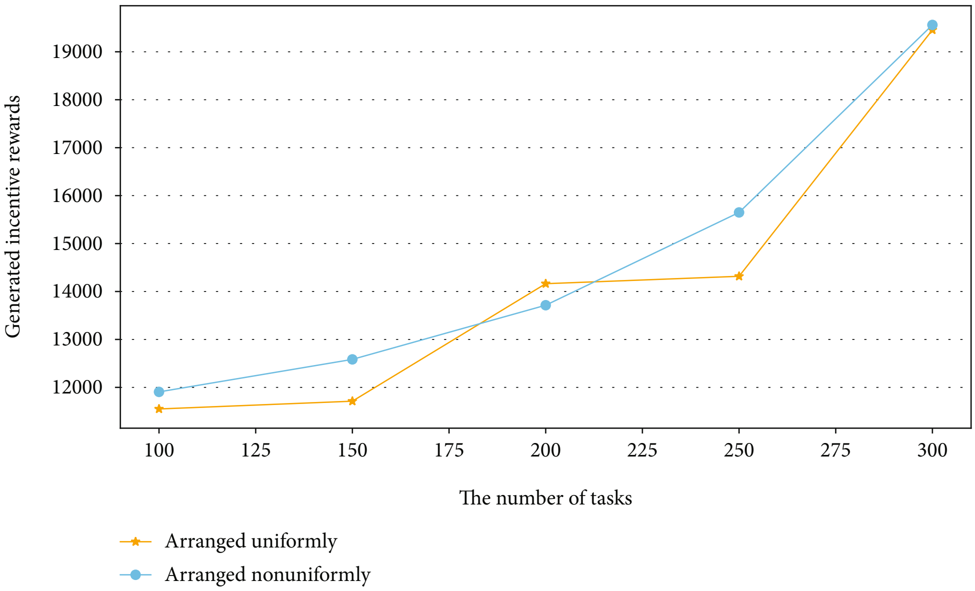


Figure 5: The impact of the number of tasks for different methods of selecting candidate parking spots.

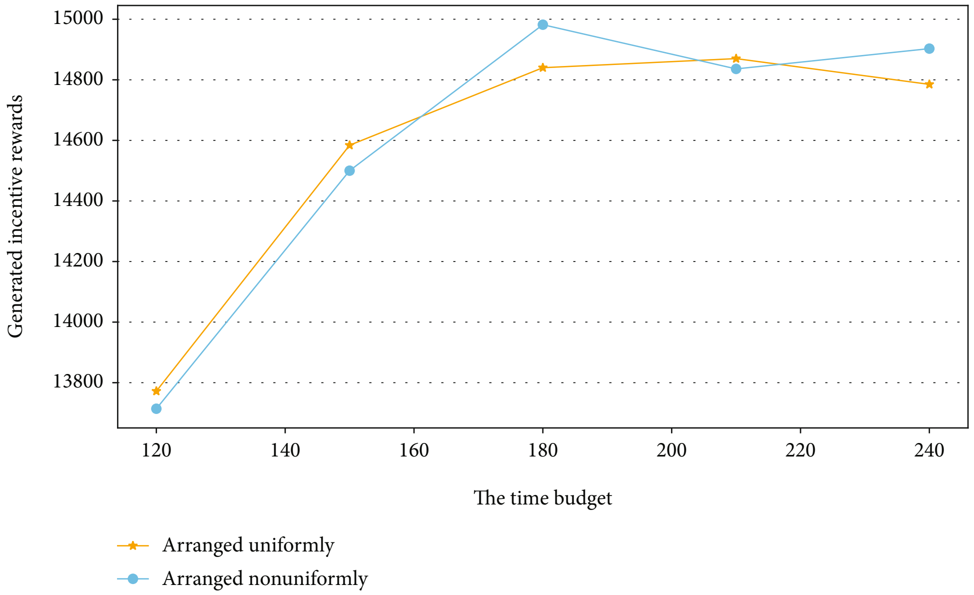


Figure 6: The impact of the time budget for different methods of selecting candidate parking spots.

detecting points allocated to it, we need to find out a flight path for each drone such that the total cost of time is minimized and that each detecting point is visited exactly once by only one drone

4.2. The Overall Architecture of MVP. The overall architecture of MVP is shown in Figure 3. First, we mark a point on roads every other distance to construct the set of initial parking spots. After that, the detecting points are clustered by allocating them to different parking spots according to their distance to the parking spots. Then, in order to improve the performance of MVP, several optimizations are used to reallocate detecting points. Subsequently, parking spots are allocated to ground vehicles in an iterative process. In each iteration, we can determine the route of one ground vehicle $\mathrm{gv}_i(\mathrm{gv}_i \in \mathrm{GV})$ and the paths of $\mathrm{gv}_{i'}s$ UAVs at each parking spot. The route of gv_i is represented as $R_{\mathrm{gv}_i} = \{p_{k_1}, p_{k_2}, \cdots, p_{k_i}, \cdots, p_{k_n}\}(p_{k_i} \in V_p)$, and the paths of $\mathrm{gv}_{i'}s$ UAVs at parking spot p_i are represented as $\mathrm{FP}_{p_j}^{\mathrm{gv}_i} = \{R_{p_j}^{u_1}, R_{p_j}^{u_2}, \cdots, R_{p_j}^{u_k}, \cdots, R_{p_j}^{u_{m_i}}\}$, where $R_{p_j}^{u_k}(1 \le k \le m_i)$ denotes the flight path of u_k at the parking spot p_j. The ground vehicle gv_i will access the parking spots in the route R_{gv_i} sequentially. Besides, gv_i will release its UAVs to visit the detecting points along the paths in $\mathrm{FP}_{p_j}^{\mathrm{gv}_i}$ when it arrives at the parking spot p_j. When the iterative process ends, we get the routes of all ground vehicles and the paths of their UAVs at each parking spot, which ensures that all detecting points in the target region have been accessed.

4.3. Parking Spot Selection. The selection of the parking spots for detecting points is divided into two steps: the initialization of the parking spots and their optimization. The details of the steps are described as follows.

4.3.1. Initialization. First, a set of candidate parking spots is determined by sampling points on roads every other distance. Then, for each detecting point, we calculate the distance between it and its nearby candidate parking spot. After that, the nearest candidate parking spot of each detecting point is determined. These parking spots are composed of the initial set of parking spots. Besides, there is another method to mark the candidate parking spots when detecting points are distributed nonuniformly. It arranges the candidate parking spots nonuniformly. The process is described as follows:

(1) For each of the roads, we first mark the endpoint p_1 as a candidate parking spot

(2) Then, we calculate the density of detecting points around p_1. If the density is big, we mark the next candidate parking spot at a location close to the p_1. If the density is small, we mark the next candidate parking spot at a location far from p_1

(3) After that, we calculate the density of detecting points around the second candidate parking spot and repeat the process above to mark all candidate parking spots of the road

We have performed some experiments to compare these two methods. The results of the experiments are shown in Figures 4–6.

4.3.2. Optimization. Choosing the nearest candidate parking spot is not always the best strategy. Therefore, in order to get better performance, we need to optimize the allocation of detecting points.

(a) As shown in Figure 2(a), there are many selected parking spots that have few detecting points. It will incur unnecessary time cost if the ground vehicle parks at these parking spots and releases its UAVs to access corresponding detecting points. Therefore, the detecting points assigned to these parking spots should be reallocated. In detail, for a selected parking spot p_j, if the number of p_j's detecting points is less than a predefined value N_m, we will remove p_j from the set of parking spots and reallocate each detecting point of p_j to the second closest parking spot to it. The procedure will be repeated until all selected parking spots have at least N_m detecting points. In the experimental part, we will investigate the impact of N_m's value by performing some experiments

```
Input The set of initial parking spots $V_p$,
the set of corresponding detecting points $V$.
Ouput NULL
for each $p_i \in V_p$ do
    if $|V_{p_i}| < N_m$ then
reallocate each point in $V_{p_i}$ to the second
nearest parking spot to it;
$V_p = V_p \setminus p_i$
    end
    else
        for each $d_i \in V_{p_i}$ do
        calculate the distance $D_r$ between $d_i$
      and its nearest detecting point in $V_{p_i}$;
        find $d_i$'s second nearest parking spot $p_k$
          if $\exists d_k \in V_{p_k}$ is inside $Cir(d_i, D_r)$ then
          reallocate $d_i$ to $p_k$;
          end
          end
      end
end
```

Algorithm 1: Reallocate the detecting points.

(b) As for the situation shown in Figure 2(b), it is obviously a better choice to allocate the detecting point d_a to parking spot P_2 as it avoids the waste of flight time of drones. Therefore, the detecting point in such a situation should be reallocated. We use $\mathrm{Cir}(c, r)$ to denote the circle with c as the center and r as the radius. Besides, V_{p_j} represents the detecting points of the parking spot p_j. For a detecting point d_i of a parking spot p_i, we suppose that d_k is the nearest detecting point to it in V_{p_i} and the distance between d_i and d_k is D_r. Furthermore, p_k is supposed to be the second nearest parking spot to d_i. If there is at least one detecting point in V_{p_k} that is inside the circle $\mathrm{Cir}(d_i, D_r)$, the detecting point d_i will be reallocated to p_k

The pseudocode of the reallocation procedure is shown in Algorithm 1.

4.4. Allocation of Parking Spots. After the selection of parking spots, for each parking spot, we allocate it to a proper ground vehicle. The travel route of each ground vehicle and the flight paths of corresponding drones at each parking spot are determined in the procedure of allocation. In a word, we will get the final solution of RPTSP when the procedure ends. An efficient heuristic allocation algorithm EHA is proposed to effectively allocate the parking spots.

In the procedure of allocation, for a ground vehicle gv_i, we need to find the K nearest neighbor parking spots to g v_i. A naive method is to calculate the distances between gv_i and all parking spots, which obviously incurs a large amount

```
Input The set of selected parking spots $V_p$,
the set of corresponding detecting points $V$,
the set of candidate vehicles $GV$ and its UAVs $U$
Output The final solution $S_{final}$
1 $S_{final} \longleftarrow \varnothing$;
2 build R-tree $RT$ using the points in $V_p$
3 do
4   for $n = 1 \longrightarrow |GV|$
5     select a vehicle $gv_i$ from $GV$;
6       $R_{gv_i}, R_{gv_i}^{uav} \longleftarrow \varnothing$;
7             $Time_{gv_i} = 0$;
8       do
9       search $gv_{i'}s$ nearby parking spots $P$ according to $RT$;
10        $IR_{\min} = MaxNumber$;
11        $R_{temp} \longleftarrow \varnothing$;
12 for each $p_j \in P$ do
13 $R_{p_j} \longleftarrow$ utilize GA to plan the paths of $U_{gv_i}$ at $p_j$;
14 calculate the $l(gv_i, p_j)$ based on $R_{p_j}$;
15 calculate the current $l(gv_i)$ when visit $p_j$;
16 $IR_{temp} = IR_{base} + \alpha l(gv_i) + \beta \sum_{p_j \in R_{gv_i}} l(gv_i, p_j)$;
17        if $IR_{temp} < IR_{\min}$ then
18        $IR_{\min} = IR_{temp}$;
19 $R_{temp} = R_{p_j}$;
20        $p_{temp} = p_j$;
21        $Time_{temp} = (l(gv_i, p_j)/V_{uav}) + (l(gv_i)/V_{vehicle})$
22 end
23 end
24 $R_{gv_i}^{uav} = R_{gv_i}^{uav} \cup R_{temp}$;
25 $R_{gv_i} = R_{gv_i} \cup p_{temp}$;
26 $Time_{gv_i} = Time_{gv_i} + Time_{temp}$;
27 $Time_{gv_i} = Time_{budget}$;
28      $GV = GV \setminus gv_i$
29   end
30 select the vehicle $gv_k$ with maximum $|R_{gv_k}|$
31 $S_{final} = S_{final} + \{R_{gv_k}, R_{gv_k}^{uav}\}$
32 $V_p = V_p \setminus R_{gv_k}$;
33 while $V_p \neq \varnothing$
34 return $S_{final}$
```

Algorithm 2: Plan the paths of ground vehicles.

of time consumption. To improve the performance, in EHA, R-tree [19] is introduced to index the locations of all parking spots. After that, we utilize the branch-and-bound R-tree traversal algorithm proposed in [20] to find the K nearest neighbor parking spots to gv_i.

The details of EHA are described in Algorithm 2. EHA starts with building R-tree RT, which utilizes the locations of all parking spots (line 2). Then, it chooses a vehicle gv_i (line 5) and iterates to generate gv_i's candidate routes until the consumed time exceeds the time budget constraint (lines 8-27). In each iteration, we search RT to get $\mathrm{gv}_{i'}s$ nearby parking spots P (line 9), from which the parking spot that produces minimum incentive rewards is selected (lines 12-23). The procedure is repeated until all vehicles' candidate routes have been determined (lines 4-29). Then, we choose the vehicle gv_k with maximum $|R_{\mathrm{gv}_k}|$ and add $\mathrm{gv}_{k'}s$ corresponding routes $\{R_{\mathrm{gv}_k}, R_{\mathrm{gv}_k}^{\mathrm{uav}}\}$ into the solution S_{final} (lines 30-31). The above process will keep looping until all parking spots have been allocated (lines 3-33).

Algorithm 2 is designed to minimize the generated incentive rewards with a time budget constraint. In the meantime, it minimizes the number of employed ground vehicles.

4.5. Path Planning of UAVs. When a ground vehicle arrives at a parking spot, the paths of its UAVs need to be planned. Notice that the path planning of UAVs is involved in the

```
Input The parking spot $p_j$,
the set of corresponding detecting points $V_{p_j}$,
the UAVs $U_{t_i}$
Output Optimal routes of UAVs $routes$
1 generate initial population $Pop$,
$|Pop| = populationSize$;
2 for $n = 1 \longrightarrow iteration$ do
3       calculate the fitness of each individual in $Pop$;
4       $newPop \longleftarrow \varnothing$;
5       for $m = 1 \longrightarrow populationSize$ do
6           select two individuals from $Pop$ according
to the fitness function
7       if $random(0, 1) < P_c$ perform crossover to generate a child
8       $R_{temp}$
9       end
10          if $random(0, 1) < P_m$
11        perform mutation on $R_{temp}$;
12        end
13        $newPop = newPop \cup R_{temp}$;
14        end
15        $pop = newPop$;
16 end
17 select the individual $routes$ in $pop$ with maximum fitness;
18 return $routes$;
```

Algorithm 3: Plan the paths of UAVs.

procedure of allocating parking spots for ground vehicles, as shown in Algorithm 2. As mentioned before, this problem can be transformed into the MTSP, which is a typical NP-hard problem. There have been some works that concern on the trajectory scheduling of mobile vehicles [21–23]. In this paper, we adopt the genetic algorithm (GA) and reinforcement learning (RL) to solve MTSP, respectively. For convenience, MVP that uses GA to plan the paths of UAVs is named GA-MVP and MVP that uses RL to plan the paths of UAVs is named RL-MVP.

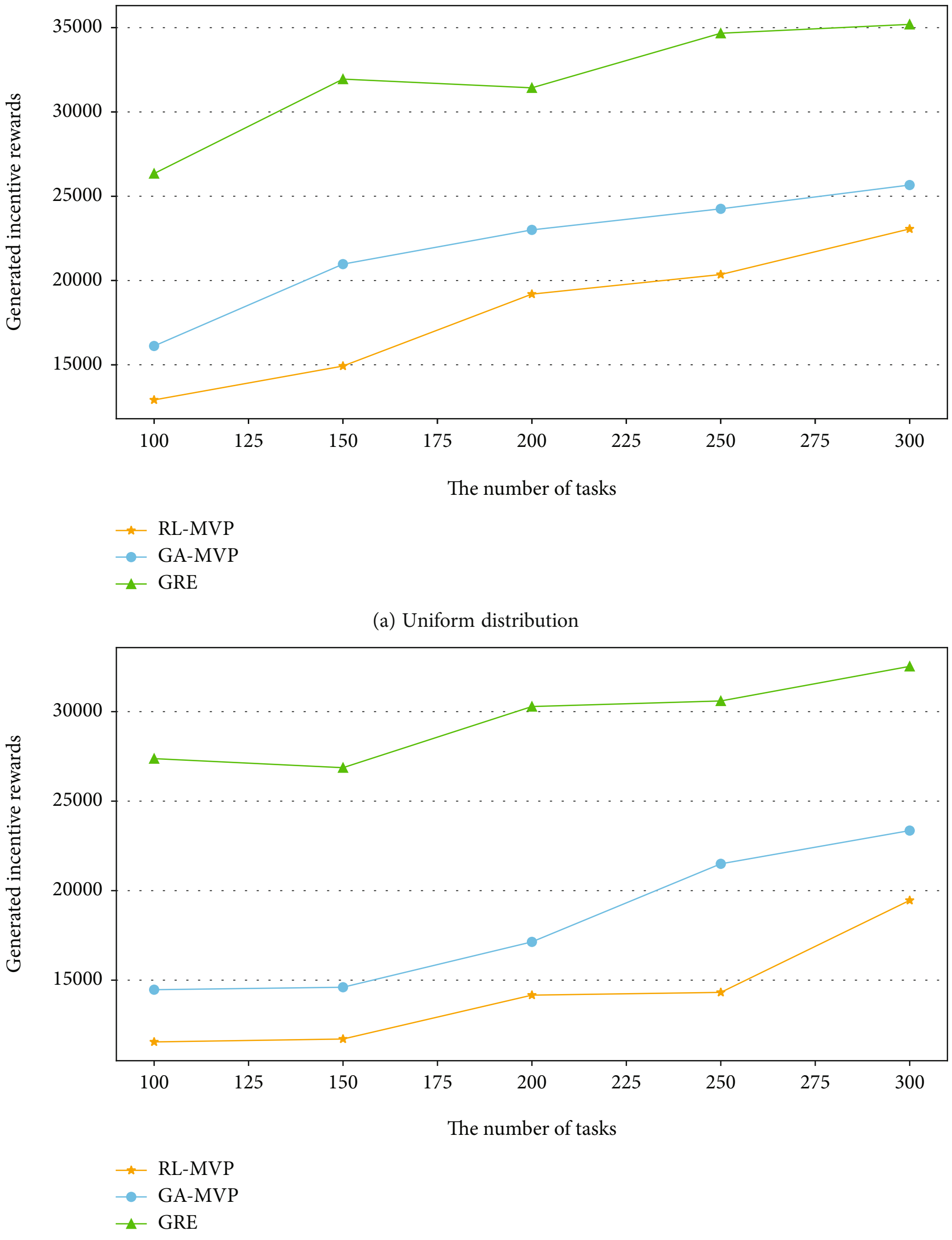
4.5.1. Genetic Algorithm. GA is a search algorithm used in computational mathematics to solve optimization. And it is a type of evolutionary algorithm. It has been widely used in various combinatorial optimization problems. The procedure of planning paths of UAVs based on GA is illustrated in Algorithm 3.

In the algorithm, one individual of the population is a 2D array, which is the respective routes taken by UAVs. P_c and P_m are the probability of crossover and the probability of mutation, respectively. Besides, the fitness function is defined as the reciprocal of the flight distance of UAVs.

The algorithm first randomly generates some individuals as the initial population (line 1) and then starts an iterative process to evolve the initial population (lines 2-16). In each iteration, the fitness of each individual in the population is calculated (line 3). Besides, we take out two individuals that have the largest fitness iteratively and perform crossover on these two individuals with the probability P_c to generate a new individual (lines 6-9). After that, the mutation is performed on the new individual with the probability P_m (lines 10-12). When the iteration ends, we get a new population newPop (line 15). Finally, we select the individual with maximum fitness in the newPop as the final routes of UAVs (lines 17-18).

4.5.2. Reinforcement Learning. Reinforcement learning is an area of machine learning that learns what to do in an environment to maximize a numerical reward. Since a traditional heuristic algorithm for solving combinatorial optimization problems may often be suboptimal due to the hard nature of the problems, RL is a good alternative to search the solution. In our work, we adopt the learning-based approach in [24] to optimize the MTSP, i.e., the path planning of UAVs.

4.6. Algorithm Complexity Analysis. In this section, we analyze the time complexity of GA-MVP. GA needs to evolve the population |iteration| times. In each evolving process, it will perform crossover or mutation action to generate a new population that contains |populationSize| individuals. Hence, the running time complexity of GA will be $O(|\text{iteration}| \times |\text{populationSize}|)$. To generate the route of a ground vehicle, GA-MVP needs to generate |GA| candidate routes. For each candidate route, GA-MVP needs to search at most $|V_p|$ parking spots. Then, GA will be used to solve MTSP in each parking spot. Since we will generate at most |GA| routes, the time complexity of GA-MVP is $O(|\text{GA}|^2 \times |V_p| \times |\text{iteration}| \times |\text{populationSize}|)$.



(a) Uniform distribution

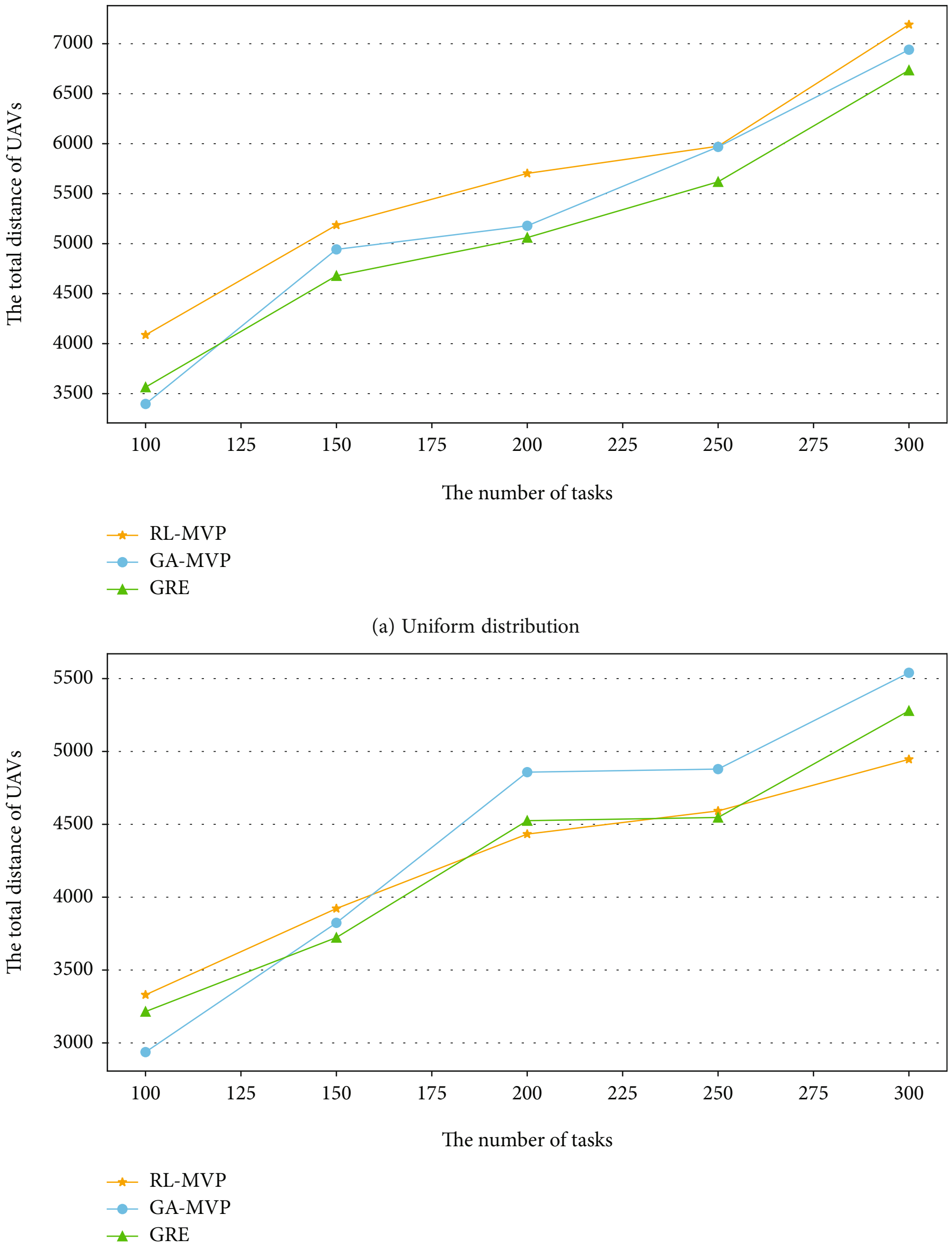
(b) Nonuniform distribution

Figure 7: The impact of the number of tasks on the generated incentive rewards in the environment of multiple vehicles.

5. Experimental Simulation

In this section, we evaluate the proposed algorithm through some simulation experiments. We use the simulator implemented in [25]. In our experiments, a number of detecting points are distributed in the 240 units × 240 units region, in which 1 unit represents 50 m in reality. The roads in the region are generated randomly. For detecting points, we consider two types of distribution: uniform distribution and nonuniform distribution. Besides, a set of ground vehicles is distributed randomly in the target region and is waiting to be employed. The hovering time and speed of all drones possessed by different ground vehicles are identical. However, different ground vehicles may have varying numbers of drones. The speed of drones and ground vehicles are set as 5 m/s and 10 m/s, respectively.

Since we are the first to study the path planning of multi-vehicle-assisted multi-UAVs, there is no existing algorithm that supports employing multiple vehicles. Hence, to evaluate the performance of MVP in the environment of multiple vehicles, we design a naive greedy algorithm (GRE) as the baseline algorithm. In GRE, we utilize the method in VAMU [12] to select parking spots and allocate detecting points. The allocation of parking spots in GRE is greedy as it simply



(a) Uniform distribution

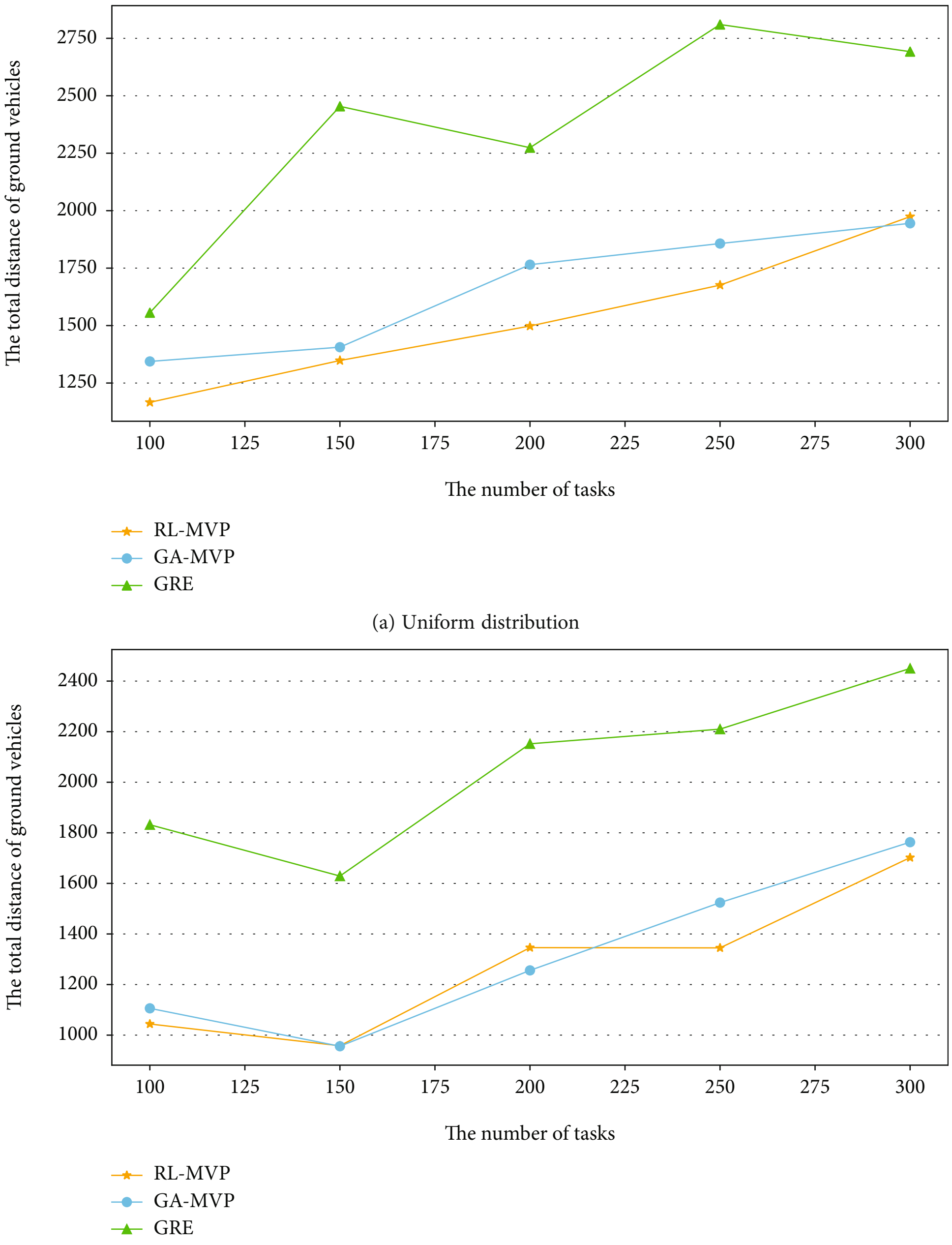
(b) Nonuniform distribution

Figure 8: The impact of the number of tasks on the distance of UAVs in the environment of multiple vehicles.

employs all ground vehicles and allocates the closest parking spots to each of them. After that, for each ground vehicle, GRE utilizes GA to plan the path of the ground vehicle and UAVs. Besides, to make the experiments more complete, we compare MVP with VURA [18]. VURA is a vehicle-assisted multi-UAV routing and scheduling algorithm, but it only employs one vehicle. So the number of available vehicles of MVP is limited to 1 when compared with VURA. In a word, we compare MVP with GRE in the environment of multiple vehicles but compare MVP with VURA in the environment of a single vehicle.

In our experiments, we focus on two metrics. The first is the incentive cost of the algorithm. It is defined by the total incentive rewards produced in the whole procedure of performing all sensing tasks. The second metric is the number of employed vehicles. The smaller the number of required vehicles, the better the performance of the algorithm in the situation where vehicles may not be enough.

We designed some experiments to compare the performance of MVP and GRE. First, we varied the number of tasks (i.e., the number of detecting points) to study the impact of the number of tasks. Besides, we also investigated



(a) Uniform distribution

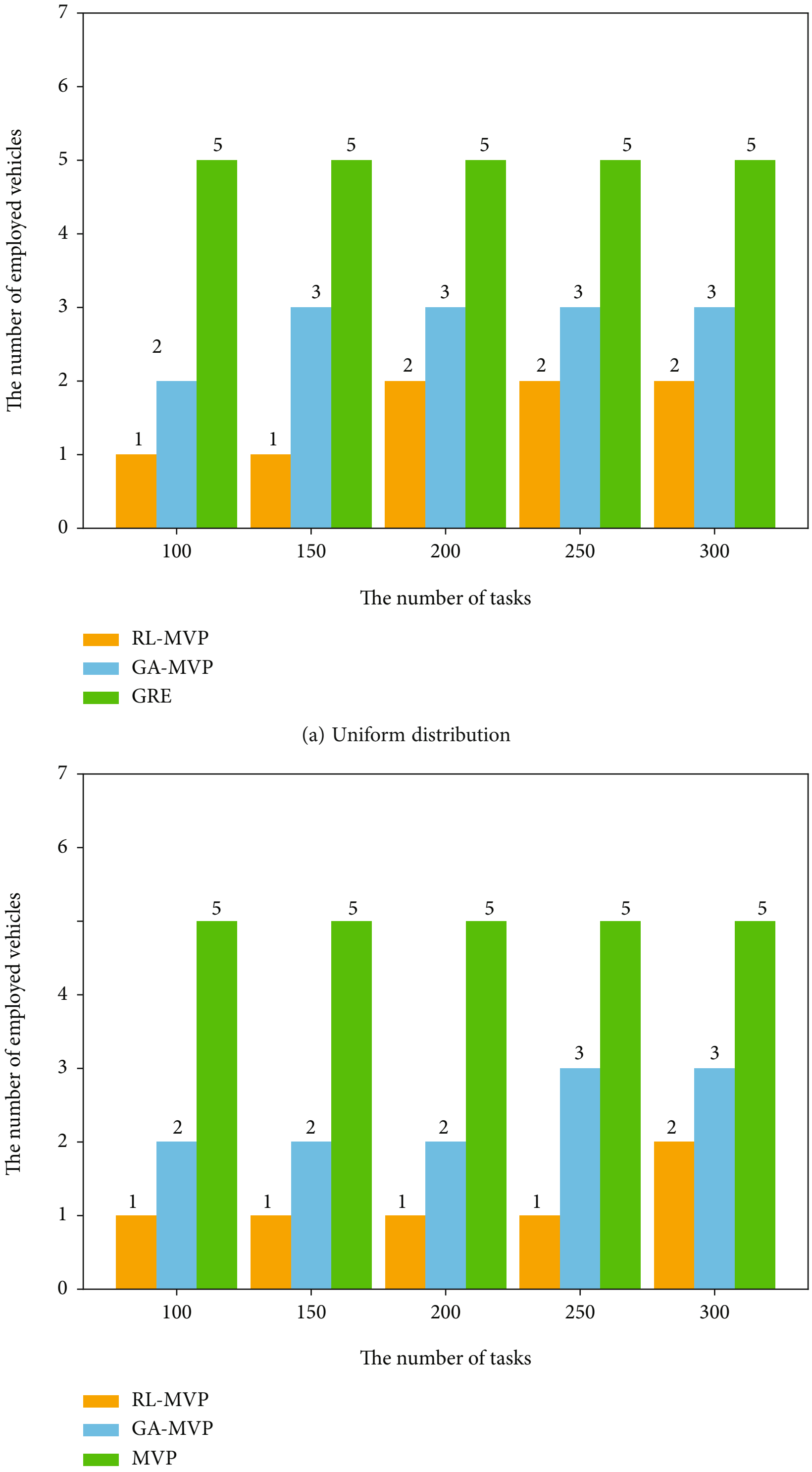
(b) Nonuniform distribution

Figure 9: The impact of the number of tasks on the distance of vehicles in the environment of multiple vehicles.

the impact of the number of tasks on the travel distance of UAVs and vehicles. Then, the time budget is varied to study its impact. After that, we varied the speed of UAVs and the speed of ground vehicles, respectively. Finally, several experiments are performed to investigate the impact of N_m for MVP. For each of the above experiments, we performed it in two environments: (1) uniform environment: detecting points are distributed uniformly, and (2) nonuniform environment: detecting points are distributed nonuniformly.

5.1. The Impact of the Number of Tasks in the Environment of Multiple Vehicles. Figure 7 shows the impact of the number of sensing tasks on the generated incentive rewards. It can be observed that the incentive rewards generated in all algorithms are positively related to the number of sensing tasks. Moreover, GA-MVP and RL-MVP outperform GRE significantly. Besides, RL-MVP outperforms GA-MVP, which represents that RL performs better than GA in the path planning of UAVs.

Figures 8 and 9 present the impact of the number of tasks on the total distance of drones and vehicles, respectively. Figure 8 shows that the total flight distance of all drones in all algorithms is approximately the same regardless of the varying number of tasks. Details shown in



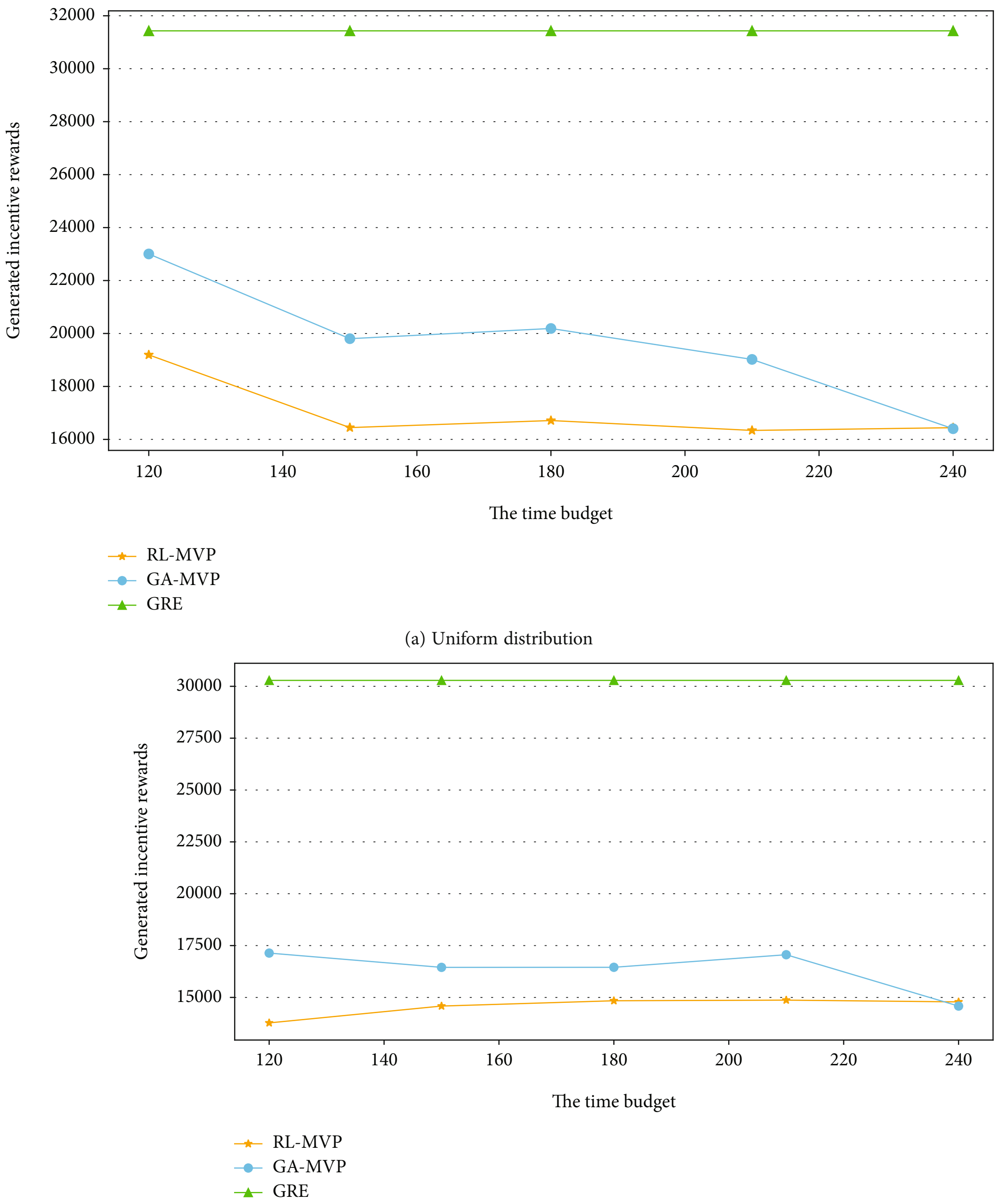
(a) Uniform distribution

(b) Nonuniform distribution

Figure 10: The impact of the number of tasks on the number of employed vehicles in the environment of multiple vehicles.

Figure 9 indicate that the total travel distance of all vehicles in GA-MVP and RL-MVP is smaller than that of GRE, which is contributed by the strategy of allocating parking spots in MVP. Nevertheless, the total travel distance of all vehicles increases as the number of tasks increases.

Figure 10 shows that GA-MVP and RL-MVP perform better than GRE in terms of the number of employed vehicles. The number of employed vehicles in GRE is a constant as GRE simply employs all candidate vehicles to perform tasks. Besides, from an overall perspective, the number of



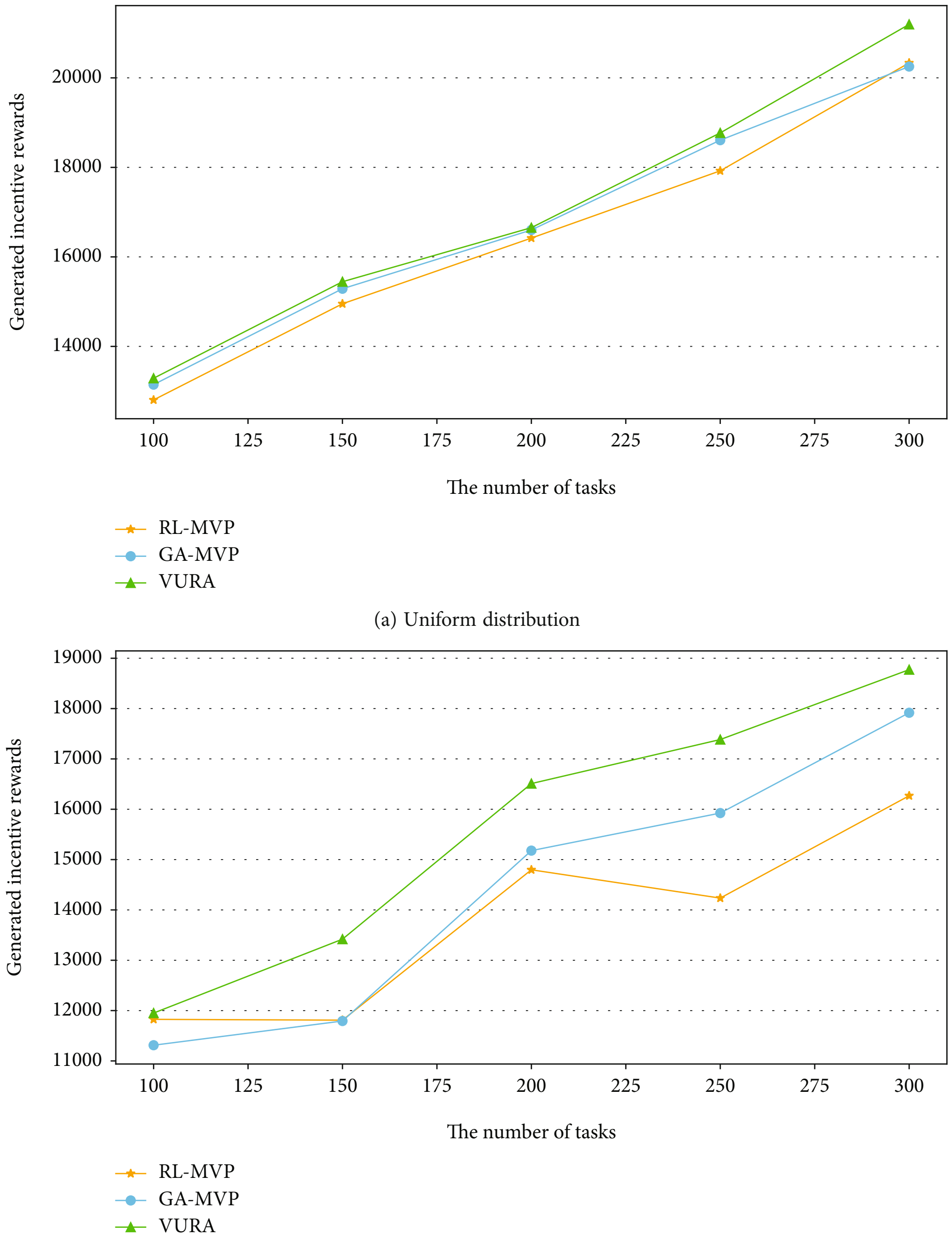
(a) Uniform distribution

(b) Nonuniform distribution

Figure 11: The impact of the time budget on the generated incentive rewards in the environment of multiple vehicles.

employed vehicles in GA-MVP and RL-MVP increases as the number of tasks increases. This is because when the number of tasks increases, more ground vehicles need to be employed due to the fixed time budget.

5.2. The Impact of the Time Budget in the Environment of Multiple Vehicles. Figure 11 indicates that GA-MVP and RL-MVP outperform GRE regardless of the varying time budget. It can be seen that the performance of GRE is not affected regardless of the varying time budget. This is because the time cost of GRE will not exceed the time budget as there are enough ground vehicles. Thus, the time budget does not affect the performance of GRE. Besides, the generated incentive rewards in GA-MVP and RL-MVP decrease as time budget increases in the uniform environment but almost remains unchanged in the nonuniform environment. This is because the number of employed vehicles almost cannot decrease anymore in nonuniform environment.



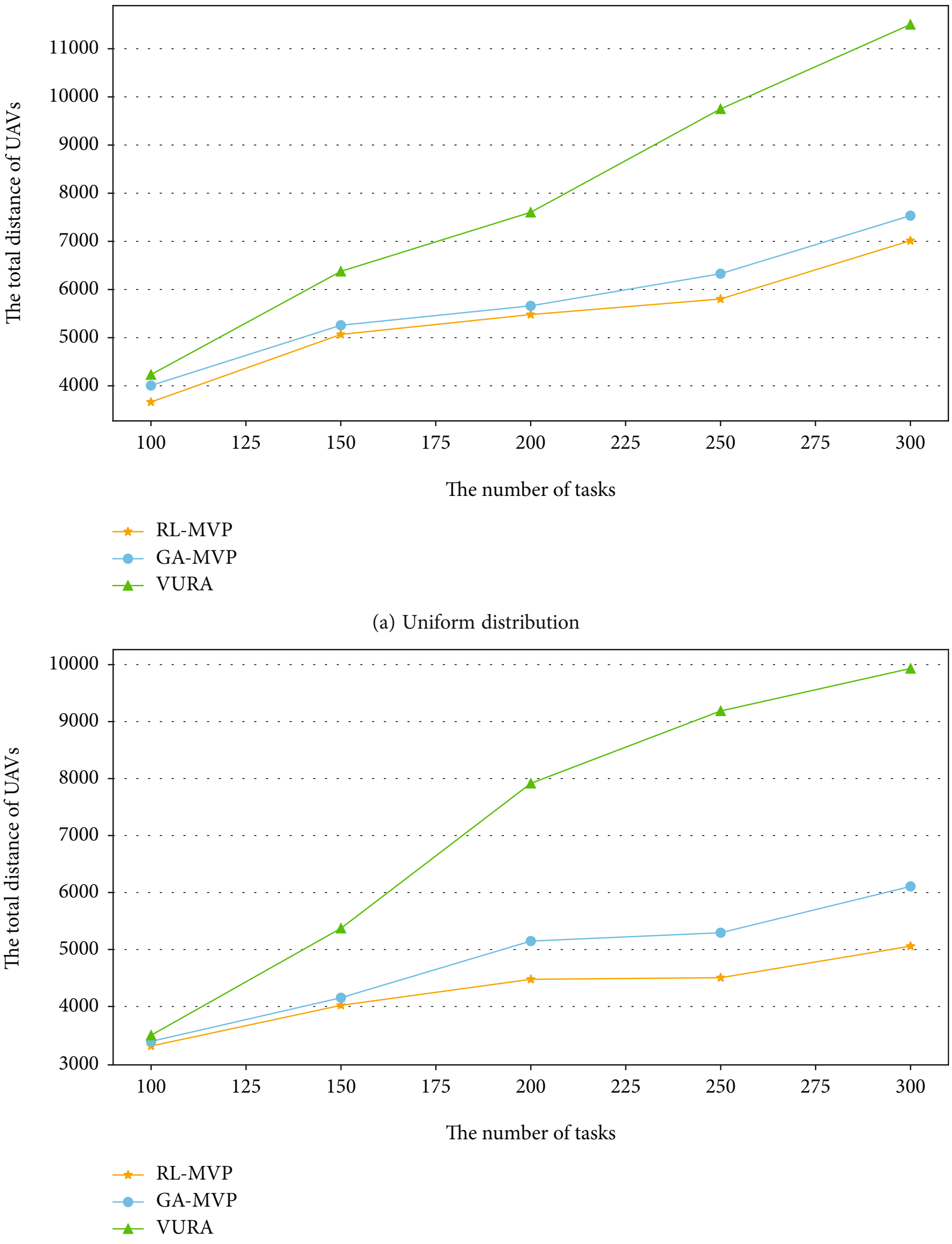
(a) Uniform distribution

(b) Nonuniform distribution

FIGURE 12: The impact of the number of tasks on the generated incentive rewards in the environment of a single vehicle.

5.3. The Impact of the Number of Tasks in the Environment of a Single Vehicle. Figures 12–14 compare RL-MVP and GA-MVP with VURA in the environment of a single vehicle. Figure 12 indicates that the incentive rewards generated in all algorithms are positively related to the number of sensing tasks. It can be observed that RL-MVP outperforms other algorithms in both uniform and nonuniform environments. Besides, VURA performs as well as GA-MVP in the uniform environment but does not in the nonuniform environment. Figures 13 and 14 show that GA-MVP and RL-MVP outperform VURA in the path planning of UAVs but are inferior to VURA in the path planning of the vehicle. The results are acceptable for us since MVP is designed to solve the problem in the environment of multiple vehicles rather than in the environment of a single vehicle.

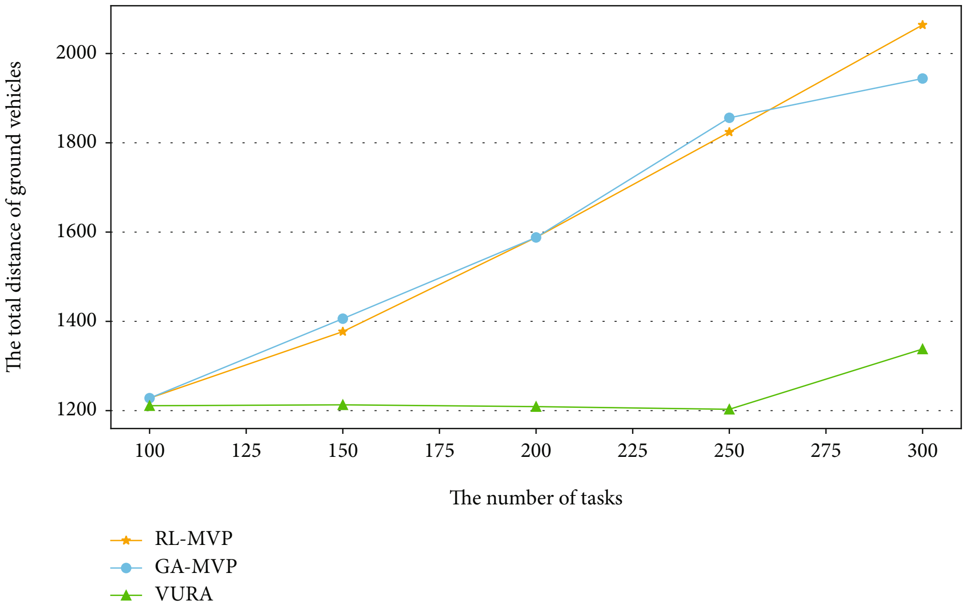
5.4. The Impact of N_m. Figure 15 indicates that the generated incentive rewards in MVP decrease as the value of N_m increases. Besides, we can observe that the generated incentive rewards reduce more gently in the nonuniform environment. This is because when the detecting points are distributed nonuniformly, most parking spots have enough detecting points, which will not cause some drones to sit idle. Finally, as shown in Figure 15, the value of N_m can be set to 6 so that MVP can work well in both uniform environment and nonuniform environment.



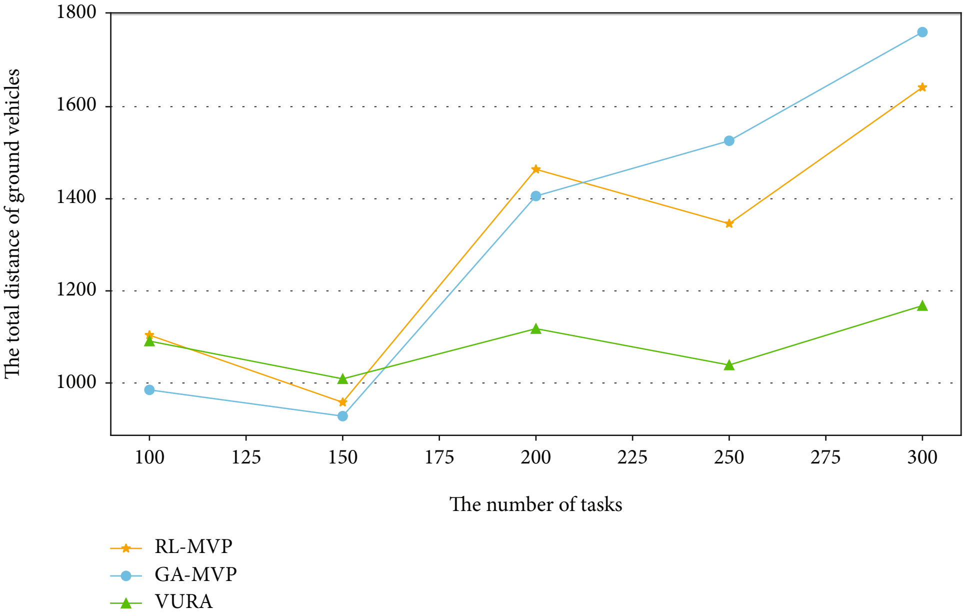
(a) Uniform distribution

(b) Nonuniform distribution

Figure 13: The impact of the number of tasks on the distance of UAVs in the environment of a single vehicle.



(a) Uniform distribution



(b) Nonuniform distribution

Figure 14: The impact of the number of tasks on the distance of vehicles in the environment of a single vehicle.

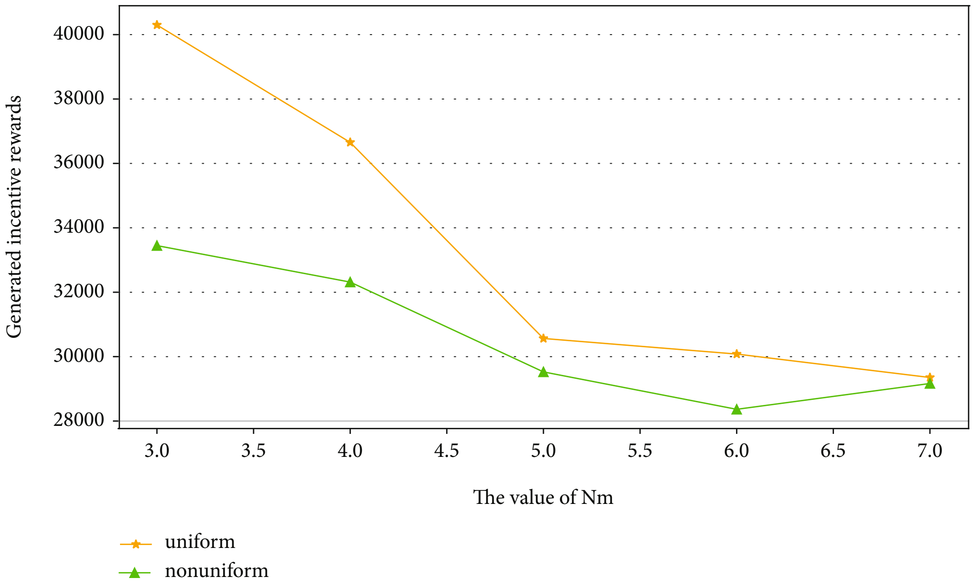


Figure 15: The impact of N_m for MVP.

5.5. The Impact of the Method of Selecting Candidate Parking Spot. Figure 4 presents the total time cost of arranging candidate parking spots, allocating detecting points, and optimizing the allocation in both methods. It can be noticed that the method that arranges candidate parking spots nonuniformly costs more time than the other method. This is because the calculation of density is time-consuming. Besides, Figures 5 and 6 indicate that the generated incentive rewards in both methods are approximately the same.

In a word, RL-MVP always performs better significantly, i.e., generating fewer incentive rewards and requiring fewer ground vehicles when the number of tasks, the time budget, the speed of UAVs, or the speed of ground vehicles varies.

6. Conclusion

In this paper, we propose an efficient algorithm called MVP to address the multi-vehicle-assisted multi-UAVs path planning problem in MCS. To the best of our knowledge, we are the first to take multiple ground vehicles into consideration. In MVP, at first, the detecting points are allocated to proper parking spots. Subsequently, we propose an efficient heuristic allocation algorithm EHA to plan the paths of ground vehicles. Besides, the genetic algorithm and reinforcement learning are utilized to plan the paths of UAVs. Simulation results show that RL-MVP outperforms the other algorithms in terms of the generated incentive rewards in most cases. In the environment of multiple vehicles, GRE produces about 50%-80% more incentive rewards than RL-MVP and GA-MVP produces about 20% more incentive rewards than RL-MVP. In the environment where only one vehicle is available and detecting points are distributed nonuniformly, VURA produces about 15% more incentive rewards than RL-MVP and GA-MVP produces about 7% more incentive rewards than RL-MVP.

In this work, we assume that all UAVs have the same flying speed and hovering time. However, in the reality, one ground vehicle may carry different kinds of UAVs. And different types of UAVs have different flight speeds and hovering time. Hence, as for further work, we will consider the situation where the UAVs carried by one vehicle may be heterogeneous.

Data Availability

The data used to support the findings of this study are included within the article

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. U20B2050 and the Science and Technology Funds from National State Grid Ltd. (the Research on Key Technologies of Distributed Parallel Database Storage and Processing based on Big Data).

References

[1] H. Ma, D. Zhao, and P. Yuan, “Opportunities in mobile crowd sensing,” *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[2] A. M. A. al-muqarm and F. Rabee, “IoT technologies for mobile crowd sensing in smart cities,” *Journal of Communications*, vol. 14, no. 8, pp. 745–757, 2019.

[3] Z. Zhou, J. Feng, B. Ai, and M. Guizani, “Energy-efficient mobile crowd sensing based on unmanned aerial vehicles,” in *2018 IEEE global communications conference (GLOBECOM)*, pp. 1–6, Abu Dhabi, United Arab Emirates, 2018.

[4] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, “Mobility-aware caching and computation offloading in 5G

ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. 974, 2016.

[5] Y. Yang, Z. Zheng, K. Bian, L. Song, and Z. Han, "Real-time profiling of fine-grained air quality index distribution using UAV sensing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 186–198, 2017.

[6] 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.

[7] M. Chen, Y. Tian, G. Fortino, J. Zhang, and I. Humar, "Cognitive internet of vehicles," *Computer Communications*, vol. 120, pp. 58–70, 2018.

[8] 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.

[9] A. Ponza, *Optimization of Drone-Assisted Parcel Delivery*, 2016.

[10] S. Mourelo 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 (JIEM)*, vol. 9, no. 2, pp. 374–388, 2016.

[11] 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, p. 1144, 2017.

[12] M. Hu, W. Liu, J. Lu et al., "On the joint design of routing and scheduling for vehicle-assisted multi-UAV inspection," *Future Generation Computer Systems*, vol. 94, pp. 214–223, 2019.

[13] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 745–753, Toronto, ON, Canada, April 2014.

[14] H. Xiong, D. Zhang, Z. Guo, G. Chen, and L. E. Barnes, "Near-optimal incentive allocation for piggyback crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 120–125, 2017.

[15] Z. Shi, H. Huang, Y.-E. Sun, X. Wu, F. Li, and M. Tian, "An efficient task assignment mechanism for crowdsensing systems," *International Conference on Cloud Computing and Security*, 2016, pp. 14–24, Springer, 2016.

[16] Y. Chen, P. Lv, D. Guo, T. Zhou, and M. Xu, "Trajectory segment selection with limited budget in mobile crowd sensing," *Pervasive and Mobile Computing*, vol. 40, pp. 123–138, 2017.

[17] 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.

[18] M. Hu, W. Liu, K. Peng et al., "Joint routing and scheduling for vehicle-assisted multidrone surveillance," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1781–1790, 2018.

[19] A. Guttman, "R-trees: a dynamic index for geometric data," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 47–57, Brighton, England, 1984.

[20] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pp. 71–79, San Jose California USA, 1995.

[21] P. Li and J. Xu, "Fundamental rate limits of UAV-enabled multiple access channel with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 458–474, 2019.

[22] Z. Fang, J. Wang, C. Jiang, Q. Zhang, and Y. Ren, "AoI-inspired collaborative information collection for AUV-assisted Internet of underwater things," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14559–14571, 2021.

[23] Z. Fang, J. Wang, J. Du, X. Hou, Y. Ren, and Z. Han, "Stochastic optimization aided energy-efficient information collection in Internet of underwater things networks," *IEEE Internet of Things Journal*, 2021.

[24] Y. Hu, Y. Yao, and W. S. Lee, "A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs," *Knowledge-Based Systems*, vol. 204, article 106244, 2020.

[25] J. Xi, L. Liu, M. Li, X. Li, and J. Peng, "Power-aware path planning for vehicle-assisted multi-UAVs in mobile crowd sensing," in *2021 IFIP/IEEE international symposium on integrated network management (IM)*, pp. 687–691, Bordeaux, France, 2021.