Article

Cooperated Routing Problem for Ground Vehicle and Unmanned Aerial Vehicle: The Application on Intelligence, Surveillance and Reconnaissance Missions

Yao Liu [1], Zhihao Luo [1,*], Zhong Liu [1], Jianmai Shi [1], and Guangquan Cheng [1]

[1] Science and Technology on Information Systems Engineering Laboratory, College of System Engineering, National University of Defense Technology, Changsha, 410073, P. R. China; liuyao13@nudt.edu.cn (Y.L.); liuzhong@nudt.edu.cn (Z.L.); jianmaishi@nudt.edu.cn (J.S.); cgq299@nudt.edu.cn (G.C.)

* Correspondence: luozhihao15@nudt.edu.cn; Tel.: +86-151-1106-6043

Abstract: In this paper, we present a novel Two-Echelon Ground Vehicle and Its Mounted Unmanned Aerial Vehicle Cooperated Routing Problem (2E-GUCRP). The 2E-GUCRP arises in the field of Unmanned Aerial Vehicle (UAV) Routing Problem such as those encountered in the context of city logistics. In a typical cooperated system, the UAV is launched from the Ground Vehicle (GV) and automatically flies to the designated target. Meanwhile, acting as a mobile base station, the GV can charge or change the UAV's battery on the designated landing points to enable the UAV to continue its mission. The objective is to design efficient GV and UAV routes to minimize the total mission time while meeting the operational constraints. A Mixed Integer Programming (MIP) model, which could be solved by commercial software, is constructed to describe this problem. In order to quickly solve the medium-scale problems, two existing heuristics to solve 2E-VRP are improved. The computational experiments are set up to compare our model with the 2E-VRP. The results indicate that the 2E-GUCRP obtains a better efficiency. Further discussion of the practical instance points out that the increase in efficiency is related to the speed relationship between the GV and the UAV.

Keywords: two-echelon routing; vehicle routing; vehicle-mounted UAVs; ISR mission

1. Introduction

With the miniaturization of high-accuracy sensors, UAVs have played increasing roles in both military and civilian applications, such as reconnaissance and surveillance expeditions, border patrol, weather and hurricane monitoring, crop monitoring. Performing as a mobile platform, the UAV has advantages of portability, effectivity and low risk, which is more versatile for intelligence, surveillance, and reconnaissance (ISR) missions. In a typical ISR mission, the UAV is required to transmit the collected data or images to the base station. Large fixed-wing UAVs can use satellite for real-time communication, while small rotorcrafts must transmit data through signal towers. However, in the environment with poor communication conditions, both large UAVs and small rotorcrafts would encounter communication difficulties and can only keep the data in their storage devices. Besides, with limited capacity of battery power, the UAV cannot execute tasks for a long time. In this case, it would greatly improve the efficiency of ISR mission to have a GV as the mobile base station, which can both charge the UAVs and collect data from their storages.

Actually, this mode of cooperated GV and UAV has been gradually popularized in recent years. At the beginning of 2017, UPS tested this working mode and successfully employed a GV and its mounted UAV for parcel delivery [1]. The similar tests have also been carried out in China. In 2016, JD.com applied UAVs on delivering online purchases to rural shoppers in Jiangsu province, kicking

 doi:10.20944/preprints201902.0110.v1

off the trail operation of "last mile" distribution. As for the applications of reconnaissance, DJI-Innovations and Ford proposed a novel mode for wild search in 2016 International Consumer Electronics Show (CES). In this mode, the F-150 and its carried UAV cooperatively conducted the search for multiple targets in the wilderness [2].

There also has been some pioneering work for investigating the parcel delivery problem for cooperated GV and UAV. To our knowledge, the general idea of existing methods is to describe the process with a classic two-echelon vehicle routing problem(2E-VRP), where the first level route includes the selection of depot locations and delivery route to depots while the second level routes are from depots to customers. When applying 2E-VRP model to solve the routing problem of GV and its mounted UAV, the temporary stopping point of GV is considered as the depot, and the GV route is in the first level route while the UAV travels the second level routes.

On this basis, Manyam et.al. [3] proposed the cooperative air-ground vehicle routing problem (CAGVRP) and apply this cooperation model to gather information. In their problem, the UAV is unable to control completely autonomously, and a reliable communication link between the UAV and the GV must be guaranteed. Thus, they keep the GV staying still when the UAV performs the task, and consider the UAV route as a subtour of ground vehicle route. A feasible solution to their problem is shown in Figure 1. Essentially, this problem is also a variation of the typical two-echelon vehicle routing problem(2E-VRP). However, with the development of automatic obstacle avoidance and navigation technology, the current UAV is capable to finish the simple tasks automatically without real-time control. Regardless of the constraints about communication distance, the GV can synchronously complete other tasks or travel to another rendezvous node when UAV performs tasks. Thus, the main difference between our problem and 2E-VRP is that two-echelon routes in our problem are operating in the exact same time and have a strong synchronization.

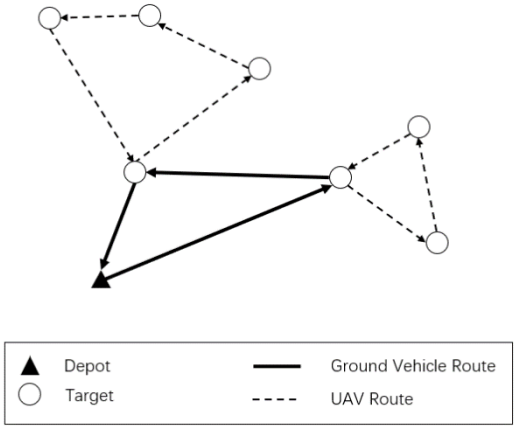

Figure 1. Typical CAGVRP Solution

Another typical extensive study on this model is the application in the field of logistics transportation. Murray and Chu [4] analyzed a variant of Travelling Salesman Problem (TSP) with single GV and single UAV called the flying sidekick TSP (FSTSP), where both the GV and the UAV start and end at one depot, the GV travels on the road network while the UAV can deliver by itself. Agatz et al. [5] and Ha et al. [6, 7] also present the similar formulation, the TSP with a drone (TSP-D). In these researches, the ground vehicles are settled to move to the next rendezvous node for recycling and recharging the UAV. Besides, these studies focus on the logistics, in which the UAV is restricted to serve only one customer in one flight. A feasible solution to their problem is shown in Figure 2. Unlike FSTSP/TSP-D, this paper concentrates on the ISR mission, where UAV is capable to serve multiple targets in one flight.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

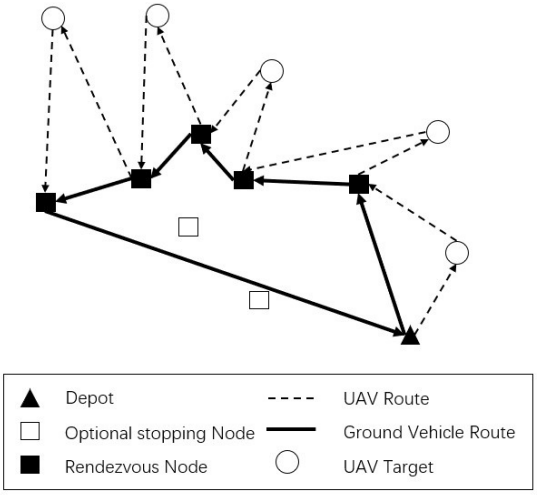

Figure 2. Typical FSTSP/ TSP-D Solution

In this paper, we focus on the applications in ISR missions and promote a novel problem, the Two-Echelon Ground Vehicle and Its Mounted UAV Cooperated Routing Problem (2E-GUCRP). In our problem, a very simple system of heterogeneous vehicles is involved, where a UAV with fast speed and limited endurance is deployed on a slower GV with much longer operating range. Besides, there are a set of predetermined targets located outside the road, which can only be visited by the UAV. As for the GV, it can only stop on the optional stopping nodes (parking lots) in the road network, which could allow the UAV to take off or/and land. Thus, in the cooperation process illustrated in Figure 3, the GV departures from a depot with its mounted UAV, takes a trip on road network and select some stopping nodes to launch and recycle the UAV, and then travels back to the previous depot or another depot. Synchronously, the UAV takes off from the GV, automatically flies to the target points, collect the target information and then returns to the GV before the battery powers off.

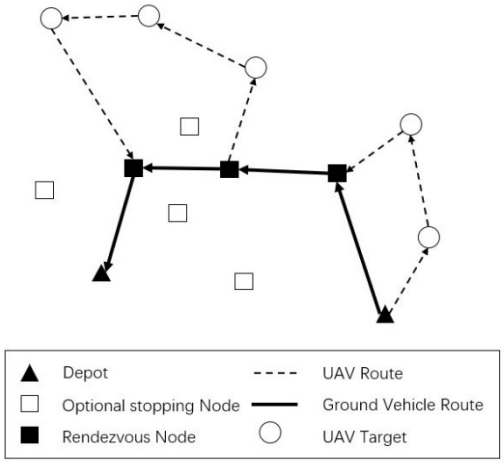

Figure 3. Typical 2E-GUCRP Solution

The main advantages of this reconnaissance pattern are three folds: (i) the shortage of UAV's endurance can be made up as the GV can carry UAV to patrol more extensive reconnaissance areas; (ii) with the GV restricted on partial road networks, the involvement of the UAV enables the cooperative system to rapidly explore some areas unreachable by the GV; (iii) with no drivers (or pilots) on board, UAV is able to perform more dirty, dangerous and dull missions. Nevertheless, with two-echelon routes involved and their strong synchronization, it also brings many challenges on the route planning. To minimize the time of the UAV route, there are three critical decisions to be made:

 doi:10.20944/preprints201902.0110.v1

planning the GV's route, determining where to stop the GV and launch and/or recycle the UAV, and planning the UAV's route for visiting targets during each flight.

Since this is a new phenomenon, there is an urgent need for new models and innovative algorithms to help to better understand the related planning problems and potential benefits. This paper aims to fill this gap by developing a new mixed integer programming formulation as well as two fast heuristics based on the deconstruction of this problem. At the same time, the differences between the new model and the traditional 2E-VRP are compared thorough solving the same problem, so as to explore the improvement of the new working model in efficiency.

The remainder of this paper is organized as follows. In the 2nd section, previous studies on relative problems have been classified. Three meta-problems are constructed according to the shape of the route and the differences among previous literatures and this paper are further compared. In the 3rd section, we analyze the characteristics of the cooperated work model and the challenges in the routing progress. In view of the characteristics of this two-echelon cooperative routing problem, a mixed integer programming (MIP) model is established after considering complex aspects such as different modalities of cooperation, synchronization and endurance constraints. In the 4th section, a Clustered Assignment (CA) heuristic is proposed to construct feasible solutions of 2E-GUCRP in seconds. And a variation of custom Split heuristic is proposed for comparison. The effectiveness of the proposed heuristics is verified in the 5th section with a series of randomly generated instances and a practical instance of Changsha city. in the last section, we summarized the conclusion and the innovation of this paper.

2. Literature Review

According to the choice of starting point and destination in the routing process, the classic routing problem can be divided into the following three meta-problem:

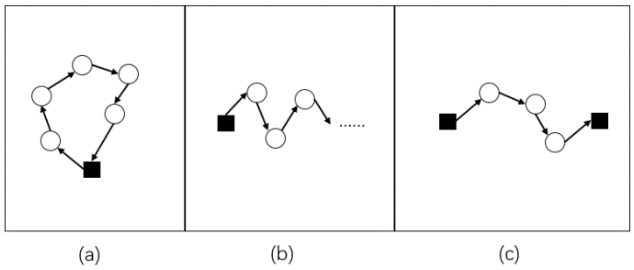

Figure 4. Three Classic Routing Problems

(i) The circulation routing problem: as shown in Figure 4(a), this type of problem requires the vehicle to start from a fixed base station and return to the base station after all the target points have been visited/served.

(ii) The radiation routing problem: as shown in Figure 4(b), this type of problem also requires the vehicle to start from a fixed base station but does not need to return the base station after visiting all the target points. Since this route can end at any target point, there is no limit to the destination of the route.

(iii) The segment routing problem: as shown in Figure 4(c), this type of problem requires the vehicle to start from a fixed base station and return to another designated base station after all the target points have been visited/served.

The circulation routing problems are the most widely studied problems. The Travelling Salesman Problem (TSP) and Vehicle Routing Problem (VRP) are both typical circulation problems. But differing from TSP, VRP considers the constraints of the endurance (or capability) of the routing process, which limits the number of points (or customers) that can be accessed per trip.

As a variation of the basic circulation problems, the radiation routing problems are also widely studied as Open Problem in transaction field. There are some common open questions including Open Travelling Salesman Problem (OTSP) and Open Vehicle Routing Problem (OVRP), which are quite practical in some special situations.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

The segment routing problem is a combination of the circulation routing problem and the radiation routing problem. There are handful researches about the segment routing problem in academic. In actual fact, Specifically, if the start point and end point is determined, we could add a virtual point as Figure 5, whose out-degree is equal to that of end point and in-degree is equal to that of start point, then the segment routing problem can be transformed into a TSP. Thus, it doesn't make much sense to study the problem separately. However, the problem would become much more difficult when the start point and the end point need to be selected among the candidate points. Therefore, we consider the segment routing problem as another separate class of problems.

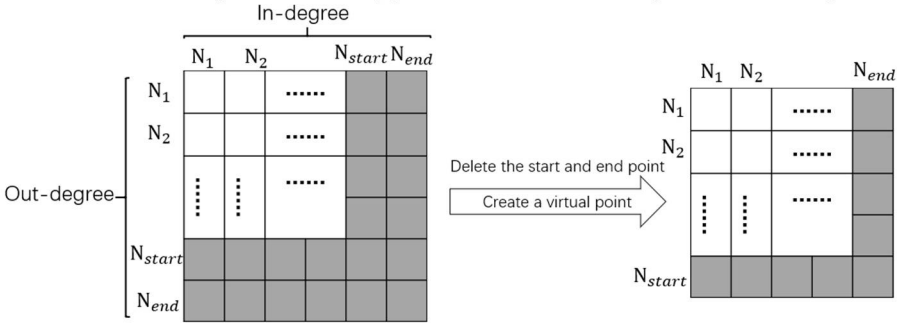

Figure 5. The 'virtual point'

From this point of view, the existing literatures are classified into three classic routing problems.

Generally, since there is more than one vehicle employed in each depot in 2E-VRP, this problem can be reclassified as a single/multiple-circulation-single/multiple-circulation problem. For instance, Zeng et.al [8] presents a Two-Echelon Open Vehicle Routing Problem(2E-OVRP) model to solve the problem where vehicles of both echelons are not required to return to starting points after finishing their missions. Their problem can be classified as a multiple-radiation-multiple-radiation problem. Besides, Manyam et.al [3] applied a typical 2E-VRP model to solve the problem of the ISR mission performed by a GV and its mounted UAV. As mentioned above, the first level of GV route is a classic TSP, which is also a typical circulation problem. Since the GV would stay still when the UAV flies, the second layer of the UAV route is still a classic TSP. Thus, their problems can be considered as a single-circulation-single-circulation problem.

There are also some researches about Traveling Salesman Problem with Drone (TSP-D), such as Agatz et.al [5] and Ha et.al [6, 7]. In their problems, the cooperated system of GV and UAV is applied on the city logistics. The GV carries a UAV to complete the delivery task of multiple target points. All target points can be served by GV or UAV. However, every time the UAV starts from the vehicle, it can only serve one target point, and then return to the vehicle immediately. There is a limitation for GV that it must launch the UAV at one customer point and recycle it at the next point, in which the UAV route would form a "triangle" with GV routing. Thus, their problems can be classified as a single-circulation-single-segment problem. Similar to TSP-D, the FSTSP that Murray and Chu [4] proposed is also a single-circulation-single-segment problem. But since the GV can serve multiple costumers while UAV delivers the parcel, this problem is more practical and expands the "triangle routes" to the "polygon routes".

Dorling et.al [9] and Wang et.al [10] constructed a Vehicle Routing Problem with Drone (VRP-D) problem. Unlike the TSP-D, the first level of GV route in VRP-D considers the constraints of the endurance (or capability), which is a specialized VRP. Thus, their problems can be classified as a multi-circulation-single-segment problem. Poikonen et.al [11] and Wang et.al [12] extend the research of VRP-D with the analysis of the theoretical maximum benefit under ideal circumstances.

In addition, Mathew et.al [13] proposed Multiple Warehouse Delivery Problem (MWDP) for a team of cooperating vehicles performing autonomous deliveries in urban environments. Unlike VRP-D and FSTSP, MWDP allows vehicles to stay in place and repeatedly launch and recycle UAVs. Their problems can be classified as a single-circulation-multi-segment problem. Othman et.al [14] proposed a variant of TSP-D called Alternating Last-Stretch Delivery Problem (ALSDP), in which the route of

 doi:10.20944/preprints201902.0110.v1

ground vehicle is fixed and the main decision is to choose a set of rendezvous points for launching and recycling.

The literatures mentioned above mainly involve combined routes for a set of locations. However, a few articles consider other operations, such as area coverage and applying UAVs to expand connectivity of GVs. Halil Savuran and Murat Karakaya [15, 16] proposed a largest coverage problem of a vehicle-mounted UAV (Mobile Depot VRP, MoDVRP). In their case, the vehicle keeps moving along a straight line, and a large number of UAV target points locate on both sides of the vehicle's path. The UAV starts from the vehicle, visits as many target points as possible within the endurance, and then returns to the vehicle at next rendezvous node. This problem is a combination of a specialized radiation routing problem and a partial segment problem. Thus, their problems can be classified as a partial-radiation-partial-segment problem.

Our previous research [17] constructed a partial-circulation-single-segment problem for ISR mission. In that paper, we assumed that there are usually sufficient rendezvous nodes in the road network, and that the GV can always find a rendezvous node for recycling the UAV no matter where it launches the UAV, which means that each road arc traversed by the GV corresponds to a flight route of the UAV. As the research goes further, it is found that this hypothesis limits the problem to have wide applicability. Therefore, in further study, we relaxed this hypothesis and proposed two different special cases based on actual application scenarios. The analysis of these two special cases will unfold in the next section.

Through the literature review above, the differences of models in previous studies can be learned. In this paper, we focus on the 2E-GUCRP. Since the GV only selects some rendezvous nodes among optional stopping nodes on the road network, and the starting node and end node of the GV route may not be the same one, the first level of the 2E-GUCRP is a partial-segment problem. Combined with the UAV's flight route for a typical segment routing, this problem is classified as a partial-segment-single-segment problem.

Table 1. Literature Review

Literature	1st-Level	2nd-level	Notes
Typical 2E-VRP	multi-circulation	multi-circulation	
Zeng et.al[8]	multiple-radiation	multiple-radiation	
Manyam et.al[3]	1-circulation	1-circulation	the GV stays still when the UAV performs its tasks.
Agatz et.al[5] Ha et.al [6] Ha et.al [7]	1-circulation	1- segment	"triangle routes"
Dorling et.al[9] Wang et.al[10]	multi-circulation	1- segment	
Murray and Chu[4]	1-circulation	1- segment	"polygon routes"
Mathew et.al[13]	1-circulation	multi-segment	Multiple UAV routes
Halil Savuran and Murat Karakaya[15] Halil Savuran and Murat Karakaya[16]	partial-radiation	partial -segment	The GV route is fixed on a straight line
Authors previous research[17]	partial -circulation	1- segment	Has a hypothesis of rendezvous nodes
This paper	partial -segment	1- segment	Section 3

3. Mathematical Model

In this section, we will focus on the application background of the problem and establish a MIP model based on the reasonable hypotheses. Finally, we verified the correctness of the MIP model by using the commercial software to solve a randomly generated instance.

3.1. The Discussion of Hypotheses

Describing the task execution of a GV and its mounted UAV is a complicated process. To facilitate modeling, we propose a series of hypotheses to simplify and abstract this problem. They are organized into the following aspects:

3.1.1. General hypothesis

In order to facilitate the modeling and analysis, we abstract the task area into a target point without considering the specific task execution process. Besides, with the improvement of the capacity and endurance, it is capable for UAV to carry out multiple missions during a single flight. The assumption that a UAV can only access one point at a time is not in line with current development.

H1: The UAV's task execution areas and the GV's optional temporary stop areas are abstracted as points.

H2: The UAV can serve one or more targets during one flight.

3.1.2. The discussion of starting points and end points

In most of the previous literature, there was a base, which is both the start point and end point of the route of GVs. It is reasonable as the vehicle is not a disposable consumable in the vast majority of situations. With the recycle of the vehicle, it can be convenient for the unified dispatching management and reduce its costs greatly.

Nevertheless, it is still necessary to study the routing problems with different starting point and end point. There are main considerations are introduced below:

From the practical point of view, this model has wide application prospect. In the civil field, when there are sufficient base stations that can be managed uniformly, the vehicles are capable to return to another base, which could vastly improve the efficiency of the task. In the military field, the adoption of the model that returns to a fixed base will increase the risk of exposing its own forces, deployment and command centers. It is also relatively difficult to build a base in some specific areas such as the area near the frontier. Besides, in actual combat operations, the areas of supply and maintenance tend to differ from the initial assembly areas.

From the perspective for a quick solution, this model provides an approach to deconstruct large-scale problems. It is well known that vehicle routing problem is a NP-hard problem. With the expansion of the problem scale, the growth of solution space is much bigger than that of the problem scale, which will inevitably lead to a sharp increase in the time for solving the problem. However, in this method, the large-scale task can be divided into several sub-tasks through finding some middle points as the temporary end points or start points according to its temporal and spatial characteristics. Then it is much quicker and more effective to obtain the solution of large-scale tasks by solving the sub-tasks parallelly.

In summary, During the task execution, the GV starts from a specified starting depot and chooses partial rendezvous nodes among the optional stopping nodes to complete the launching and recycling work of the UAV. After all the missions ends, the GV travels to another designated depot.

H3: The starting depot and the designated depot of the GV are different points.

3.1.3. The special cases of segments

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

Generally, when a UAV completes its mission, a closer rendezvous node for the GV to recycle and resupply the UAV can reduce the flight time to the land node or the next target, which can improve the efficiency of the whole missions. However, in some extreme cases, such strategy may be ineffective. In the following sections we will describe two extreme cases and build the appropriate hypotheses.

a) Empty edge

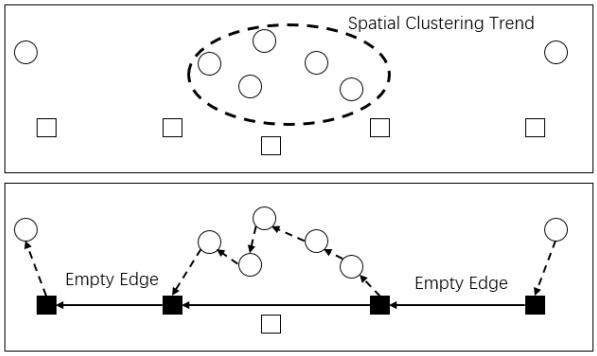

Figure 6. Empty Edge

This scenario arises when there is a spatial clustering trend among the target points. As shown in Figure 6, some target points are close to each other while far away from other target points. In practical application scenarios, spatial clustering trends are very common. For example, when delivering goods in the cities, the distribution of customers is mainly concentrated in various residential areas. Customers in the same residential area are close to each other while far from customers in other residential areas. Therefore, it is necessary to cover this scenario in modeling.

H4: The UAV does not need to take off immediately after landing on the GV to complete the supply and maintenance work and can take off again as the GV travels to the next rendezvous node.

b) Holding Strategy

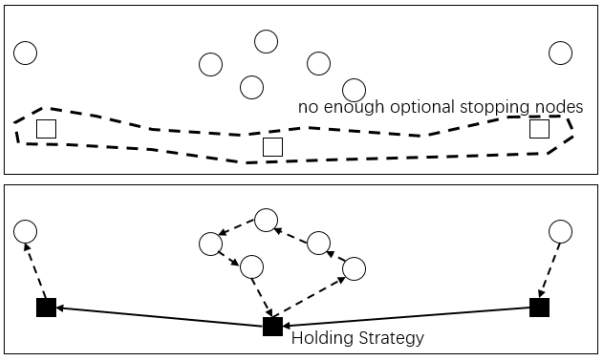

Figure 7. Holding Strategy

This scenario arises when there are not enough optional stopping nodes. Specifically, as shown in Figure 7, the endurance of the UAV cannot support it to fly to a far stopping node after visiting targets. Thus, in order to ensure the existence of a feasible solution, the GV is allowed to stay in place and wait for the UAV. This scenario is also acceptable in practice, such as in the central area of the city where the taking off of the UAV is restricted by air traffic control. Therefore, it is also necessary to cover this scenario in modeling.

H5: The GV can stay in place while the UAV performs its tasks, waiting for the UAV or heading for the next rendezvous node.

 doi:10.20944/preprints201902.0110.v1

3.1.4. Complementary hypothesis

The five hypotheses mentioned above describe the basic routing rules and two extended scenarios of GV and UAV. To better describe our model, we add several related hypotheses here.

Existing technologies can fully support the UAV to perform an autonomous task in the context of a serious of given input. For example, in Amazon's Prime Air plan, they use a combination of GPS and image recognition technology for autonomous delivery [18]. Besides, the United Parcel Service's on-board drone delivery and Google X lab's Project Wing, can also do the self-control process. Thus, without the need for the real-time control of the UAV, we could not consider the limitation of communication distance between GV and UAV.

H6: The limitation of communication distance between GV and UAV is not considered.

At this stage, small and medium-sized UAVs are mainly battery-powered. In order to perform tasks quickly and continuously, a good pattern is to prepare two or more batteries for alternate use. When the UAV performs its task, other batteries can be charged in the GV. In this pattern, the replacement time of the UAV's battery is negligible. In addition, with the development of data transmission technology, the time to import reconnaissance data from UAV into GV is also negligible.

H7: The time of taking off, landing, replacing the battery as well as transmitting data can be ignored. After completing the former flight, the UAV can take off immediately to complete the follow-up tasks.

To simplify the model, two assumptions are also proposed as follows.

H8: The location of the task target point is predefined and its service time is known as a fixed number.

H9: The UAV flies in a straight line at a constant speed during the flight.

3.2. The Discussion of Synchronization

Michael Drexl [19-21] offers a synchronic view of the problem of vehicle routing. He first proposed Vehicle Routing Problems with Multiple Synchronization Constraints (VRPMSs), and then illustrated the influence of the concept of planning Problem through the summary of Vehicle Routing Problem with Trailers and Transshipments (VRPTT) literature. Synchronization is usually required whenever a UAV needs to land on a GV. With different speeds and travel distances of the UAV and the GV, one vehicle may need to wait for the other one. Based on the synchronization analysis of Michael, this paper discusses the synchronic constraints in the ISR missions from the following two aspects.

3.2.1. Generalized synchronization

It is different from the previous multi-echelon vehicle routing problems that the two-echelon routes in 2E-GUCRP need to be carried out in the same time period. As for the traditional 2E-RP, the larger transport vehicles are generally adopted in the first echelon and transports the goods at a lower frequency (once a week, even a month); the second echelon routing is usually completed by smaller transport vehicles with a higher frequency (multiple times a day or a week). Since 2e-RP does not require the co-time of the two stages, changing the second echelon of the route has less impact on the first phase. However, in 2E-GUCRP, the generalized synchronization of the echelon qualifies a macro time period and requires that the tasks of GV and UAV are performed simultaneously.

3.2.2. Spatial-temporal synchronization

Whether the previous literature research or this study, we need to ensure the synchronization between the GV and UAV in time and space. That means when the UAV arrives at a rendezvous node, the GV should arrive at the same rendezvous node before the UAV's landing. To simplify the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

calculation of total time in a segment, we add a restriction that the GV must arrive the rendezvous node before the UAV.

3.3. Mixed Integer Programming Model

Table 2. Notation and its implications

Notations	
G	The graph of nodes and edges.
V	The set of all nodes.
V_s	The set of all optional stopping nodes
V_t	The set of all target nodes.
E	The set of all edge.
E_1	The edge for GV routing
E_2	The edge for UAV routing
C_i	The serve time of node i.
d_{ij}	The distance between node i and node j
v_1	The average velocity of GV
v_2	The average velocity of UAV
θ	The maximum endurance of UAV
M	A sufficiently large positive number

The mathematical formulation is defined on a graph $G = (V, E)$. V is the set of all nodes, where {0} indicates that the vehicle's starting assembly area or base, {*} indicates the destination of the vehicle to assemble a region or a base, and V is composed of the following independent set of sets: $V = \{0\} \cup \{*\} \cup V_s \cup V_t$. The $V_s = \{1,2,\dots,m\}$ represents the set of all optional stopping nodes and the $V_t = \{m+1, m+2, \dots, m+n\}$ represents the set of all target nodes. For each target $j \in V_t$, there is a corresponding service time C_j for the UAV to complete the specified task. Since we assume that UAV is moving at a constant speed, the residual power of UAV can be converted directly to the remaining time. We introduce the following variables:

$$x_{ij} = \begin{cases} 1 & if\ GV\ travels\ from\ node\ i\ to\ node\ j \\ 0 & otherwise \end{cases}$$

$$y_{ij} = \begin{cases} 1 & if\ UAV\ travels\ from\ node\ i\ to\ node\ j \\ 0 & otherwise \end{cases}$$

$$S_{ij} = \begin{cases} 1 & if\ target\ i\ is\ served\ from\ \ node\ j \\ 0 & otherwise \end{cases}$$

$$Q_i = the\ consumed\ capacity\ of\ battery\ in\ node\ i$$

$$T_i = the\ access\ order\ of\ rendezvous\ node\ i$$

$$t_i = the\ departure\ time\ of\ rendezvous\ node\ i$$

$$s_i = the\ standing\ time\ of\ rendezvous\ node\ i$$

The 2E-GUCRP can be formulated as the following mixed integer programming in following model:

Minimize:

$$t_* \tag{0}$$

Subject to:

$$\sum_{i\in\{0\}\cup V_s} x_{ij} = \sum_{i\in\{*\}\cup V_s} x_{ji} \le 1, j \in V_s \tag{1}$$

$$\sum_{i\in V_s} x_{i*} = \sum_{i\in V_s} x_{0i} = 1 \tag{2}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

$$\sum_{i\in V_s} x_{*\mathrm{i}} = \sum_{i\in V_s} x_{i0} = 0 \quad (3)$$

$$N - 1 \geq T_i - T_j + N \times x_{ij}, i \in \{0\} \cup V_s, j \in \{*\} \cup V_s \quad (4)$$

$$\sum_{i\in V} y_{ij} = \sum_{i\in V} y_{ji} = 1, j \in V_t \quad (5)$$

$$M \times (1 - y_{ij}) \geq \left|C_j + d_{ij}/v_1 - Q_j\right| + \left|S_{ji} - 1\right|, i \in \{0\} \cup V_s, j \in V_t \quad (6)$$

$$M \times (2 - y_{ij} - S_{ik}) \geq \left|Q_i + C_j + d_{ij}/v_1 - Q_j\right| + \left|S_{ik} - S_{jk}\right|, \ i \in V_t, j \in V_t, k \in \{0\} \cup V_s \quad (7)$$

$$M \times (1 - y_{ij}) \geq (Q_i + d_{ij}/v_1 - \theta), i \in V_t, j \in \{*\} \cup V_s \quad (8)$$

$$M \times (2 - y_{ij} - S_{ik}) \geq \left|x_{kj} - 1\right|, i \in V_t, j \in \{*\} \cup V_s, k \in \{0\} \cup V_s, j \neq k \quad (9)$$

$$M \times (3 - y_{ij} - x_{kj} - S_{ik}) \geq (d_{kj}/v_2 - d_{ij}/v_1 - Q_i), i \in V_t, j \in \{*\} \cup V_s, k \in \{0\} \cup V_s \quad (10)$$

$$y_{ij} = 0, i \in \{0\} \cup \{*\} \cup V_s, j \in \{0\} \cup \{*\} \cup V_s \quad (11)$$

$$\sum_{i\in\{0\}\cup V_s} x_{ij} \leq \sum_{i\in V_t} y_{ij} + \sum_{i\in V_t} y_{ji}, j \in \{*\} \cup V_s \quad (12)$$

$$M \times (2 - y_{ij} - S_{ij}) \geq \left|s_j - Q_i - d_{ij}/v_2\right|, \ i \in V_t, j \in \{0\} \cup \{*\} \cup V_s \quad (13)$$

$$M \times (3 - y_{ij} - x_{kj} - S_{ik}) \geq \left|t_j - t_k - d_{ij}/v_1 - Q_i - s_j\right|, i \in V_t, j \in \{*\} \cup V_s, k \in \{0\} \cup V_s \quad (14)$$

$$M \times (1 - x_{ij}) \geq t_i - t_j + d_{ij}/v_1, i \in \{0\} \cup V_s, j \in \{*\} \cup V_s \quad (15)$$

$$M \times \sum_{i\in\{0\}\cup V_s} x_{ij} \geq t_j, \ j \in \{*\} \cup V_s \quad (16)$$

$$\sum_{j\in\{0\}\cup\{*\}\cup V_s} S_{ij} = 1, i \in V_t \quad (17)$$

$$0 \leq Q_i \leq \theta, \ i \in V \quad (18)$$

$$T_0 = 1 \quad (19)$$

$$t_0 = s_0 \quad (20)$$

Constraint (1) ensures that each optional stopping node can be accessed at most once, and the indegree and outdegree of the same node is equal except for the starting node and end node. The indegree of end node must be 1 which is equal to the outdegree of starting node. The outdegree of end node is also the same as the indegree of starting node, and they are both equal to 0. Constraint (4) is a typical Miller–Tucker–Zemlin subtour elimination constraints which makes sure that there is no subloop in GV's route through marking the accessing order of the rendezvous nodes. Constraint (5) guarantees that all target nodes are served once only.

Constraint (6) describes the taking off progress of UAV. When UAV takes off from node *i* and heads to the target node *j*, the S_{ji} should be equal to 1 and the consumed capacity of battery in node *j* should be equal to the sum of the duration of the flight and the duration of the service time in node *j*. Constraint (7) describes the flying progress between targets. When the UAV flies from target *i* to target *j*, the node *i* and node *j* should be in the same segment. Thus, if S_{ik} equals to 1 and y_{ij} equals to 1, then $S_{ik} = S_{jk}$. Under this circumstance, the consumed capacity of battery in node *j* should be equal to the sum of the duration of the flight, the consumed capacity of battery in node *i* and the duration of the service time in node *j*. Constraint (8) describes the landing progress. When the UAV lands at a rendezvous node *j*, the y_{ij} equals to 1. At this circumstance, the left-hand side of the inequality is zero. In order to meet the constraint, the following is true that $Q_i + d_{ij}/v_1 \leq \theta$, which means that the entire flight process in each UAV route will not exceed the endurance limit. Constraint (6) ~ (8) ensures the continuity of UAV's flight in each segment, and assigns a value to Q through those constraints, thus it ensures that each flight can meet the endurance capability.

The constraint (9) ensures that if the UAV lands at a certain rendezvous node, then this rendezvous node must be on the GV's route, which connects the decision variable x_{ij} to y_{ij}. The constraint (10) limits that the GV must arrive at the designated landing rendezvous node before the UAV. Constraints (11) ensures that UAV does not fly on the road network. The constraint (12) guarantees that if the vehicle has access to an optional stopping node, this node must be a rendezvous node at which the UAV takes off or lands.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

The constraint (13) is the calculation of the waiting time with considering the situation of the Holding Strategy in H5. On the basis of constraint (13), constraint (14) calculates the specific time of the GV leaving each rendezvous node. Constraint (15) is a complement to the previous constraint, ensuring the consistency of the time to leave the rendezvous node and the order of the rendezvous node.

Constraint (16) makes sure that the values of t for all unvisited stopping nodes are equal to 0. Constraints (17) ensures that each UAV's target node is assigned to a certain segment. Constraints (18) limits the consumed capacity of battery. Constraint (19) initializes the value of T in the starting point. Constraints (20) ensures a special scenario where the UAV needs to start from the starting point and land at the starting point.

The goal of optimization is to minimize t_*. As we mentioned above, {*} indicates the destination. Thus, the t_* indicates the total time of task completion. To minimize the t_* is equal to minimize the time of whole missions.

There are two main differences between this model and 2E-VRP model: (1) the shape of routes in each echelon is different: as we reviewed above, the 2E-VRP is a typical single/multiple-circulation-single/multiple-circulation problem while this model represents a partial-segment-single-segment problem; (2) the synchronization are much more complex: as one vehicle must wait for the other one in each segment, the more precisely time control method is applied with the setting of decision variables t_i and s_i.

This is a very complicated model, and the actual number of constraints is going to change in a power series growth with the scale of the problem. If we assume that the number of target nodes is n, the number of vehicle optional stopping nodes is N (including start and end), the problem contains, in fact, $3 \times N^2 \times n + 3 \times N^2 + N \times n^2 - 4 \times N \times n - n^2 + 5 \times n + 5$ constraints.

For example, constraint (1) contains N-2 constraints because constraint (1) has to limit all the j in V_s. Constraint (2) contains two constraints, which is actually a combination of $\sum_{i \in V_s} x_{i*} = \sum_{i \in V_s} x_{0i}$ and $\sum_{i \in V_s} x_{0i} = 1$. We analyzed all the constraints and constructed the following tables in sequence.

Table 3. Constraint quantity analysis

1	$N-2$	11	N^2
2	2	12	N-1
3	2	13	$n \times N$
4	$(N-1)^2$	14	$n \times (N-1)^2$
5	2×n	15	$(N-1)^2$
6	$(N-1) \times n$	16	N-1
7	$n^2 \times (N-1)$	17	N
8	$n \times (N-1)$	18	1
9	$n \times (N-1)^2 - n \times (N-2)$	19	1
10	$n \times (N-1)^2$	20	1
SUM:$3 \times N^2 \times n + 3 \times N^2 + N \times n^2 - 4 \times N \times n - n^2 + 5 \times n + 5$			

3.4. Verification through CPLEX

In order to verify the correctness of this model, a small-scale instance is constructed and then solved by the CPLEX software.

We randomly generated four targets and four optional stopping nodes in a 100×100 field, and the service time of target is randomly generated between 5 to 10 units. Their coordinates are as follows:

Table 4. Coordinates of random case

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

Optional Stopping Nodes	X	Y	Target	X	Y	Service Time
1	3.23	8.86	1	45.27	9.58	9.87
2	80.20	66.45	2	27.89	17.03	6.98
3	69.64	38.02	3	72.64	27.85	7.29
4	2.74	97.98	4	42.94	47.11	8.09

Assuming that the maximum range of the UAV's endurance is 100 units, the speed of the UAV is 2 units per second and the speed of the GV is 1 unit per second.

As shown in Figure 8, we mark the optional stopping nodes with red dots and the UAV targets with small blue circles. The first point in optional stopping nodes is specified as the starting point with green circle and the second point is the endpoint with blue circle.

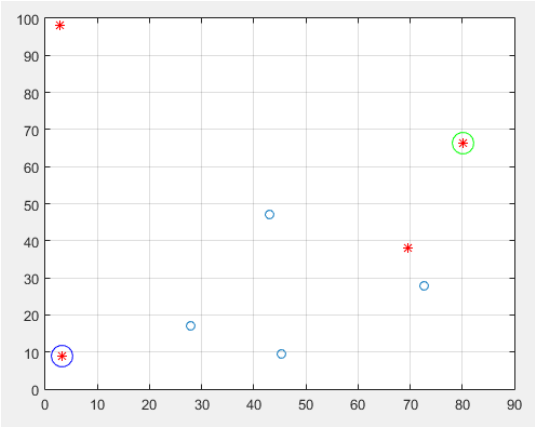

Figure 8. The position of small-scale instance

By using CPLEX software, we obtained a solution with the objective function value of 112.56 in 1.6 seconds. The UAV routes are represented as red dashed lines while GV routes are marked with green lines. The solution is as follows:

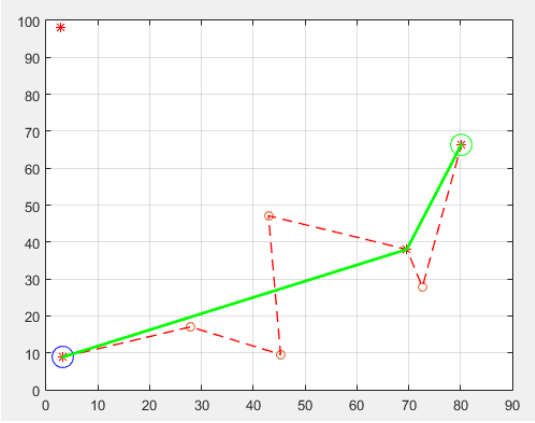

Figure 9. CPLEX solution

From the result, the model is verified and a good feasible solution is found through CPLEX software. However, according to our analysis on the scale of constraints, we cannot solve larger problems by using CPLEX software. In order to verify this hypothesis, more instances with different numbers of points are calculated and Figure 10 presents the calculation time in each instance.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

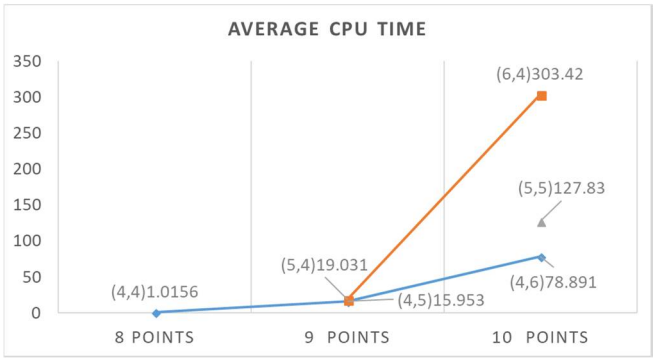

Figure 10. The Average Calculation Time in CPLEX

In this figure, the content in bracket before each data represents the size of the instances, in which the first number is the number of targets and the second number represents the number of optional stopping nodes. The blue line indicates the trend of calculation time with the increase of target points, while the red line indicates the trend of calculation time with the increase of optional stops nodes. When the number of N and n is fixed, we can calculate quantity of constraints according to the conclusion of TABLE 4. The results are shown in the table below.

Table 5. The growth of scale

The sum of points	Target	Optional stopping node	Constraint quantity	CPU time
8	4	4	249	1.0156
9	4	5	384	15.953
9	5	4	313	19.031
10	4	6	549	78.891
10	5	5	480	127.83
10	6	4	383	303.42

With the expansion of instance scale, the growth of computing time is very significant. We can learn from both the Figure 10 and Table 5 that increasing the number of targets will significantly slow down the solution progress. This slowdown is not due to the increase in the scale of constraints, but because it becomes more difficult to deal with the constraints associated with the targets. In fact, when setting 8 targets and optional stopping nodes, the solving time is more than 1 hour. Thus, it is impossible to use CPLEX software to solve a problem more than 10 targets in a short time.

From the above experiments, we can conclude that, for medium-sized experiments, it is impossible to directly use commercial software to find an optimal solution in a short time. Therefore, in the next section, we improve two existing 2E-VRP algorithms to quickly construct a feasible solution for this problem.

4. Heuristics

This section presents the heuristics which could be applied to solve both 2E-VRP and our 2E-GUCRP. The "Drone First Truck Second (DFTS)" is proposed by Ha et.al [9] to solve the drone delivery problem. We modify the heuristics and apply them to the 2E-GUCRP. Furthermore, we proposed a Clustered Assignment (CA) Heuristic based on the analysis of the DFTS. The CA heuristic is constructed in two phases: in the first phase, we cluster the targets and construct a feasible segment in each cluster; in the second phase, we use two proposed connecting algorithms to connect the segments as a feasible solution.

4.1. Drone First Truck Second (DFTS) Heuristic

This heuristic, also known as Split Algorithm, was first proposed by Prins [22] in 2004 to solve vehicle routing problems. And a survey of this heuristic is also presented by Prins [23]. Murray and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

Chu [4] and Ha et.al [9] applied this heuristic into the drone delivery problem. Split heuristic is also capable to solve a 2E-VRP. Zeng et.al [24] solved the 2E-VRP with a hybrid GRASP+VND heuristic which is a combination of an improved Split algorithm and neighborhood search. This paper is basically consistent with the methods in Ha et.al [9], but with more than one targets in one UAV route, the main difference is the number of target points of the UAV during the split process. The main flow of heuristic are as follows:

Heuristic 1: Split

```
Bigtour ← TSP(startpoint, endpoint, UAVnode) , i ← 1, flag ← 1
WHILE i < n + 1
    subtour ← HamiltonRoute(flag, Bigtour, i)
    IF subtour. length > θ
        Tour ← ConstructTour(subtour)
        flag ← i
        subtour ← flag
    END IF
    i ← i + 1
END WHILE
```

We firstly construct a big tour include all target, starting point and end point by solving a Travelling Salesman Problem (TSP) among them. Then split the big tour in order from the starting point. In each subtour we split, there are as many target points as possible under the constrain of endurance. Each subtour represents a segment in final solution, which includes one or two rendezvous nodes and a series of targets. The rendezvous nodes are selected by a greedy principle that we always find the nearest optional stopping nodes for the first and last target points in each segment.

After the split process, we got a number of viable segments. However, they might be not connected as the last rendezvous node in the former segment may not be the first rendezvous node in the later segment as we discussed in section 3. Thus, we need a connecting algorithm to construct a feasible solution. This problem cannot be converted to TSP by adding virtual points, as we don't know which rendezvous node should be selected as the starting node for each segment. To solve this problem, we present a Queue-Based Construct (QBC) Algorithm to connecting the separated segments.

Queue-based Construct Algorithm

```
Queue ←< startpoint, endpoint >
WHILE NotIsEmpty(Tour)
        Queue ← FindMinInQueue(Tour)
END WHILE
```

QBC Algorithm is designed by a greedy thought. We firstly add the starting point and end point into the queue. Then add one segment at a time by find the min distance between the head or end of the queue and the rendezvous node in the out-queue subtour. If there are no intersection of rendezvous nodes between the queue and out-queue subtour, we add an empty edge segment to keep the connection.

4.2. Clustered Assignment (CA) Heuristic

 doi:10.20944/preprints201902.0110.v1

This heuristic is firstly presented by Crainic et.al [25] to solve the Two-Echelon Vehicle Routing Problem. We propose a variation to better solve our problem. In our variation, we apply the hierarchical clustering method to divide the target points into different cluster, and make sure each cluster is a feasible division of segment. The main flow of heuristic are as follows:

Heuristic 2: Clustered Assignment

```
flag ← minClusterNumber
WHILE flag ≤ n
    Cluster ← KClustering(UAVnode, flag)
    Clustered ← HamiltonRoute(Cluster)
    IF isFeasiable(Clustered)
        RETURN Clustered
    ELSE
        flag ← flag + 1
    END IF
END WHILE
```

We choose the hierarchical clustering method to cluster the target points. Hierarchical clustering analysis method is one of the essential mathematic methods in produce practice. The custom hierarchical clustering is usually constructed in two phases: firstly, decide the definition of distance and construct a vertical icicle or an agglomeration schedule through the agglomeration progress; secondly, decide the cluster number and final cluster through the analysis of vertical icicle.

In CA heuristic, we firstly set a minimum cluster number, and cluster the target point into clusters. For each cluster, try to build a minimum route to cover all target points through solving a variation of Hamilton Route Problem, which is also known as the Open Travelling Salesman Problem (OTSP). After the OTSP answer constructed, we check whether the minimum route in each cluster is capable to construct a feasible segment. If there are any clusters that cannot construct a feasible segment, we add up the cluster number and re-cluster the target points until each cluster is in line with the constraints.

The feasible segments that are constructed by CA heuristic may not be interconnected with each other. Thus, we also need to construct a feasible solution by adding empty edge segments through QBC algorithm.

5. Computational Experiment

In this section, we present two randomly generated instances and a practical instance of Changsha city. For each instance, two different models, 2E-VRP and 2E-GUCRP are respectively studied and solved by the heuristics mentioned above. The computational experiment indicates that the 2E-GUCRP obtains a better efficiency. CA performs batter for 2E-VRP, while Split is more suitable for 2E-GUCRP. Further discussion of the instance in Changsha points out that Holding strategy may be a better strategy when the ground vehicle is fast enough.

5.1. Experimental Environment and Parameter Setting

Both heuristics were implemented in MATLAB R2015a 8.5.0. All of the experiments were carried out in a DELL desktop with a four core i7 3.7GHz processor, 16GB of ram, and Windows 10 operating system. As far as we known, there are no public test instance for 2E-GUCRP. Thus, we apply two different methods to generate the instances for testing. In both methods, the nodes are generated in a square area with a length of 100 units. In the first method, the positions of all rendezvous nodes

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

and targets are randomly generated in a two-dimensional square space. In order to be realistic, the distances for the GV among the rendezvous nodes are Manhattan distance, while the distance of the UAV's flight is calculated according to the straight-line distance. The data generation progress in the second method is shown in Figure 11. Initially, N nodes is randomly generated as road intersections, which is represented as red asterisk in Figure 11(a). Then at least 2N edges are selected and marked as red dotted line to form a road network. On this basis, the stopping points are built randomly on the existing roads and marked as blue circle. Finally, UAV target points, which are green circles in Figure 11(d), are generated randomly in the square space.

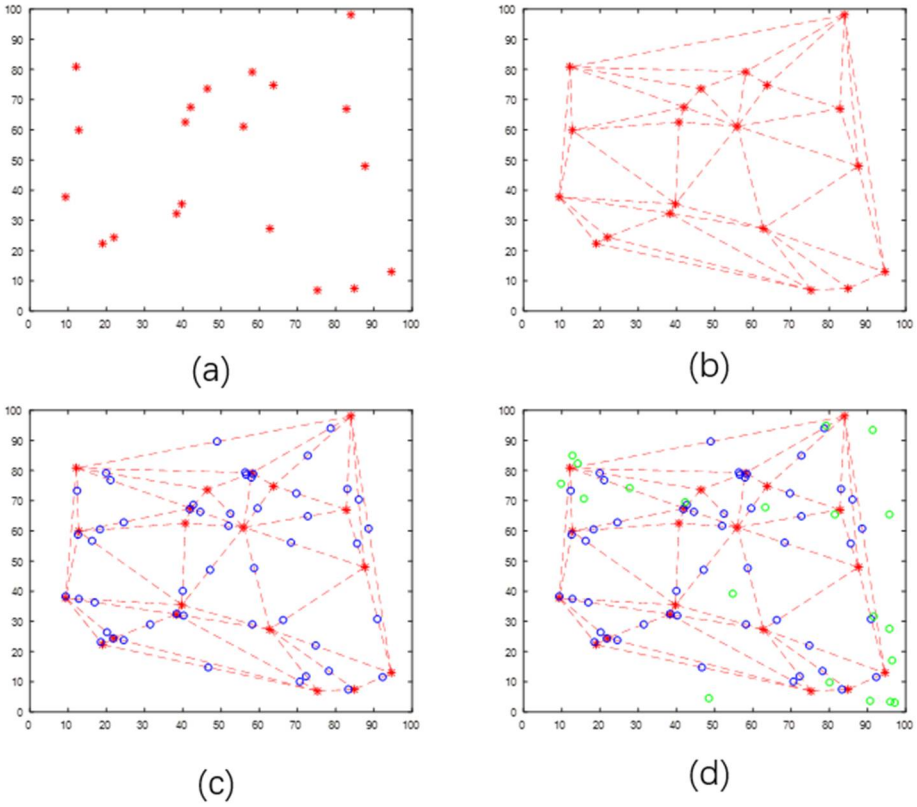

Figure 11. Data Generation Progress

The main difference between the two data generation methods is the distance calculation between rendezvous nodes. The first method is to construct a complete graph of rendezvous nodes, while not any two nodes in the second method can be reached directly, thus the shortest distance between any two points needs to be computed based on the adjacency matrix. Thus, in our point of views, the second method is closer to reality. On the other hand, the second method also avoids the aggregation of optional stopping points in a certain area, which makes the distribution more widely dispersed.

The scales of instances are as follow:

Table 6. The scale of instances

Num.	Targets	Optional stopping node
1	12	12
2	20	12
3	20	20
4	40	20
5	40	40
6	100	40

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

In each scale, 100 instances are generated with two different method, the first method is noted as Dataset 1 while the second one is Dataset 2. Both two datasets are solved by the proposed heuristics separately. In each instance, the service time of each target is uniformly distributed between 5 and 10-time units and specify that the first point in optional stopping nodes is the starting point and the second point is the endpoint.

There are also some parameters about the UAV and the GV. The UAV has 100 units of maximal working time with the fixed speed of 2-unit distance per unit time, while the GV travels with a fixed speed of 1-unit distance per unit time.

5.2. The Experiment of Dataset 1

All the instances in Dataset 1 is formulated by the first method. We calculate the average calculational time of each heuristic and compare their solutions on the object function value. Among all 100 instances in each scale, we count the times in which the combination of heuristic and model performs better than others. The results are as follows:

Table 7. The Result of Dataset 1

		Avg. Value	**Best times**	**Avg. CPU time**	**Avg. Value**	**Best Times**	**Avg. CPU time**
		Scale 1			Scale 2		
2E-VRP	Split	552.46	0	**0.0230**	566.51	0	**0.0219**
	CA	519.68	1	0.0242	534.06	0	0.0227
2E-GUCRP	Split	**415.09**	**57**	0.0233	**421.79**	**51**	0.0244
	CA	426.21	44	0.0239	424.40	50	0.0245
		Scale 3			Scale 4		
2E-VRP	Split	734.74	0	**0.0305**	741.33	0	**0.0308**
	CA	704.14	0	0.0359	697.91	0	0.0378
2E-GUCRP	Split	**544.63**	**62**	0.0338	**553.93**	**60**	0.0339
	CA	568.55	38	0.0380	570.40	40	0.0353
		Scale 5			Scale 6		
2E-VRP	Split	1079.91	0	**0.0477**	1071.33	0	**0.0442**
	CA	1031.10	0	0.0602	1022.87	0	0.0583
2E-GUCRP	Split	**820.02**	**74**	0.0555	**825.23**	**57**	0.0506
	CA	875.76	26	0.0617	854.38	43	0.0628

The computational results obtained by two proposed heuristics for two models are presented in Table 7. Column 'Avg. Value' indicates the average objective value among all 100 instances. Column 'Best Times' reports the comparation among different combinations of heuristics and models. It indicates how many times that the solution is better than others. In some special instances, it is possible for two heuristics to obtain the same solution, so the sum of this column values may be greater than 100. Column 'Avg. CPU time' give the average CPU time of solving one instance.

The results of 2E-GUCRP model are better than those of 2E-VRP in both average objective value and the number of best times. There is only one instance in Scale 1, in which both two models obtain the same solution. As for the algorithm, Split is more suitable for solving 2E-GUCRP, and CA is more suitable for solving 2E-VRP. Generally speaking, the heuristics can solve any one of the randomly-generated instances in less than 0.1 s, and the solution time of the Split algorithm is slightly shorter than Clustering algorithm.

To better compare the performance differences between the two models, we calculated the GAP value in each case. The calculation is as follows:

$$\mathrm{GAP} = \frac{\mathrm{BestVRPsolution} - \mathrm{BestGUCRPsolution}}{\max\{\mathrm{BestVRPsolution}, \mathrm{BestGUCRPsolution}\}} \times 100\%$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

The GAP value indicates the percentage of time saved in each instance. The average GAP value of Dataset 1 is presented in TABLE 8, and we use a boxplot graph to measure the level of dispersion of all instances.

Table 8. The average gap value in Dataset 1

Scale.	1	2	3	4	5	6
Average Gap	22.79%	24.45%	24.48%	22.73%	20.83%	20.36%

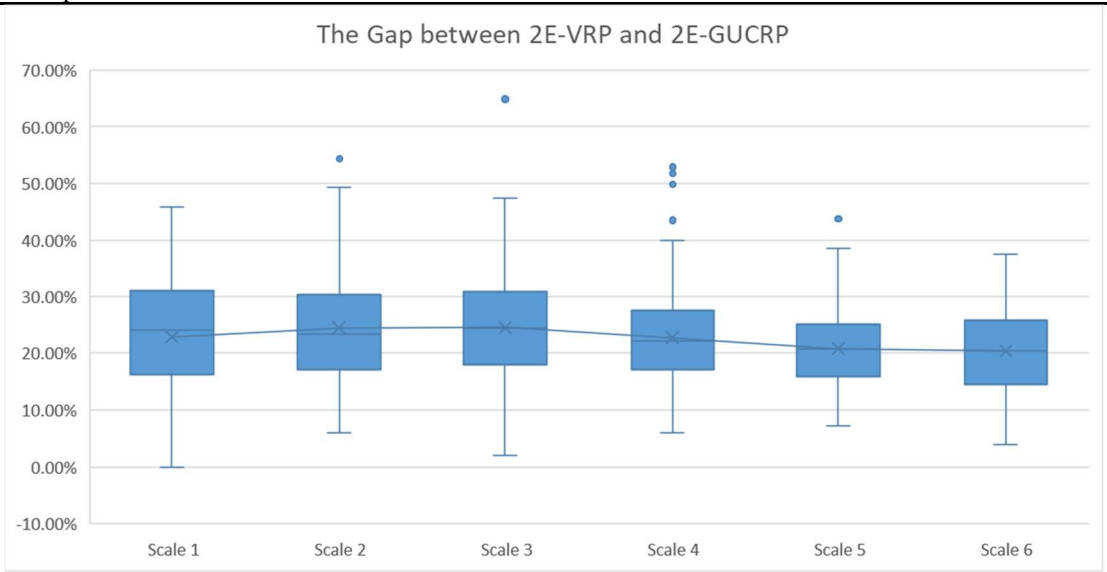

Figure 12. Dataset 1 GAP Boxplots

In general, the GUCRP saved an average of more than 20 percent of the time. As the instance scale changes, we can see that the range of GAP is narrowing. In the worst instance, there is no gap between the two models. In the best instance, the gap between the two models is more than 60%. But overall, only a fraction of the results are more than 40 percent.

The results of Dataset 1 indicate that the 2E-GUCRP is a more efficient model. But with the generation process of Dataset 1 completely random, different instances presents the different characteristics, and the proposed heuristics also performs with its own merits. In order to obtain more reliable conclusions, we conducted further experiments on the second dataset.

5.3. The Experiment of Dataset 2

All the instances in Dataset 2 is formulated by the second method in which the positions of all rendezvous nodes and targets are crossly divided. We calculate the average solving time and compare their solution as what we did in the experiment of Dataset 1. The results are as follows:

Table 9. The Result of Dataset 2

		Avg. Value	Best times	Avg. CPU time	Avg. Value	Best Times	Avg. CPU time
		Scale 1			Scale 2		
2E-VRP	Split	521.24	0	**0.0222**	695.44	0	**0.0350**
	CA	497.90	0	0.0256	650.75	0	0.0397
2E-GUCRP	Split	402.41	44	0.0261	526.37	**50**	0.0383
	CA	**390.31**	**57**	0.0273	**526.14**	**50**	0.0405
		Scale 3			Scale 4		
2E-VRP	Split	696.66	0	**0.0303**	1053.59	0	**0.0511**
	CA	655.79	1	0.0380	1004.95	0	0.0670

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

2E-GUCRP	Split	541.22	49	0.0344	**831.38**	**50**	0.0570
	CA	**536.42**	**51**	0.0373	840.45	**50**	0.0655
		Scale 5			Scale 6		
2E-VRP	Split	1044.71	0	**0.0539**	1866.58	0	**0.1156**
	CA	986.18	0	0.0673	1806.01	0	0.1759
2E-GUCRP	Split	**821.86**	**52**	0.0587	**1549.09**	**50**	0.1205
	CA	826.98	48	0.0717	1554.29	**50**	0.1777

The computational results obtained by the proposed heuristics are presented in Table 9. On the comparison of two models, the same conclusion can be drawn that the 2E-GUCRP model is more efficient. There is also one special instance in Scale 3, in which both two models obtain the same solution. However, on the performance for two heuristics, the results of the Dataset 2 are significantly different with the result of Dataset 1. There is no significant difference between Split heuristic and CA in their performance on Dataset 2. The main reason for this phenomenon is different distributions of optional stopping nodes. In the second data set, the distribution of optional stop nodes is more dispersed due to the existence of road network. Therefore, the frequency of Holding Strategy we mentioned above is reduced.

Table 9. The average gap value in Dataset 2

Scale.	**1**	**2**	**3**	**4**	**5**	**6**
Average Gap	22.97%	22.52%	20.88%	18.88%	19.34%	15.59%

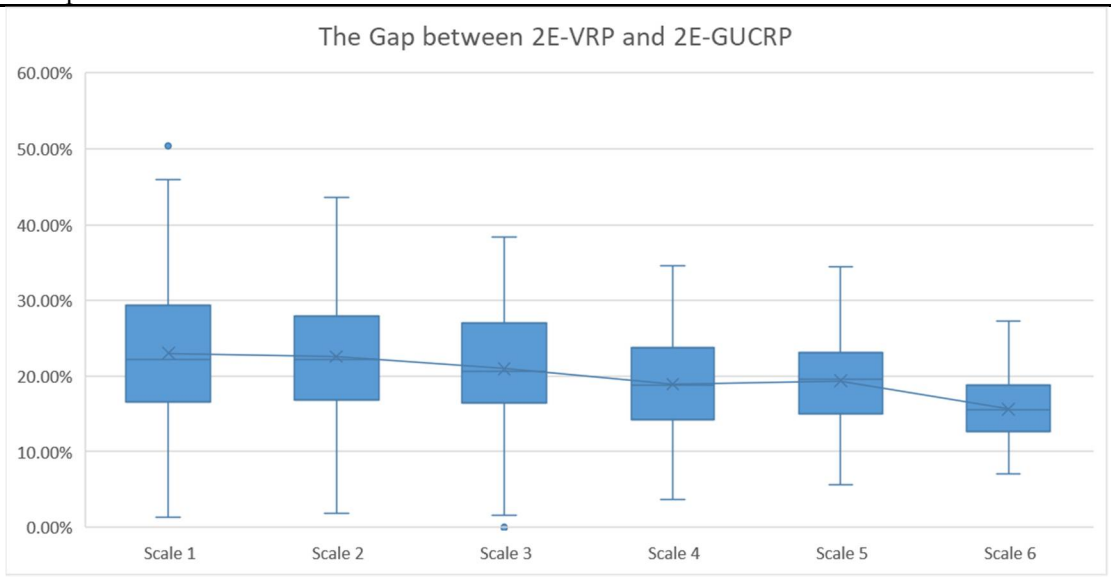

Figure 13. Dataset 2 GAP Boxplots

In general, the GUCRP saved an average of more than 15 percent of the time. However, as the scale increases, the trend of gap narrowing is more obvious. In the worst instance, there is no gap between the two models. In the best instance, the gap between the two models is more than 50%. But overall, only a fraction of the results are more than 30 percent.

The results on the second dataset also verify that the proposed model is more effective in solving the routing problem of UAV ISR missions. The heuristics we proposed vary in their performance. In practical application, it is suggested that two heuristics be solved respectively, and then the better solution should be taken as the solution of the problem.

5.4. The practical instance in Changsha city

5.4.1. Design of the practical instance

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

In recent years express delivery in China is growing fast and multi-echelon distribution systems have been used in some express companies, such as SF Express. Thus, we take Changsha, a fast-growing city in China, as an example to verify the validity of the proposed model and heuristics.

Figure 14 presents the progress of data extraction. Firstly, the main roads of the city are selected and the coordinates of road crossings are obtained from the GPS system. To simplify the problem, each road is represented as a line segment, which is shown in the second graph. Furthermore, three districts are included in this map: Yuelu district with blue background, Kaifu district with the green background and Furong district with yellow background. In the following experiment, not only the whole Changsha city is applied as a large-scale application case, but three districts are also considered separately.

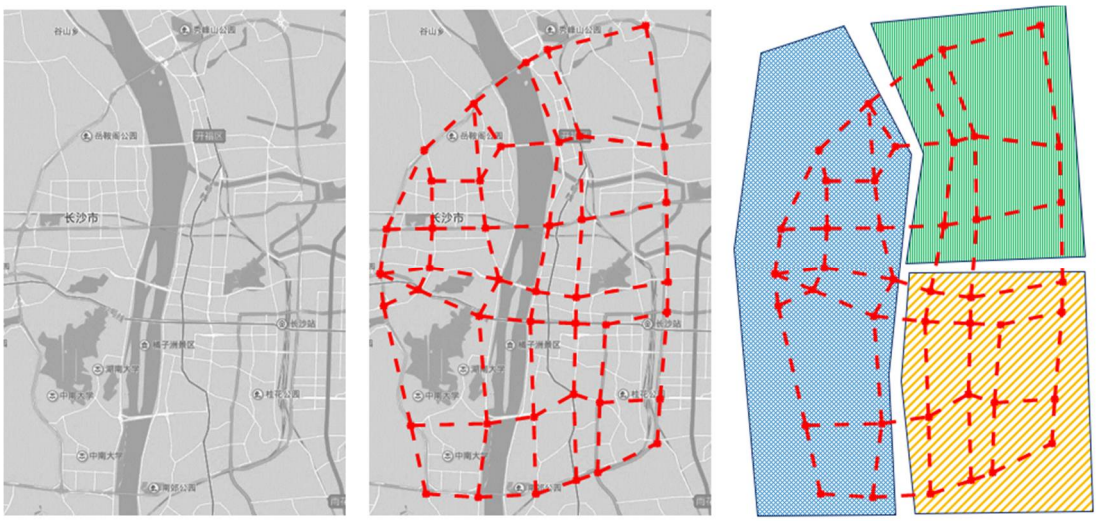

Figure 14. Main Road of Changsha

After building the road network, we assume that optional stop nodes are on the road, and there is only one optional stop node in each road. For node i on the edge $< a, b >$, a ratio $0 \leq \lambda_i \leq 1$ is set to measure the distance, then the distance from the node i to the end of the line is $\lambda_i \times \boldsymbol{d_{ab}}$, and the distance to the other end is $(1 - \lambda_i) \times \boldsymbol{d_{ab}}$. Based on the structure of the road network in Figure 14, we can calculate the shortest distance between any two optional stop nodes.

Then the locations of targets are selected based on some key landmarks in Changsha, where the UAV needs to hover for a period of time (assuming 2-5 minutes) to complete the reconnaissance mission. The number of targets and stopping nodes are presented in the following table. It should be noted that since the division of the region is not absolute, some target points can belong to both regions at the same time. Therefore, the sum of the numbers of each region is greater than the total number.

Table 10. The Number of Targets and optional stopping nodes

Area	Changsha city	Yuelu district	Furong district	Kaifu district
Number of Targets	40	20	16	12
Number of Optional Stopping Nodes	60	25	30	20

As for the parameters of the UAV, we selected three different types of DJI UAVs, which are all supported by batteries. The actual maximum speed and flight time is shown in TABLE XII. Since we assume that the UAV travels at a uniform speed, some adjustments are applied to generate the tentative data, which is also listed in TABLE 11. And in the absence of congestion, the average speed of the ground vehicles is set as 50km/h on the main road in the Changsha.

638 **Table 11.** The UAV related data

Version	Actual Data		Tentative Data	
	MAX Speed	MAX Flight time	AVG Speed	MAX Flight time
PHANTOM	72km/h	Approx. 30 minutes.	45km/h	25 minutes
MAVIC	65km/h	Approx. 25 minutes.	36km/h	20 minutes
SPARK	50km/h	Approx. 15 minutes.	24km/h	15 minutes

639 5.4.2. The experiment of the practical instance

640 In the experiments in each region, we run both two algorithms 10 times separately and select the
641 best results to report. The results are shown in TABLE 12. Due to the limitation of the endurance, the
642 type SPARK can only complete part of the missions, so it is only used in the Kaifu district and Furong
643 district instances.

644 **Table 12.** The results of practical instance

Area	PHANTOM			MAVIC			SPARK		
	2E-VRP	2E-GUCRP	GAP	2E-VRP	2E-GUCRP	GAP	2E-VRP	2E-GUCRP	GAP
Changsha	298.79	253.76	15.07%	332.43	286.93	13.69%	\	\	\
Yuelu	152.80	121.42	20.53%	154.61	127.15	17.76%	\	\	\
Furong	104.46	84.91	18.72%	115.58	96.88	16.18%	125.24	114.87	8.28%
Kaifu	103.62	82.47	20.41%	119.12	92.52	22.33%	138.48	138.48	0%

645 In order to make the data easy to read, the unit of the data shown in the table is minutes. It costs
646 only 253.76 minutes using the PHANTOM to visit the 40 target points in the city of Changsha, which
647 is realistic. Consistent with randomized experiments, the results of 2E-GUCRP model are better than
648 those of 2E-VRP model. The gap, however, has narrowed. Compared with the gap of 18.88% in a
649 similar scale of instances in Dataset 2, the gap between the two models in the instance of Changsha
650 City are 15.07% (PHANTOM) and 13.69% (MAVIC).

651 In order to further explore the reason of the gap narrowing, we draw the solution of the instance
652 in Yuelu district, for which three reasons are considered: (i) The number of the points is relatively
653 suitable. There are too many points in the whole Changsha city, and the solution is too complex to be
654 read; (ii) The size of this area is appropriate. UAV will not visit too many targets in one flight nor
655 serve only one target at a time. (iii) The solution gap between different types of UAV are small, which
656 is worth further discussion.

657 Figure 15 shows the solutions by using the PHANTOM, while Figure 16 displays the solutions
658 by the MAVIC. The black triangle represents the starting point, while the inverted black triangle is
659 the end point. The red dotted line represents the road network with intersection of the road as red
660 star. The blue circle represents the optional stopping nodes on the road. The target point is marked
661 as green circle. The blue solid line in the figure represents the GV route, while the green dotted line
662 indicates the UAV route.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

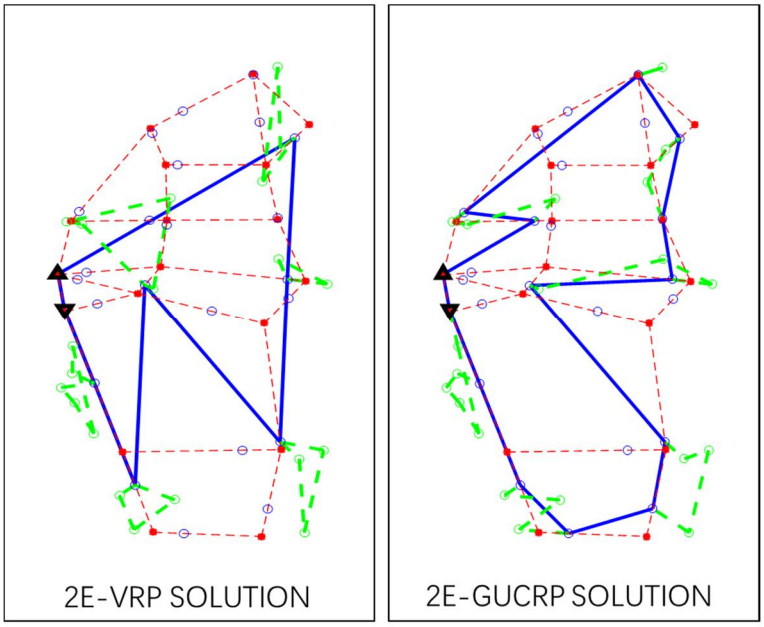

Figure 15. Solution in Yuelu district (PHANTOM)

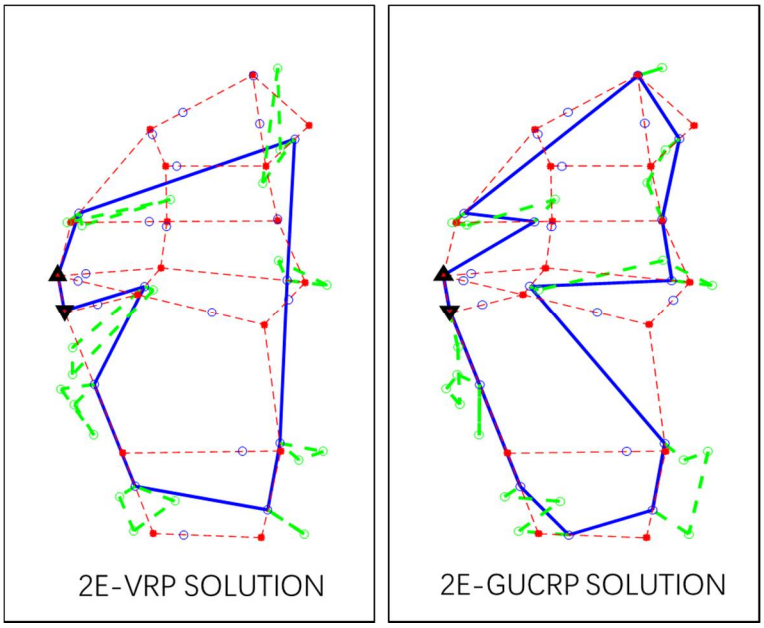

Figure 16. Solution in Yuelu district (MAVIC)

From the visual judgment of the route, the heuristics we proposed is capable to find an acceptable solution. However, this is obviously not an optimal solution, because there is an intersection in the route of the UAV. The reason is that the greedy TSP algorithm is designed without considering the intersection problem.

From the perspective of different UAVs, the larger UAV reduces the number of takeoff and landing due to its advantage of endurance. Thus, the overall completion time of the mission is effectively reduced (see the southeast corner and the middle position on the west in figure 15 and figure 16). The advantages of endurance would not be obvious in the instance with relatively sparse target points. But from TABLE 12, we can see that when the density of targets increases, the UAV can access more targets each fight, and the advantage is more obvious.

From the perspective of model differences, the flying distance of UAV in 2E-GUCRP is increased. However, due to the synergy of GV, the time of stagnation has been reduced. Therefore, the overall

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 February 2019 doi:10.20944/preprints201902.0110.v1

operation efficiency is better than 2E-VRP. Nevertheless, we can still find out the situation that the GV stays in place waiting for the UAV in the 2E-GUCRP solution (see the northernmost target point in Figure 15 and Figure 16). To further analyze the reasons for this situation, we might as well establish a general mathematical model for representation. For a flight, we assume that the optimal route of visiting the target points has been obtained, which is $Perm_a = \{N_1, n_1, n_2, \dots, n_x, N_2\}$. The total time to complete this route is:

$$NonStop = max\{\frac{d(N_1, N_2)}{\boldsymbol{v_1}}, \frac{\sum_i^{x+1} d(Perm_i, Perm_{i+1})}{\boldsymbol{v_2}}\}$$

And if we assume that the GV stays at N_1, waiting for the drone to come down, and then travels to the next location N_2, then the process will be calculated as:

$$Holding = \frac{\sum_i^{x} d(Perm_i, Perm_{i+1}) + d(n_x, N_1)}{\boldsymbol{v_2}} + \frac{d(N_1, N_2)}{\boldsymbol{v_1}}$$

The difference of the formulation before is:

$$Difference = Holding - NonStop$$

When the vehicle arrives later, the value of Difference is always positive, which means that the Holding Strategy takes more time. However, when the vehicle arrives earlier, the value of Difference is:

$$Difference = \frac{d(n_x, N_1)}{\boldsymbol{v_2}} + \frac{d(N_1, N_2)}{\boldsymbol{v_1}} - \frac{d(n_x, N_2)}{\boldsymbol{v_2}}$$

When $\boldsymbol{v_1}$ is less than or equal to $\boldsymbol{v_2}$, the value of Difference is always positive according to the principle that the sum of the sides of a triangle is greater than the third side.

$$Difference \geq \frac{d(n_x, N_1) + d(N_1, N_2) - d(n_x, N_2)}{\boldsymbol{v_2}} \geq 0$$

When $\boldsymbol{v_1}$ is greater than $\boldsymbol{v_2}$, this inequality is not true. Therefore, Holding Strategy is not only shown in the situation where there are no enough stopping nodes, but also may be shown in the situation where the ground is unobstructed and the driving speed of the vehicle is higher than that of the UAV.

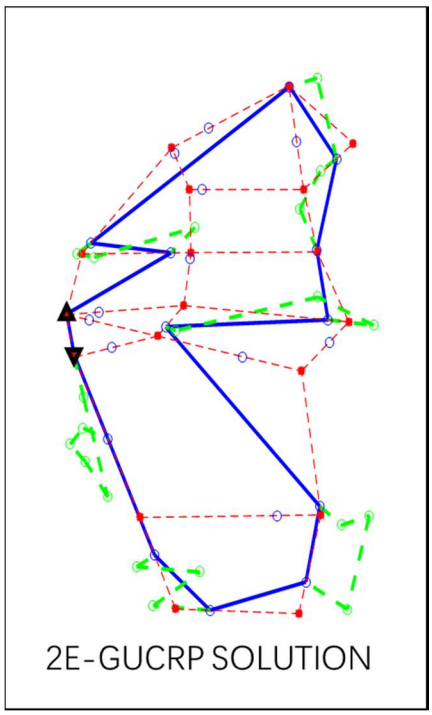

Figure 17. Solution in Yuelu district (PHANTOM) with traffic congestion

Back to the instance of Yuelu district, if we assume there is traffic congestion and the average driving speed of the GV is reduced to 40km/h, then we can get the solution above. At this time, we can find that there is no longer Holding Strategy in northern region, which verifies our analysis above.

In general, 2E-GUCRP is better than 2E-VRP when implementing the cooperative ISR missions of GV and its mounted UAV. However, there are two situations in which the Holding Strategy in the 2E-VRP is worth adopting: this strategy would be adopted to obtain a feasible solution when there are not enough stopping points on the road network; the other situation is that Holding Strategy may lead to better solutions when the ground vehicle travels faster than UAV.

When it comes to choosing a UAV type, the larger one can always complete the missions in a shorter time with the advantage of endurance. However, the efficiency of larger UAV is limited by the size of the mission and the sparse situation of the target point. For small-size mission or scattered targets, the smaller UAV may accomplish missions in a more cost-effective way. The analysis of cost is another interesting topic, which may deserve further discussion in the future research.

6. Conclusion

In this paper, we summarize and reclassify the previous literatures about the GV-UAV cooperated working model through the analysis of the routes shape in different echelons. We present a 2E-GUCRP through the systematic analyses on hypotheses. The general hypotheses and the extreme cases of segments are discussed with practical needs. Through the discussion of hypotheses, the MIP model of 2E-GUCRP is built and verified by the CPLEX software in small-scale instances. With the power level growth of the constrains, it is inefficient to solve the problem through the CPLEX. We modify the previous researches and solve this problem with Split and CA heuristics.

Further research on the solution methodology may focus on neighborhood search algorithms. With the different aspects in constrains, the removal and repair operators will be modified corresponding the specified constrains. And more discussions about the practical application will emerge in future researches.

Acknowledgment

The research is supported by National Natural Science Foundation of China (no. 71771215 and no. 71471174)

References

1. (2017, UPS tests drone delivery system. Available: http://www.businessinsider.com/ups-tests-drone-delivery-system-2017-2
2. (2016, DJI Kicks Off 2016 DJI Developer Challenge. Available: http://www.dji.com/newsroom/news/dji-kicks-off-2016-dji-developer-challenge
3. S. G. Manyam, D. W. Casbeer, and K. Sundar, "Path planning for cooperative routing of air-ground vehicles," presented at the American Control Conference, 2016.
4. 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.
5. N. Agatz, P. Bouman, and M. Schmidt, "Optimization Approaches for the Traveling Salesman Problem with Drone," *Social Science Electronic Publishing*, 2015.
6. Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "Heuristic methods for the Traveling Salesman Problem with Drone," *Computer Science,* 2015.
7. Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, "On the Min-cost Traveling Salesman Problem with Drone," *Transportation Research Part C Emerging Technologies,* vol. 86, pp. 597-621, 2018.
8. Z. Y. Zeng, X. U. Wei-Sheng, and X. U. Zhi-Yu, "Modeling and Multi-start Variable Neighborhood Descent Solution of Two-echelon Open Vehicle Routing Problem," *Computer Science,* 2014.
9. K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle Routing Problems for Drone Delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* pp. 1-16, 2016.

10. X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: several worst-case results," *Optimization Letters,* pp. 1-19, 2016.
11. S. Poikonen, X. Wang, and B. Golden, "The vehicle routing problem with drones: Extended models and connections," *Networks,* vol. 70, pp. 34-43, 2017.
12. X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: several worst-case results," *Optimization Letters,* vol. 11, pp. 679-697, 2017.
13. N. Mathew, S. L. Smith, and S. L. Waslander, "Planning Paths for Package Delivery in Heterogeneous Multirobot Teams," *IEEE Transactions on Automation Science & Engineering,* vol. 12, pp. 1298-1308, 2015.
14. M. S. B. Othman, A. Shurbevski, H. Nagamochi, M. S. B. Othman, A. Shurbevski, H. Nagamochi, *et al.*, "Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle," *Journal of Information Processing,* vol. 25, pp. 655-666, 2017.
15. H. Savuran and M. Karakaya, "Route Optimization Method for Unmanned Air Vehicle Launched from a Carrier," *Lecture Notes on Software Engineering,* vol. 3, pp. 279-284, 2015.
16. H. Savuran and M. Karakaya, "Efficient route planning for an unmanned air vehicle deployed on a moving carrier," *Soft Computing,* vol. 20, pp. 2905-2920, 2016.
17. 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, 2017.
18. C. Rose. (2013, Amazon's Jeff Bezos looks to the future. Available: http://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/
19. M. Drexl, "On Some Generalized Routing Problems," Rheinisch-Westf¨alischen Technischen Hochschule Aachen, 2007.
20. M. Drexl, "Synchronization in Vehicle Routing---A Survey of VRPs with Multiple Synchronization Constraints," *Transportation Science,* vol. 46, pp. 297-316, 2012.
21. M. Drexl, "Applications of the vehicle routing problem with trailers and transshipments," *European Journal of Operational Research,* vol. 227, pp. 275-283, 2013.
22. C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research,* vol. 31, pp. 1985-2002, 2004.
23. C. Prins, P. Lacomme, and C. Prodhon, "Order-first split-second methods for vehicle routing problems: A review," *Transportation Research Part C Emerging Technologies,* vol. 40, pp. 179-200, 2014.
24. Z. Y. Zeng, W. S. Xu, and Z. Y. Xu, "A two-phase hybrid heuristic for the two-echelon vehicle routing problem," in *Chinese Automation Congress,* 2014, pp. 625-630.
25. T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei, "Clustering-Based Heuristics for the Two-Echelon Vehicle Routing Problem," 2008.