A Hybrid Large Neighborhood Search Algorithm for Solving the Multi Depot UAV Swarm Routing Problem

XIAOHUI LI, PEIFAN LI, YI ZHAO, (Member, IEEE), LU ZHANG, AND YUAN DONG
School of Electronics and Control Engineering, Chang’an University, Xi’an 710064, China
Corresponding author: Yuan Dong (yuandong@chd.edu.cn)

This work was supported in part by the Fundamental Research Funds for the National Key Research and Development Program of China under Grant 2020YFB1600400, in part by the Key Project of National Internet of Things Integrated Innovation and Integration under Grant 2018-470, and in part by the Shaanxi Key Project of Research and Development Plan under Grant 2019ZDLGY03-01.

ABSTRACT This paper focuses on a modified Multi-Depot Unmanned Aerial Vehicle Routing Problem (MMDUA VRP). Comparing to classic multi-depot vehicle routing problem, our studied problem has no constraints to restrict the depot where the Unmanned Aerial Vehicle (UAV) departs and returns. This work aims to minimize the number of UAVs and total distance traveled by all UAVs. This problem is mathematically formulated in this paper and a heuristic-assignment based hybrid large neighborhood search (HLNS) is proposed to solve it. Extensive computational experiments are conducted to verify the performance of HLNS. The HLNS algorithm was first tested on Multi Traveling Salesman Problem which is a simplified version of the MMDUA VRP, and high quality solutions have been obtained. Experimental results by compared with CPLEX and other well-known algorithms suggest that our proposed algorithm provides better solutions within a comparatively shorter period of time. In addition, we also conduct sensitivity analysis on the location of depots and task points that may affect the total cost of solution.

INDEX TERMS Unmanned aerial vehicle, routing problem, large neighborhood search, local search.

I. INTRODUCTION
Application of unmanned aerial vehicles (UAVs) have boomed in variety of domains which gained more and more attention. The UAVs were initially applied in military missions and rapidly expanded to other fields in recent years due to the continuous advances in this technology. Now, UAVs are increasingly being used in logistics, military operations, public security, traffic surveillance and monitoring according to Macrina et al. [1]. Among these, patrol is one of the most promising areas. There are many advantages of using UAV for patrol. UAV could work in complex environments and has advantages that traditional transportations don’t have. For example, UAV could fly over rugged landscapes and traffic congested areas. In practical, route planning is an important component while using drones for patrolling. Reasonable route planning could reduce the whole number of UAVs used and complete the mission with a shorter flight distance. In this paper, we mainly focus on route planning in UAV patrolling.

There are some works related to UAV patrolling as follows. An indoor autonomous patrol and surveillance system using UAV was proposed by Lee et al. [2]. Zhou et al. [3] mainly focused on a UAV patrol system which is based on panoramic image stitching and object detection. Dorling et al. [4] pointed out that the drones have the potential to reduce the cost and time in packages delivery and a number of large organizations have shown interest in drone delivery. UAVs are increasingly used for surveillance tasks due to their long endurance which was introduced in the work of Zaza and Richards [5]. Ries and Ishizaka [6] aimed to implement a decision support system for UAV routing in the context for maritime surveillance. Furthermore, a mathematical programming was proposed to establish a real-time adaptive routing system. Thakoor et al. [7] addressed the multi UAV routing problem where a set of UAVs need to collect information via surveillance of an area. Apart from route planning another important consideration is to optimize the scheduling
in a drone-based delivery system. For example, a heuristic was proposed to address the problem of delivery scheduling for drones in [8]. Manyam et al. [9] considered a persistent monitoring scenario with a set of task locations visited persistently by multiple UAVs. Coelho et al. [10] studied limited autonomy heterogeneous fleet routing problem considering multiple charging stations and respecting operational requirements.

In particular, a reasonable routing could save a lot of resources when the number of task points increases. This paper focuses on efficient route planning based on the need of many companies.

The route planning of the UAV could be considered as an extension of Vehicle Routing Problem (VRP), which is introduced by Dantzig and Ramser [11]. They tried to address the problem of delivering gasoline to service stations. There are multiple depots and the UAVs have a limited flight endurance in our problem compared to the classic VRP. The relevant literature will be briefly introduced below.

Gen and Syarif [12] formulated the problem of transportation raw materials from multiple depots to a set of plants, which is defined as the Multi Depot Vehicle Routing Problem (MDVRP) and proposed a heuristic approach based on the Clarke and Wright [13] saving method. Zhen et al. [14] investigated a practical problem in the last mile distribution: a multi-depot multi-trip vehicle routing problem with time windows and release dates. This problem aimed to put forward a set of trips for the fleet of vehicles supplied by different depots with considering some realistic situation, such as the customers’ time windows and release data of customers’ goods. The objective is to minimize total traveling time. The multi-commodity two-echelon capacitated vehicle routing problem with time windows (MC-2E-VRPTW) is studied in [15]. They take customer-specific origin to destination, non-substitutable demands into account. Based on this, they presented a number of model formulations for the MC-2E-VRPTW and developed an exact solution approach for the problem. Küçükoğlu et al. [16] studied the vehicle routing problem with backhauls and time windows (VRPBTW) and formulated the problem with the existing mathematical formulation of Vehicle Routing Problem with Backhauls (VRPB). The Multiple Trip Vehicle Routing Problem with Backhauls (MT-VRPB) was investigated in [17]. They presented an integer linear programming formulation of the MT-VRPB and reported the CPLEX results for small and medium size instances. Poikonen and Golden [18] studied the k-Multi-visit Drone Routing Problem(k-MVDRP). They considered a tandem between a truck and k drones. And the proposed model not only allows for a drone to carry multiple heterogeneous packages but also allows the specification of a drone energy drain function consider each package’s weight. Kitjacharoenchai et al. [19] introduced a new routing model that considers a synchronized truck-drone operation by allowing multiple drones to fly from a truck. These drones serve one or multiple customers and return to the same truck for a battery swap and package retrieval. Two efficient heuristic algorithms were proposed to solve the problem. Wang and Sheu [20] studied the vehicle routing problem with drones (VRPD), where not only trucks but drones were used to deliver parcels to customers. One distinctive feature of the VRPD is that a drone may travel with a truck, take off from its stop to serve customers, and land at service hub to travel with another truck as long as the flying range and loading capacity limitations are satisfied.

There are various methods investigated by researchers to solve these problems. These methods could be roughly divided into two categories: exact methods and approximate approaches. The exact methods used for VRP and its variants mainly are Branch-and-Bound algorithm. For instance, Lalla-Ruiz et al. [21] presented an exact method based on a new integer programming formulation to address the Multi-Depot Open Vehicle Routing Problem (MDOVRP). The experiment showed that the new programming formulation has a great advantage in reducing computation time. The Branch-cut-and-price method was used in [22] to solve the robust capacitated vehicle routing problem with knapsack uncertainty. A heuristic improvement procedure for Open Vehicle Routing Problem (OVRP) based on Integer Linear Programming techniques were presented in [23]. The proposed algorithm can find the new best solution for the considered instances in several cases. In the article [24], a novel model for the solution of a fuzzy vehicle routing problem was presented where the times and distance are allowed to be fuzzy numbers. Dondo and Cerda [25] introduced a new model-based improvement methodology for the multi-depot heterogeneous-fleet VRPTW problem to enhance an initial solution through solving a series of Mixed Integer Linear Programming mathematical problems. In their problem, exchanging and reordering of customers among routes are allowed.

However, the above exact methods were unable to solve VRPs within a reasonable time with larger instances in practice. Researchers have begun to utilize approximate approaches, including dedicated heuristics and metaheuristics, to solve VRPs due to the mentioned reason. Some of the literatures about approximate algorithms are listed below. Zaza and Richards [5] applied Ant Colony Optimization (ACO) to solve the routing and tasking problems for swarm of drones and obtain a novel result. Compared to the first version of ACO, a bi-criterion optimization was used to enhance the ACO. Because the MDVRP integrates some hard optimization problems, a hybrid genetic algorithm was proposed in [26]. To be specific, there are three heuristic was hybridized in the algorithm: Clarke and Wright saving methods, the nearest neighbor heuristic, and the iterated swap procedure. A method entitled as cluster-first, route-second is proposed by Fisher and Jaikumar [27]. The method firstly clustered customers into K clusters and made route planning for each cluster of customers. This method can reduce the large-scale problems into small ones and get the ideal results. Olgun et al. [28] studied the green vehicle routing problem with Simultaneous pickup and delivery(G-VRPSPD) and
proposed a hyper-heuristic (HH-ILS) algorithm based on iterative local search and variable neighborhood descent heuristics to effectively solve the problem. The HH-ILS algorithm yields competitive results when comparing with the state-of-the-art heuristics in the literature. An algorithmic framework was proposed in OR which successfully solves three vehicle routing problems: the multi-depot VRP, the periodic VRP, and the multi-depot periodic VRP with capacitated vehicles and constrained route duration. Their computational experiments obtained new best solutions for all available benchmark instances. Furthermore, their algorithms also provided extremely competitive results for the capacitated VRP.

In this paper, we studied a problem called modified multi-depot UAV routing problem (MMDUAVRP), which is different from the classical multi-depot vehicle routing problem: the UAV may not return to the start depot after completing the task and just need to return to any depot. This is a very reasonable arrangement in the UAV patrol because the provided UAV at all depots are generally the same. Therefore, there is no reason to limit the drone to return to its starting point. In the work of [30], they proposed a new problem which is similar to ours. In the MDOVRP, the vehicle do not return to the depot after delivering the goods to the customers i.e., the end of the route is not depot. However, this is not feasible during UAV patrol due to the fact that drone must return to the depot for recharging, maintenance, etc. A heuristic-assignment based hybrid large neighborhood search (HLNS) is proposed to solve the MMDUAVRP problem.

Our major contributions in this work are as follows:
- We adjusted the classic MDVRP to MMDUAVRP which meets the practical needs for UAV patrolling.
- We proposed the HLNS algorithm to address the MMDUAVRP problem with competitive results.

This paper is organized as follows: The problem definition and mathematical formulation are presented in II. The section III provides the details of the HLNS. The section IV presents the computational experiments and comparative study results with other methods. Finally, the section V states the conclusion.

II. PROBLEM FORMULATION
In the section II-A, we defined the problem verbally and present problem assumptions. A Mixed Integer Linear Programming (MILP) model is presented in section II-B.

A. PROBLEM DESCRIPTION
As a variation of the conventional vehicle routing problem, the MMDUAVRP supposes that the UAV departs from depots and return to the depots (it may not necessarily be exact the departure-depot) after completing all tasks with considering all constraints. Our primary objective is to minimize the number of UAVs used and the secondary objective is the total distance traveled by all UAVs. In this problem, all task points must be served by UAVs. A UAV departs from one depot to serve task points and returns to depot after finish its mission. For a UAV, the departure depot and the return depot can be different. For each depot, the number of UAVs leaving and returning should be equal. The UAV must ensure that its remaining power could ensure the returning to the depot while performing missions.

Let \( S_D \) be the set of depots. Let \( S_K \) and \( S_J \) denote the set of UAVs and the set of task nodes. Let \( ST_f \) denote the service time of the drone at each task point. This means that each drone must wait \( ST_f \) at task points before flying to the next task points. The speed of drone is denoted as \( TS_K \). The max flying time of drone is denoted as \( T_D \).

Each arc \( a \in A_k \) and each UAV \( k \in S_K \) is associated with a travel distance. We assume that the triangle inequality holds for distance between two points. A tour of a UAV must start from depot and return to depot in the graph \( G \). The route is feasible if the UAV \( k \) could complete all tasks and return to the depot in the \( T_D \).

This problem could be defined on a directed graph \( G = (V, A) \) where the vertex set is \( V = S_D \cup S_J \) and the arc set is \( A = \cup_{k \in S_K} A_k \), with \( A_k \in V \times V \) representing the feasible movements of each UAV \( k \in S_K \). A route for a UAV corresponds to a trajectory from its departure depot and returning depot in the graph \( G \).

In summary, we made the following assumptions in this paper:
- All UAVs have equal flight endurance and speed.
- There are sufficient UAVs available for each depot.
- There are some task points that need drone to serve and the service duration for each task is the same.
- Each task point must be served once and only once by a UAV.
- For each depot, the number of UAVs departing and returning are equal.
- The departure and returning depot of UAV could be different.
- We allow the situation that some depots are not used.

A feasible solution to the problem is a collection of tours where each tour starts and ends on depot without exceeding the maximum flight time. For a route, we allow it only contains depot without task points to guarantee that the number of UAVs departing and returning from each depot are equal. There are also not constraints to restrict all depots must be used. An example of the problem is represented in Fig 1.

B. MATHEMATICAL FORMULATION
A Mixed Integer Linear Programming (MILP) model is proposed to solve the MDUAVRP problem in this section. The notations are as follows.

1) NOTATIONS
- \( S_K \): The set of UAVs
- \( S_J \): The set of task points
- \( S_D \): The set of depots
- \( S_F \): \( S_D \cup S_J \)
The index of tasks and depots

The distance between node \(i\) and \(j\)

The service time of task point \(j\)

The maximum flight time of UAV \(k\)

The travel speed of the UAV \(k\)

The number of task points

The number of UA Vs

The number of depots

Binary decision variable, \(X_{ijk}\)

Note that the service time of the each depot is 0.

2) FORMULATION

\[
\min f_1 = \sum_{i \in S_F} \sum_{k \in S_k} \sum_{j \in S_O} X_{ijk}
\]

\[
\min f_2 = \sum_{i \in S_F} \sum_{k \in S_k} \sum_{j \in S_F} X_{ijk}(D_{ij})
\]

In this paper, we consider a hierarchical objective of minimizing number of UA Vs (primary objective) and total distance (secondary objective) which are defined in Eqs. (1) and (2), respectively.

\[
\sum_{j \in S_j} \sum_{k \in S_k} X_{ijk} = \sum_{j \in S_j} \sum_{k \in S_k} X_{ijk} \quad \forall i \in S_O
\]

Constraint (3) states that the number of drones finishing flight at depot \(i\) is equal to the number of UA Vs starting flight at depot \(i\).

\[
\sum_{i \in S_F} \sum_{k \in S_k} X_{ijk} = 1 \quad \forall j \in S_J
\]

\[
\sum_{j \in S_F} \sum_{k \in S_k} X_{ijk} = 1 \quad \forall i \in S_J
\]

Constraints (4) and (5) make sure each task is visited once by only one UAV, where Constraint (4) ensures that each task is started once by a UAV, Constraint (5) guarantees that each task is finished once by a UAV.

\[
\sum_{i \in S_F} \sum_{j \in S_J} X_{ijk} \quad \forall j \in S_J, \forall k \in S_K
\]

Constraint (6) determines the UAV \(k\) does not end its flight during a mission.

\[
\sum_{i \in S_O} \sum_{j \in S_J} X_{ijk} \leq 1 \quad \forall k \in S_K
\]

Constraint (7) indicates that UAV \(k\) starts its flight from the depot.

\[
\sum_{i \in S_J} \sum_{j \in S_O} X_{ijk} \leq 1 \quad \forall k \in S_K
\]

Constraint (8) ensures that UAV \(k\) must finish flight at an depot.

\[
\sum_{i \in S_F} \sum_{j \in S_F} X_{ijk}(D_{ij}/T_{Sk} + S_{Tj}) \leq Q_k \quad \forall k \in S_K
\]

According to Constraint (9), the travel time of each flight cannot exceed the maximum flight duration of a UAV.

\[
u_{ik} - u_{jk} + N X_{ijk} \leq N - 1 \quad \forall i \in S_J, \forall j \in S_J, \forall k \in S_K, i \neq j
\]

Constraint (10) is used to eliminate tours that do not start and end at a depot and tours that visit more than \(N\) points, where the variable \(u_{ik}\) does not require any initialization and \(N\) is the sum of \(N_O\) and \(N_J\).

\[
X_{ijk} \in \{0, 1\} \quad \forall i \in S_F, \forall j \in S_F, \forall k \in S_K
\]

The decision variable of the proposed MILP is prescribed in constraint (11).

III. ALGORITHM DESCRIPTION

A heuristic-assignment based hybrid large neighborhood search (HLNS) is proposed in this paper to solve the problem raised above. This algorithm consists of three components: constructing initial solution, large neighborhood search, local search. The process of constructing initial solution is described in the section III-A. The large neighborhood

FIGURE 1. An example of the problem.
search is illustrated in the section III-B. The section III-C presents the detail of local search. The framework of the HLNS algorithm is stated in Algorithm 1.

Algorithm 1 The Pseudo-Code of HLNS
Input: Data of problem
Output: The best solution
Begin
data = get_data()
s = init(data)
global_best = s
while i < n
i = i + 1
s = large_neighborhood_search(s)
s = local_search(s)
if cal_cost(s) < cal_cost(global_best)
global_best = s
end if
end while
return global_best
End

A. CONSTRUCT INITIAL SOLUTION
Since an appropriate initial solution can accelerate the convergence speed of algorithm and improve the quality of the final solution, the initial solution is constructed in two steps. At the first step, three methods are used to generate the order of serving task points. Then, we use the heuristic-assigned to determine the departing and returning depot of the UAV. The best of them is selected as the initial solution. The details of these methods are given below and the pseudo code is stated in Algorithm 2.

Algorithm 2 The Pseudo-Code of Constructing Initial Solution
Input: The information of task points and depots
Output: Initial solutions
Begin
data = get_information_of_taskPoints_depos()  
a1 = modified_minimum_spanning_tree(data)  
a2 = random_1(data)  
a3 = random_2(a1, data)  
a1 = heuristic_assignment(a1)  
a2 = heuristic_assignment(a2)  
a3 = heuristic_assignment(a3)  
a = best of a1, a2, a3
return a
End

1) MODIFIED MINIMUM SPANNING TREE
The arcs in the minimum spanning tree have a high probability of appearing in the optimal solution according to Helsgaun [31]. For example, Gen and Syarif [12] used the spanning tree based GA(Genetic algorithm) to address a production/distribution problem and get a novel result. Based on these ideas, the modified spanning tree is applied in constructing initial solution. The specific steps are as follows.

At the beginning, we construct a complete graph whose vertex is composed of a depot and all unserved task points, where the depot is randomly selected through all depots. Next, we could obtain a minimum spanning tree of the graph. Then, we find a path in the minimum spanning tree which should start from the depot and contain most task points. If there are still unserved task points, we continue to build the minimum spanning tree for remaining task points. Iteration of the above process until all task points have been served. After all task points are served, we concatenate these paths after removing all depots.

A simple example is illustrated in Fig. 1 to explain the method mentioned above where the served task points are colored. At Step 0, there are 8 task points to be served and 2 depots could be used. Then, the depot 1 is selected randomly and a minimum spanning tree is obtained. There are two paths which start form the depot in the minimum spanning tree:[1, 5, 8, 4, 7, 6, 9], [1, 2, 3]. The first path is selected due to it contains most task points at Step 1. It is represented by the bold line. Subsequently, the minimum spanning tree of remaining task points is constructed in the same way at Step 2. The depot 0 is selected randomly and there are 2 task points that have not been served at Step 3. The path [0, 2, 3] is obtained at this step. All task points have been served so far. Then, we concatenate these paths after removing depots [0, 1]. Finally, the service order of the task points [5, 8, 4, 7, 6, 9, 2, 3] could be determined. To be noticed, it is not a feasible solution due to the limit of max flying time and the undetermined departing/returning depots. The following heuristic-assignment method is used to determine the depots of departing and returning.

2) RANDOM SELECTION
In this paper, the randomly selected method is used to diversify the initial solution. Specifically, there are two random strategies to generate initial solution. The first one is to calculate a random permutation of the task points. Although the initial solution produced by this method is currently very poor, it is helpful to explore the solution space. Another method is to randomly exchange the order of the two tasks determined by modified spanning tree. This method could balance the quality of the initial solution and the exploitation of the solution space. The first methods is denoted as random_1 and the second method is denoted as random_2.

3) HEURISTIC-ASSIGNMENT
This section would provide the detail of heuristic-assignment. At the beginning, all possible combinations of departure and returning depots are considered and the best is chosen. For example, an order of servicing the task points is [5, 8, 4, 7, 6, 9, 2, 3] and the departure depot is 0 while the returning depot 1. Then, the first tour [0, 5, 8, 4, 1] is obtained with
considering the limit of max flying time. Next, we set the departure depot of the second tour as the return depot of the first tour. All possible return depots are considered and that we choose the best one. Here we assume the best return depot is 0 in this case and now we get the second tour [1, 7, 6, 9, 2, 3, 0]. We repeat the above process if there are still remaining unserved task points. There is a special case in which the return depot of second tour is 1. We need to add an extra tour [1, 0] to ensure the number of UAVs departing and returning for this depot is equal. Furthermore, the method ensures that we only need to add at most one “zero-task” tour at the end in any case, where the “zero-task” tour means that it only contains depots without task points.

B. LARGE NEIGHBORHOOD SEARCH

This section introduces the details of Large Neighborhood Search (LNS). The independent heuristic algorithm generally has some shortcomings such as falling into local optimum or premature stagnation. The research community attempted to develop hybrid algorithm to overcome this disadvantage since the hybrid algorithm generally represents synergy. In this paper, remove, insertion and relocation operators are utilized in the LNS process. The remove operators are applied at the beginning of the LNS and this will make the solution incomplete. Then, we use insertion operators to repair the solution. Finally, relocations operators are used to further improve the solution. The detailed description is presented in the following phase and the pseudo code of the LNS is shown in Algorithm 3.

Algorithm 3 The Pseudo-Code of LNS

| Input | A feasible solution a |
|-------|-----------------------|
| Output| A improved solution   |
| Begin  |                        |
|        | a' = remove(a)         |
|        | a' = insert(a')        |
|        | a' = relocation(a')    |
|        | return a'              |
| End    |                        |

1) REMOVAL OPERATORS

A remove operator works that a predetermined number of task points are removed from a route. Three removal operators are applied in the proposed HLNS algorithm: related removal, worst removal, random removal. The q task points are generated randomly from the interval [1, α * |Z|] at the beginning of this step. The |Z| represents the maximum number of task points which could be removed and α represents the ratio of removal.

- Related removal. It is proposed by Shaw in [32]. This idea is based on that similar task points should be given a chance to have a profitable position to re-inserted. Shaw proposed a relatedness \( R(u, v) \) to measure the similarity between two task points u and v. The distance between two task points are employed to represent the relatedness.
- Worst removal. It is introduced in [33]. At the beginning, the cost saving is calculated by removal of each task point. Then q task points are removed based on the larger saving which corresponds to a larger probability.
- Random removal. This heuristic randomly selects q task points to remove from the route. This idea is simple but could perturb slightly the current solution to avoid tamping into local optimality.

These operators aim at diversification the search such as random removal. In addition, the worst and related removal also includes intensification which is related to some indications such as cost saving to guide the removal operator.

2) INSERTION OPERATORS

The solution would be infeasible after applying remove operator. So, the infeasible solution must be repaired to make it feasible again. The works of [33] provided some simple but effective insertion operators. In this paper, the basic greedy and regret heuristic are employed to repair infeasible solution.

- Greedy insertion. This operator is firstly used to determine that each removed target point need to be reinserted into the optimal position (considering all possible route and all possible position). Let \( \delta_{i,k,u} \) denotes the change in the object function obtained by inserting target point into the position\( u \) of route \( k \). Then, we insert the target point into the best position with the minimum \( \delta_{i,k,u} \).
- Regret-k insertion. The heuristic insertion is an extension of greedy insertion. It takes the second cheapest,
the third cheapest and k-cheapest into account of considering to minimum insertion cost. This method will degenerate to greedy insertion when k is set to 1. Ropke and Pisinger [33] gave a complete description of this heuristic. This paper considered three regret-k heuristics with \( k \in \{2, 3\} \).

- Random best insertion. This method is proposed by Wolflinger [35]. A randomly task point selected is reinserted at its best position.

3) RELOCATION

This subsection describes the relocation method. This method is mainly focus on the depots. The starting depot and returning depot of the first route are very important according to the heuristic-assignment mentioned in III-A. This method selects k groups of depots and each group contains two depots. These groups are defined as \((s_i, e_i)\) where \(i \in [1, k]\). Then, we set the starting depot and returning depot of the first route as \((s_i, e_i)\) and construct a feasible solution according to the heuristic-assignment method. We could obtain k solutions. There are three strategies to select one solution from k solutions.

- We simply select the best one. This greedy strategy generally could obtain a good solution.
- Randomly choosing one. This strategy is mainly used to explore the search space.
- We select the solution based the larger value of object function which corresponds to a smaller probability. This strategy could balance the exploration and exploitation. This strategy is applied in our numerical experiments and k is set to 5.

C. LOCAL SEARCH

It is required for the neighborhood local search to tackle problem without trapped in local optima and intensify the search space. The neighborhood local search is widely and successfully applied in the combinatorial problems. Based on this, we use five neighborhood local search from [36] and [34] to further improve the quality of our solution. The following subsection provides the details of these neighborhood search structure: k-opt, exchange, swap(u,v), shift(u), crossover.

- K-opt. This structure removes k links from cyclic route and obtains k open segments. We can manipulate and combine them to get a new route. In this study, the \(k \in \{2, 3\}\) is considered.
- Exchange. This structure aims to swap the position of two task points in the same route.
- Swap(u,v). This structure means that we select u task points in a route and v task points in another route in the first place. Then, exchange these points selected with considering the limit of max flying time. The values of u and v are in the interval [1,3] in the experiment of this paper.
- Shift(u). This structure firstly removes u task points from its route and reinsert them into another route. In our implementation, u is set to 1 and 2.

- Crossover. Firstly, this structure selects two route r1 and r2. Then, the route r1 and r2 are divided into two part. To be noticed here, starting and ending points are not included while dividing route. Finally, the first part of r1 is connected to second part of r2 and the first part of r2 is connected to second part of r1.

We received a poor solution with probability \(P\) to balance exploitation and exploration when performing local search on current solution. To be specific, we receive a solution if it is better than the current solution or we receive it with a probability \(P\). The pseudo code of local search is illustrated as the Algorithm 4.

**Algorithm 4 The Pseudo-Code of Local Search**

| Input: | A feasible solution \(a\) |
| --- | --- |
| Output: | A improved solution \(a’\) |

Begin

\[
a’ = k\_opt(a)
\]

\[
a’ = exchange(a’)
\]

\[
a’ = swap(a’)
\]

\[
a’ = shift(a’)
\]

\[
a’ = crossover(a’)
\]

return \(a’\)

End

Different neighborhood local search operators are used to improve different types of solutions. The specific details are described in [36]. The following Fig. 3 visually illustrates the different effectiveness of each neighborhood local search operators in our solution.

IV. PERFORMANCE EVALUATION

In this section, the experimental results are presented to illustrate the effectiveness of the proposed HLNS algorithm to solve MMDUAVRP problem. First, we compare our proposal with state-of-art algorithm on the Multi Traveling Salesman Problem(MTSP) and obtain the desired result. Then we separately test the performance of HLNS on small-scale and large-scale MMDUAVRP. On the small-scale MMDUAVRP, we compared HLNS with CPLEX and achieved consistent results with a huge time advantage. Furthermore, we compare HLNS with some well-known algorithms on large-scale instances of MMDUAVRP and the results comprehensive verified performance of HLNS. The HLNS is coded in Visual Studio C++ 2017. All experiments have been performed on a computer with Intel Core i7-9700K at 3.60 GHz and 16Gb of RAM.

The UAV parameters used in this paper are given here. The max flying time of UAV is 90 minutes. The velocity of UAV is 1500 meters per minute. The duration required to stay at each task point is 5 minutes. Each unit of distance in the test instance represents 100 meters. In addition, we set the maximum number of iterations to 200 in testing MTSP and MMDUAVRP. The computing time of these algorithms are basically same for the fairness of comparison under the experimental parameter setting. All algorithms run ten times.
TABLE 1. Experimental results of MTSP.

| Customer | salesman | GA | GGA-SS | ACO | TCX | IWO | GVNS | MOAT-ACO | HLNS | GAP |
|----------|----------|----|--------|-----|-----|-----|------|---------|------|-----|
|          |          | Avg | Avg   | Avg | Avg | S.D. | Best | Avg     | Avg  |     |
| 50       | 5        | 543 | 449   | 448 | 492 | 446 | 449  | 2       | 446  | 449 |
| 10       | 723      | 584 | 584   | 567 | 581 | 581 | 582  | 2       | 582  | 580 |
| 100      | 26653    | 23737 | 22619 | 24130 | 24194 | 94 | 21798  | 22249 | 349 | 21798 |
| 5        | 30408    | 23768 | 24166  | 28612 | 23139 | 29 | 33924  | 24706 | 694 | 23294 |
| 10       | 31227    | 28488 | 27990  | 30998 | 24792 | 90 | 32661  | 28896 | 434 | 28380 |
| 20       | 54700    | 40492 | 39949  | 44686 | 34357 | 95 | 32845  | 40327 | 234 | 39764 |
| 150      | 56218    | 38434 | 39247  | 44674 | 36095 | 81 | 37957  | 36095 | 360 | 38535 |
| 5        | 49947    | 39682 | 40647  | 47811 | 38881 | 120 | 38714  | 39953 | 389 | 39644 |
| 10       | 54058    | 44274 | 46366  | 51326 | 41462 | 112 | 42254  | 43186 | 298 | 45264 |
| 20       | 73934    | 56412 | 55960  | 62400 | 53412 | 106 | 53475  | 55779 | 262 | 55513 |
| 30       | 99547    | 72783 | 71266  | 78023 | 69771 | 188 | 69541  | 71073 | 640 | 70421 |

for each benchmark instance and the average of these results is used.

A. ALGORITHM VERIFICATION FOR MTSP

In order to check the performance of the proposed HLNS algorithm, it is necessary to compare it with other metaheuristics for benchmarking purposes. We test HLNS in MTSP instances in this section due to the lack of MMDUA VRP benchmark. In fact, MMDUA VRP can be converted to MTSP when there is only one depot and flight time of UAV is not considered. In the MTSP problem, a set of cities must be visited by a team of salesmen such that each city is visited exactly once by only one salesman. The MTSP has two commonly optimization objectives: min-sum, min-max. The min-max is to minimize the maximum distance of sub-tours traveled by salesmen. The min-sum is to minimize the sum of the length of sub-tours by all salesmen. In this section, we use min-sum as objective of MTSP due to it is similar to the objective of MMDUA VRP. In fact, we set the flight time to infinity to handle the MTSP in this paper. The Table 1 shows the result of experiment against the state-of-art MTSP methods on the classical benchmarks from literature [37]. It includes 12 problems with n = 51, 100 and 150 points. The problem with 51 points is tested separately where the number of salesmen equals to 3, 5, 10. The problem with 100 points is tested separately with the number of salesmen equals to 3, 5, 10, 20. The problem with 100 points is tested separately when the number of salesmen equals to 3, 5, 10, 20, 30. The Table 1 summarizes the experimental results and bold text indicates the best results. We use the GAP value as a statistical indicator as the same as [37]. The GAP is computed with 

\[
gap = \frac{\text{best} - \text{BKS}}{\text{BKS}} \times 100\%
\]

where the best is best solution of HLNS and the BKS is the best solution of other algorithms. The comparison algorithms used in this experiment is as follows:

- **GAs.** A genetic algorithm using a new two-part chromosome representation and related genetic operators([38], 2006).
- **GGA-SS.** A grouping genetic algorithm using a different chromosome representation and different crossover & mutation([39], 2007).
- **ACO.** An ant-colony optimization algorithm for solving MTSP([40], 2009).
- **TCX.** A new crossover operator called two-part chromosome crossover for solving MTSP using a genetic algorithm([41],2013).
- **IWO.** An approach based on invasive weed optimization algorithm([42],2015).
- **MOAT-ACO.** Mission-oriented ant-team algorithm([37],2019).
- **GVNS.** A general variable neighborhood search approach for the MTSP([43], 2015).

These results show that the HLNS could get the best result in more than half of the instances(6 best results out of 11 instances). In the remaining instances, the gap to the best solution is only less than 2%. To be notice that, there are no specific operators for MTSP problem in our algorithm. For all these, our algorithm still achieves a competitive result when applied to other similar problems such as MTSP. This
indicates that the algorithm proposed in this paper has a better generalization performance.

B. NUMERICAL EXPERIMENT FOR MMDUAVRP

In this section, a series of experiments are represented to verify the efficiency of HLNS. We first compared the results of CPLEX on small-scale instances. Then we test HLNS on large-scale instances to further explore its potential. We implement the Discrete Particle Swarm optimization (DPSO) [44], Memetic algorithm (MA) [45] and Adaptive Large Neighborhood search (ALNS) [33] as comparison algorithms. We made some modifications on these algorithms due to they are not specifically aimed to solve MMDUAVRP. A brief implementation is as follows. Their considered vehicle routing problems are single-depot, so we randomly select a depot as the single-depot. Then, we neglect the constraints in these problems such as vehicle capacity and add the maximum travel distance of vehicle to fit our problem. We remove the single-depot selected previously and a sequence of task points being served could be obtained. Finally, departing and returning depot could be determined by using heuristic-assignment mentioned in section III.

The instances we used are based on the classic MDVRP instances [46]. The MDVRP instances contain a lot of information such as time window, demand of customer, etc., where we only use the coordinate of points. Our revised instances could be downloaded from https://github.com/mengbinghen/MMDUA VRP.

The experimental results compared with CPLEX are exhibited in Table 2. The first column represents the name of instances and the following two columns represented the number of depots and tasks used. The rest shows the results obtained by CPLEX and HLNS. We can see that CPLEX is able to solve 11 instances optimality. The rest one could not get the solution due to “out of memory” which is marked as “N/A” in the table. For 11 instances, the HLNS proposed in paper could get their optimal solution within much shorter time.(0.07-0.08 minutes). For the rest one instance could not be solved by CPLEX, the HLNS also could get solution in 0.112 minutes.

Further evaluation was carried out by testing the performance of HLNS on 11 large-scale instances. The computational results are represented in the Table 3, where the first column provides the name of problem instances, the second column provides the number of task points and the following is the number of depots. The rest is detailed result data, which contains the number of UAVs used and the total distance. The bold text indicates that the data is best among these algorithms where only current objective is considered. For example, the bold text “222.50” of p21 in the Table 3 only indicates the solution obtained by HLNS has the shortest total distance among these algorithms and it is obviously not the best solution due to more UAVs are used. Through these results, we can argue that the HLNS obtains better solutions than all other algorithms for 10 instances(on average 90% better for 11 test instances). The HLNS did not achieve the best result in only one instance p21 due to more UAVs are used. The table also indicates that HLNS could obtain better results than DPSO and MA within a shorter computing time. The HLNS could obtain better results than ALNS in similar computing time. The last two rows in the table are the result of paired t-test. The paired t-test is a well-known method to verify that whether there is a significant difference between the two sets of data. We separately conduct the paired t-test on the sets of distance obtained by DPSO, MA, ALNS since the NV values of these algorithms are almost the same. As we can see from the table, the p-values are all less than 5%, which means that our algorithm is better than other algorithms at the significance level of 5%.

C. IMPACT OF POINT LOCATION

This section mainly discusses the influence of the distribution of task points and the location of depot points on the solution. In these experiments, the service time of each task point is 1 minute and each unit distance represents 500m. The maximum number of iterations is 50, the remaining parameters are the same as the above experiment. We conducted a series of experiments and the results are shown in Table 3 (The N/A represents that there are no solutions for this situation due to some task points are too far from depot). We used two types of data distribution: R-type and C-type. R-type means that task points generated uniformly randomly over a square. In C-type we cluster task points based on R-type. Three instances are generated for each data distribution and all of these contain 50 task points. Then, we use k-means and random method to determine the depot separately and the number of depot points is set to 2, 4, 6.
The experiment of results show that the use of k-means to determine the position of the depot is significantly better than randomly determining the position of the depot, especially in C-type. The results also show that C-type cases have more positive effects on final result than R-type as illustrated in Fig. 4. In addition, randomly determining the location of the depots usually results in some task points being too far away to be accessible. Furthermore, the number of depot points is not always the more the better, when using k-means. As shown in Table 3, increasing the number of depots will lead to missing the optimal position in some cases.

V. CONCLUSION

The MMDUAVRP is studied in this paper. It is a new version of MDVRP, where the departure and return depots can be different for each UAV. The maximum flight time of UAV is also considered in the problem. Our objective is to minimize the total distance traveled by all UAVs and the mathematical model of the problem is provided in this paper. Due to the
complexity of the problem, this paper developed a heuristic-assignment based hybrid large neighborhood search. Because there are no existing benchmarks, this study first tested HLNS on MTSP and obtained competitive. Then, we generated some test instances which were revised from the well-known MDVRP instances. Comparisons between the results of CPLEX and HLNS on small-scale instances showed that the HLNS could provide the same optimal solutions within a comparatively shorter period of time. On the large-scale instances, we compared the HLNS with the DPSO, MA, and ALNS algorithms which are widely applied for solving vehicle routing problems. The results also suggest that our HLNS outperforms these algorithms in similar computing time.

REFERENCES

[1] G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte, “Drone-aided routing: A literature review,” Transp. Res. C, Emerg. Technol., vol. 120, Nov. 2020, Art. no. 102762.

[2] K. S. Lee, M. Ovinnis, T. Nagarajan, R. Seulin, and O. Morel, “Autonomous patrol and surveillance system using unmanned aerial vehicles,” in Proc. IEEE Int. Conf. on Artif. Intell.-CS/STEM. Comput. Ind. Eng. (IEEECSIC), Jun. 2015, pp. 1291–1297.

[3] Y. Zhou, T. Rui, Y. Li, and X. Zuo, “A UAV patrol system using panoramic stitching and object detection,” Comput. Electr. Eng., vol. 90, Dec. 2019, Art. no. 106473.

[4] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle routing problems for drone delivery,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 47, no. 1, pp. 70–85, Jan. 2017.

[5] T. Zaza and A. Richards, “Ant colony optimization for routing and tasking problems for teams of UAVs,” in Proc. UKACC Int. Conf. Control (CONTROL), Loughborough, U.K., Jul. 2014, pp. 652–655.

[6] J. Ries and A. Ishizaka, “A multi-criteria support system for dynamic aerial vehicle routing problems,” in Proc. CCA, Marseilles, France, Dec. 2012, pp. 1–4.

[7] O. Thakoor, J. Garg, and R. Nagi, “Multiagent UAV routing: A game theory analysis with tight price of anarchy bounds,” IEEE Trans. Automat. Sci. Eng., vol. 17, no. 1, pp. 100–116, Jan. 2020.

[8] V. Yadav and A. Narasimhamurthy, “A heuristics based approach for optimizing delivery schedule of an unmanned aerial vehicle (drone) based delivery system,” in Proc. 9th Int. Conf. Adv. Pattern Recognit. (ICAPR), Benalmádua, India, Dec. 2017, pp. 1–6.

[9] S. G. Manyam, S. Rasmussen, D. W. Casbeer, K. Kalyanam, and S. Manickam, “Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions,” in Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS), Miami, FL, USA, Jun. 2017, pp. 573–580.

[10] B. N. Coelho, V. N. Coelho, I. M. Coelho, L. S. Ochi, K. R. Haghnavaz, D. Zuidema, M. S. F. Lima, and A. R. da Costa, “A multi-objective green UAV routing,” Comput. Oper. Res., vol. 208, pp. 306–315, Dec. 2017.

[11] G. B. Dantzig and I. H. Ramser, “The truck dispatching problem,” Manage. Sci., vol. 6, no. 1, pp. 80–91, 1959.

[12] M. Gen and A. Syarif, “Hybrid genetic algorithm for multi-time period production/distribution planning,” Comput. Ind. Eng., vol. 48, no. 4, pp. 799–809, Jun. 2005.

[13] G. Clarke and J. W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” Oper. Res., vol. 12, no. 4, pp. 568–581, 1964.

[14] L. Zhou, C. Ma, K. Wang, L. Xiao, and W. Zhang, “Multi-depot multi-trip vehicle routing problem with time windows and release dates,” Transp. Res. E, Logistics Transp. Res., vol. 135, Mar. 2020, Art. no. 101866.

[15] N. Dellaert, T. Van Woensel, T. G. Crainic, and F. D. Sardaraq, “A multi-commodity two-echelon capacitated vehicle routing problem with time windows: Model formulations and solution approach,” Comput. Oper. Res., vol. 127, Mar. 2021, Art. no. 105154.

[16] I. Kičukoglu and N. Öztürk, “An advanced hybrid meta-heuristic algorithm for the vehicle routing problem with backhauls and time windows,” Comput. Ind. Eng., vol. 86, pp. 60–68, Aug. 2015.
[43] B. Soylu, “A general variable neighborhood search heuristic for multiple traveling salesmen problem,” Comput. Ind. Eng., vol. 90, pp. 390–401, Dec. 2015.

[44] Y.-J. Gong, J. Zhang, O. Liu, R.-Z. Huang, H. S.-H. Chung, and Y.-H. Shi, “Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 42, no. 2, pp. 254–267, Mar. 2012.

[45] Y. Nagata, O. Bräysy, and W. Dullaert, “A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows,” Comput. Oper. Res., vol. 37, no. 4, pp. 724–737, Apr. 2010.

[46] J. Renaud, G. Laporte, and F. F. Boctor, “A tabu search heuristic for the multi-depot vehicle routing problem,” Comput. Oper. Res., vol. 23, no. 3, pp. 229–235, Mar. 1996.

XIAOHUI LI was born in Xi’an, Shaanxi, China, in 1982. He received the B.Eng. degree from the Xi’an University of Technology, China, in 2004, the M.Eng. degree from Paul Sabatier University, France, in 2008, and the Ph.D. degree from the University of Technology of Troyes, France, in 2011. He has been with the School of Electronics and Control Engineering, Chang’an University, as a Lecturer, since 2012. His research interests include optimization algorithm, scheduling problem, vehicle routing problem, UAV path planning, and multi-objective optimization.

PEIFAN LI was born in Ruzhou, Henan, China, in 1997. He received the bachelor’s degree in electronics and control engineering from Chang’an University, in 2019, where he is currently pursuing the master’s degree. His main research interests include optimization algorithm, UAV path planning, and vehicle routing problem.

YI ZHAO (Member, IEEE) received the M.Eng. degree from Pierre and Marie Curie University (University Paris VI), in 2009, and the Ph.D. degree from the University of the South, Toulon-Var, in 2013. He has been with the School of Electronics and Control Engineering, Chang’an University, since 2014. His current research interests include machine learning and sensor networks.

LU ZHANG was born in Ya’an, Sichuan, China, in 1996. She received the bachelor’s degree in electrical engineering and its automation from the Sichuan University of Science and Engineering, in 2018. She is currently pursuing the master’s degree with Chang’an University. Her main research interests include path planning and optimization algorithm.

YUAN DONG was born in Xi’an, Shaanxi, China, in 1987. She received the B.E. degree in electronic and information engineering from Northwestern Polytechnical University, in 2009, and the master’s and Ph.D. degrees in computer science from the University of Technology of Troyes, France, in 2013. Since 2014, she has been doing her research and teaching with Chang’an University. She has authored over ten technical papers in refereed journals and conference proceedings. Her main research interests include machine learning, pattern recognition, and wireless heterogeneous networks, especially in study of image processing and decision systems.

***